UM PROVADOR PARACONSISTENTE[1]

Arthur Buchsbaum **Tarcísio Pequeno**

Resumo

Um método de prova automático para uma lógica paraconsistente, o cálculo $\mathbf{C_l}$ de da Costa, é apresentado. Trata-se de um método analítico utilizando um sistema de tableaux. Na realidade, dois sistemas de tableaux foram elaborados: um com um número pequeno de regras a partir do qual são provadas a consistência e a completude do método, e outro que, mostramos ser equivalente ao primeiro, é um sistema de regras derivadas a partir do qual foi realizada uma implementação.

Introdução

Este trabalho descreve um método de prova automático para uma lógica paraconsistente: o cálculo $\mathbf{C_l}$ de da Costa. Falando sucintamente, dizemos que uma lógica é paraconsistente se a ocorrência de uma contradição ($\mathbf{A} \wedge \neg\mathbf{A}$) em uma teoria não acarreta a sua trivialidade, isto é, toda fórmula ser demonstrável. Em outras palavras, munido de uma tal lógica um agente poderia manter crenças contraditórias sem que isso provocasse um colapso do seu sistema de crenças, o que ocorreria se tal agente usasse a lógica clássica. Na seção seguinte mostramos de que forma este efeito é conseguido na lógica de da Costa.

O método de prova aqui apresentado é um sistema de tableaux consistente e completo com respeito ao cálculo preposicional $\mathbf{C_l}$. Um protótipo baseado nesse sistema foi construído em LISP.

Na verdade foram desenvolvidos dois sistemas de tableaux para o cálculo $\mathbf{C_l}$. Tendo em vista que o "desvio" da lógica paraconsistente, respeito à lógica clássica, se dá na axiomatização da negação, enfatizamos aqui o sistema de tableaux relativo à estrutura conectiva da lógica. Assim, na seção 2, o sistema $\mathbf{SC_1}$, consistente e completo respeito ao cálculo proposicional paraconsistente $\mathbf{C_l}$, é apresentado.

Como o objetivo nessa etapa é a construção de um sistema de tableaux mais adequado para as provas de consistência e completude, o sistema $\mathbf{SC_1}$ foi projetado no sentido de tornar econômicas tais provas, buscando minimizar por exemplo o número de regras desenvolvimento. Não foram consideradas aí as propriedades do sistema com respeito à implementação. Esse problema é tratado na seção 3, na qual o sistema $\mathbf{S'C_1}$, derivado de $\mathbf{SC_1}$, é apresentado. $\mathbf{S'C_1}$ contém um número maior de regras, as quais são em geral mais especializadas, e tendem a gerar árvores menores que as árvores geradas pelas regras de $\mathbf{SC_1}$, dai o sistema $\mathbf{S'C_l}$ é mais conveniente para implementação. Mostramos que este sistema é equivalente a $\mathbf{SC_1}$, no sentido de provar os mesmos teoremas, e é este sistema que foi efetivamente utilizado na implementação de um protótipo.

1. A Lógica Paraconsistente de da Costa

Como foi dito na introdução, nas lógicas paraconsistentes, ao contrário das clássicas, do fato de **A** e $\neg\mathbf{A}$ serem dedutíveis em uma da teoria não decorre que qualquer fórmula **B** também o seja. De fato, essa propriedade é uma das três "principais" nomeadas por da Costa [74] a serem atendidas pelas lógicas paraconsistentes, e a única essencial para o

[1] Este artigo foi parcialmente financiado pela FINEP e por SID Informática, projeto ESTRA.

estabelecimento da paraconsistência, segundo Bunder [83]. Os dois outros princípios de da Costa são:

1) que o principio da não-contradição, que pode ser expresso por $\neg(\mathbf{A} \wedge \neg\mathbf{A})$, não seja em geral válido para qualquer **A**;
2) que todos os teoremas da lógica clássica que não interfiram com os dois princípios anteriores sejam mantidos.

Os dois primeiros princípios exigem o "enfraquecimento" da axiomática com relação aos sistemas clássicos. O terceiro assevera que esse enfraquecimento não deve ser maior que o necessário.

O principio da não-contradição decorre nas lógicas clássicas do axioma do absurdo

$$(\mathbf{A} \rightarrow \mathbf{B}) \rightarrow ((\mathbf{A} \rightarrow \neg\mathbf{B}) \rightarrow \neg\mathbf{A}).$$

A paraconsistência é obtida em $\mathbf{C_l}$ exatamente pela restrição desse axioma àquelas fórmulas para as quais o principio da não-contradição é explicitamente mantido. Esse axioma toma então a seguinte forma:

$$\neg(\mathbf{B} \wedge \neg\mathbf{B}) \rightarrow ((\mathbf{A} \rightarrow \mathbf{B}) \rightarrow ((\mathbf{A} \rightarrow \neg\mathbf{B}) \rightarrow \neg\mathbf{A}).$$

O cálculo $\mathbf{C_l}$ é formado por todos os axiomas proposicionais clássicos (como em Kleene [52]), com exceção do axioma do absurdo, o qual toma a forma indicada acima. Além disso é incluído como axioma

$$\mathbf{A} \vee \neg\mathbf{A},$$

uma vez que isso não é mais teorema, e axiomas que asseguram que da composição de fórmulas para as quais vale a não-contradição resultam fórmulas para as quais esse princípio permanece válido (utilizamos aqui uma abreviatura que é praxe na literatura sobre lógicas paraconsistentes: $\mathbf{A}^{\mathrm{o}}$ para $\neg(\mathbf{A} \wedge \neg\mathbf{A})$):

$$\mathbf{A}^{\mathrm{o}} \wedge \mathbf{B}^{\mathrm{o}} \rightarrow (\mathbf{A} \rightarrow \mathbf{B})^{\mathrm{o}};$$
$$\mathbf{A}^{\mathrm{o}} \wedge \mathbf{B}^{\mathrm{o}} \rightarrow (\mathbf{A} \wedge \mathbf{B})^{\mathrm{o}};$$
$$\mathbf{A}^{\mathrm{o}} \wedge \mathbf{B}^{\mathrm{o}} \rightarrow (\mathbf{A} \vee \mathbf{B})^{\mathrm{o}}.$$

Esse cálculo respeita os três requisitos mencionados acima. Todas as regras de dedução natural, por exemplo, são válidas, com exceção da regra de $\neg$-introdução (regra do absurdo).

Podemos dizer que no cálculo mostrado acima é dada uma axiomatização para a negação que a torna mais fraca que a negação clássica. Uma das motivações para as lógicas paraconsistentes é ganhar uma perspectiva para uma melhor compreensão da negação. O lógico Newton C. A. da Costa desenvolveu toda uma hierarquia de cálculos paraconsistentes $\mathbf{C_n}$, nos quais a negação é progressivamente mais fraca, obtidos como o descrito acima, colocando-se "(n)" no "o", sendo que $\mathbf{A}^{(n)}$ é definido recursivamente como segue:

$$\mathbf{A}^1 \equiv \mathbf{A}^{\mathrm{o}};$$
$$\mathbf{A}^{n+1} \equiv (\mathbf{A}^n)^{\mathrm{o}};$$
$$\mathbf{A}^{(n)} \equiv \mathbf{A}^1 \wedge \mathbf{A}^2 \wedge \ldots \wedge \mathbf{A}^n.$$

Uma versão predicativa de um cálculo paraconsistente é apresentada em da Costa [74]. Uma semântica paraconsistente é dada em da Costa & Wolf.

Neste trabalho apresentamos um método automático de prova apenas para o primeiro cálculo $\mathbf{C_l}$ da hierarquia, embora acreditamos que não é difícil a sua generalização para os demais membros, e na realidade estendemos o método para o cálculo de predicados paraconsistente $\mathbf{C_l}^*$.

2. SC_1: Um Sistema de Tableaux para C_1

Nessa seção um sistema de tableaux consistente e completo para o cálculo proposicional paraconsistente $\mathbf{C_1}$ é apresentado.

Um sistema de tableaux é um método de prova por refutação que consiste na geração de uma árvore (tableau), a partir do tableau inicial para o teorema a ser demonstrado. A árvore é gerada pela aplicação sucessiva de "regras de desenvolvimento de tableaux", que fazem brotar de todas as folhas descendentes de um nó escolhido não usado a respectiva aplicação ao nó. A proliferação de ramos é contida pela operação de "fechamento", através da qual um ramo, desde a raiz, é fechado, quando determinada condição ocorre nele. O objetivo do procedimento é "fechar" todos os ramos, o que provaria o teorema. Visto semanticamente, o fechamento de todos os ramos de um tableau .estabelece a insatisfatibilidade da fórmula ocupando a raiz e portanto a refuta.

O método dos tableaux é dito ser analítico, pois procede pela decomposição gradativa da fórmula proposta, em oposição ao que fazem métodos convencionais de prova automática por resolução. Note ainda que no método dos tableaux trabalhamos diretamente com as fórmulas dadas, sem recorrer à normalização na forma de cláusulas, por exemplo.

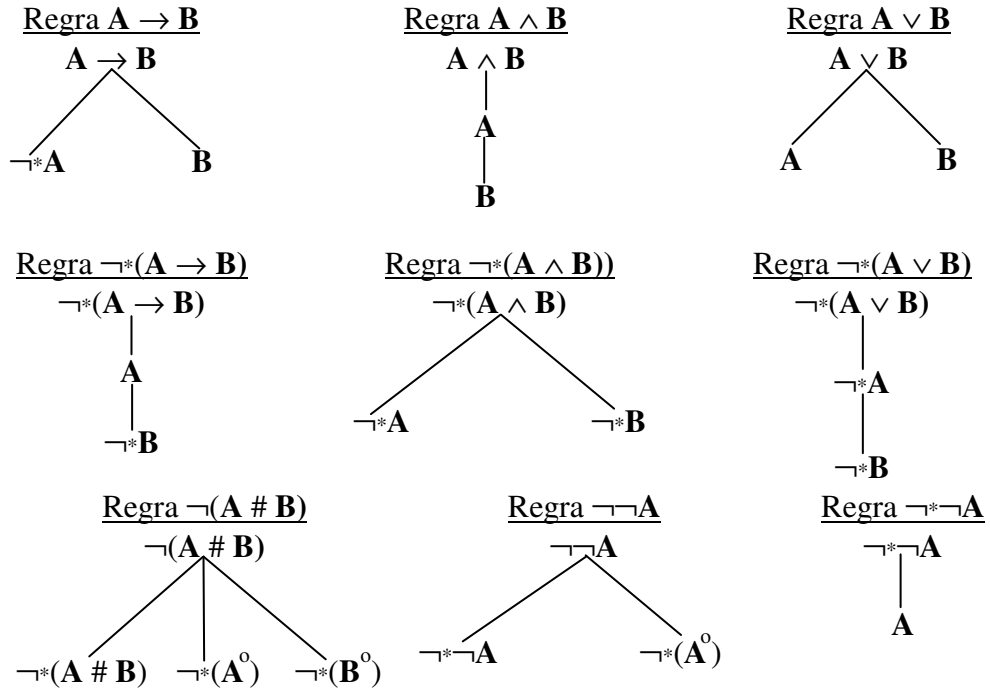
Para que um sistema de tableaux seja definido três coisas portanto devem ser fornecidas: um conjunto de regras de desenvolvimento, doravante denominadas simplesmente de "regras", um critério de fechamento de ramos, e uma função de inicialização.

Na apresentação dos nossos sistemas de tableaux utilizaremos uma outra abreviatura, também de praxe na literatura paraconsistente:

$$\neg^*\mathbf{A} \text{ para } \neg\mathbf{A} \wedge \mathbf{A}^o.$$

O símbolo "$\neg^*$" é designado negação forte e possui todas as propriedades da negação clássica. As abreviaturas $\mathbf{A}^o$ e $\neg^*\mathbf{A}$ são tão convenientes que são adotadas no próprio provador por nós implementado, capaz de manipular fórmulas que as contenham.

O critério de fechamento de ramos para $\mathbf{SC_1}$ é a ocorrência de um par complementar $\mathbf{A}$ e $\neg^*\mathbf{A}$ no ramo. A função de inicialização de $\mathbf{SC_1}$ faz corresponder a cada fórmula um tableau com um único nó contendo a negação forte da fórmula dada. As regras para $\mathbf{SC_1}$ são dadas a seguir. Usaremos o sinal "#" para representar qualquer um dos conectivos "$\rightarrow$", "$\wedge$" ou "$\vee$". O nome de cada regra indica o tipo de fórmulas para as quais a regra descrita é aplicável.



Sobre cada uma dessas regras pode ser demonstrado que, dada uma valoração que satisfaz **A**, essa valoração satisfaz a pelo menos um dos ramos da regra aplicável a **A**. Se chamarmos de satisfatível a um tableau com pelo menos um ramo no qual todas as fórmulas são simultaneamente satisfatíveis, podemos mostrar que, se **T** é um tableau satisfatível, então o tableau resultante do desenvolvimento de **T** por qualquer uma das regras descritas acima é também satisfatível. Podemos então mostrar que, se **A** é satisfatível, qualquer tableau para **A** também o é, ou, por outro lado, que a existência de um tableau insatisfatível para **A** acarreta a insatisfatibilidade de **A**. Agora, um tableau do sistema $\mathbf{SC_1}$ no qual todos os ramos são fechados é insatisfatível. $\mathbf{SC_1}$ é portanto consistente com respeito à semântica de $\mathbf{C_1}$, ou seja, se há um tableau para **A** (a raiz contém $\neg\mathbf{A}$) no qual todos os seus ramos são fechados, então **A** é logicamente válido em $\mathbf{C_1}$.

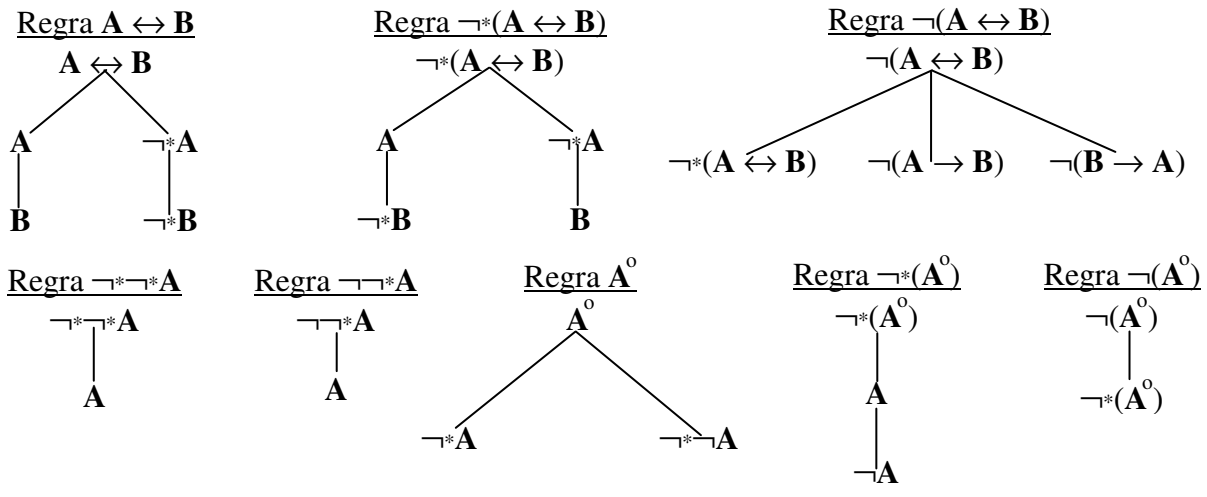
Podemos mostrar que todo ramo exaurido (um ramo no qual as únicas fórmulas não usadas são atômicas ou negações fortes de fórmulas atômicas) aberto de um tableau de $\mathbf{SC_1}$ é satisfatível. Como as regras de $\mathbf{SC_1}$ geram sempre fórmulas menos complexas do que as fórmulas às quais estas se aplicam, pode ser demonstrado também que há um tableau exaurido (um tableau no qual todos os seus ramos são exauridos) para toda fórmula. Dai, se **A** é uma fórmula insatisfatível e **T** é um tableau exaurido para **A**, então todos os ramos de **T** devem estar fechados, pois caso contrário **T** teria um ramo exaurido aberto, o qual seria satisfatível, e portanto A seria satisfatível. $\mathbf{SC_1}$ é portanto completo com respeito à semântica de $\mathbf{C_1}$, ou seja, se **A** é uma fórmula logicamente válida em $\mathbf{C_1}$, então há um tableau para **A** no qual todos os seus ramos estão fechados.

De qualquer modo que desenvolvamos o tableau inicial para uma fórmula dada, sempre iremos deparar com um tableau exaurido. Daí, para o cálculo paraconsistente $\mathbf{C_1}$, dispomos de um método que decide se uma dada fórmula é logicamente válida ou não. Provas detalhadas podem ser encontradas em Buchsbaum [88].

3. S'C_1: Um Sistema de Tableaux para Implementação

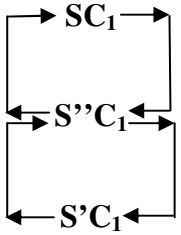
Nesta seção apresentamos um sistema de tableaux derivado de $\mathbf{SC_1}$, projetado de forma a permitir uma implementação mais eficiente. A idéia foi dotar $\mathbf{S'C_1}$ de um número maior de regras mais especializadas, de forma que as árvores geradas tendam a conter menos nós que aquelas de $\mathbf{SC_1}$. $\mathbf{S'C_1}$ contém regras novas com respeito a $\mathbf{SC_1}$, e regras de $\mathbf{SC_1}$ nas quais são ampliadas as restrições à sua aplicabilidade.

O critério de fechamento de $\mathbf{S'C_1}$ também foi modificado, passando ser o seguinte: são fechados ramos que contenham um par de fórmulas forma **A** e $\neg^*\mathbf{A}$ ou alguma fórmula de uma das formas $\neg^*((\mathbf{A} \wedge \neg\mathbf{A})^o)$, $\neg^*(\mathbf{A}^{oo})$, ou $\neg^*((\neg^*\mathbf{A})^o)$. As regras de $\mathbf{S'C_1}$ que diferem das de $\mathbf{SC_1}$ são as seguintes:



Demonstramos a equivalência de $\mathbf{S'C_1}$ e $\mathbf{SC_1}$ no sentido de que há uma confutação (um tableau cujos ramos estão todos fechados) de uma fórmula **A** em $\mathbf{S'C_1}$ se, e somente se, há uma confutação de **A** em $\mathbf{SC_1}$. Para isso definimos um sistema auxiliar $\mathbf{S''C_1}$ cujo critério de fechamento é o mesmo de $\mathbf{S'C_1}$ e cujas regras são a união das regras de $\mathbf{SC_1}$ e $\mathbf{S'C_1}$.

Mostramos então os resultados indicados no diagrama abaixo:



Lema 1: Se há uma confutação para **A** em $\mathbf{SC_1}$, então há uma confutação para **A** em $\mathbf{S''C_1}$.

Lema 2: Se há uma confutação para **A** em $\mathbf{S''C_1}$, então há uma confutação para **A** em $\mathbf{S'C_1}$.

Lema 3: Se há uma confutação para **A** em $\mathbf{S'C_1}$, então há uma confutação para **A** em $\mathbf{S''C_1}$.

Lema 4: Se há uma confutação para **A** em $\mathbf{S''C_1}$, então há uma confutação para **A** em $\mathbf{SC_1}$.

Novamente provas detalhadas podem ser encontradas em Buchsbaum [88].

Nesse ponto do projeto duas linhas de desenvolvimento são possíveis. Por um lado a obtenção de métodos de prova para variantes do cálculo $\mathbf{C_1}$. Embora tenhamos descrito aqui um método de tableaux proposicional, na realidade já. desenvolvemos uma versão (e um protótipo) para o cálculo de predicados paraconsistente $\mathbf{C_1^*}$ (vide Buchsbaum [88]). A passagem para a primeira ordem é em si um problema de interesse para cuja exposição não dispúnhamos de espaço adequado no presente artigo.

Uma próxima expansão de interesse seria a generalização do método para as famílias de cálculos $\mathbf{C_n}$ e $\mathbf{C_n^*}$.

Uma outra linha de desenvolvimento diz respeito ao aprimoramento do próprio provador. O protótipo desenvolvido não apresenta ainda todas as características desejáveis em um provador. Sua principal função é demonstrar a efetividade do método e servir como instrumento e laboratório para os desenvolvimentos subseqüentes. Assim, a sua arquitetura foi desenhada de forma a facilitar a obtenção de protótipos diferentes pela alteração ou inclusão de regras.

As principais alterações a serem suportadas pelo protótipo no sentido de torná-lo um provador razoável dizem respeito ao desenvolvimento de estratégias que o tornem mais eficiente e na construção de interfaces que o façam mais conveniente ao uso. Prevemos a sua utilização tanto como máquina de inferência, como por exemplo de um sistema baseado em conhecimento, quanto como um provador interativo.

Omitimos ainda, no presente trabalho, considerações sobre as nossas motivações para a construção do provador em questão, além da provocação que o problema em si colocava e da oportunidade de desenvolver a arte da construção de provadores. Isso conduziria à discussão da relevância da lógica paraconsistente para I.A., e novamente consideramos que não havia espaço suficiente para abordar esse problema devidamente. A questão é tratada em Pequeno [88].

Referências

da Costa, Newton C. A., "Ensaio sobre os Fundamentos da Lógica", Editora Hucitec e Editora da Universidade de São Paulo, São Paulo, 1980.

da Costa, Newton C. A., "On the theory of inconsistent formal systems", Notre Dame Journal of Formal Logic XV (1974), pp. 497-510.

da Costa, Newton C. A. & Alves, Elias H., "Une sémantique pour le calculi $\mathbf{C_1}$", C. R. Acad. Sc. Paris, 238 A (1977), pp. 729-731.

da Costa, Newton C. A. & Alves, Elias H., "A semantical analysis of the calculi $\mathbf{C_n}$", Notre Dame Journal of Formal Logic XVIII (1977), pp. 621-630.

Loparic, Andréa & Alves, Elias H., "The semantics of the systems $\mathbf{C_n}$ of da Costa", Anais da Terceira Conferência Brasileira de Lógica Matemática, © Sociedade Brasileira de Lógica, São Paulo, 1980, pp. 161-172.

da Costa, Newton C. A. & Wolf, Robert G., "Studies in Paraconsistent Logic II: Quantifiers and the Unity of Opposites", pré-publicação do Centro de Lógica, Epistemologia e História da Ciência, Universidade Estadual de Campinas.

Bunder, M. W., "On Arruda and da Costa's Logics $\mathbf{J_1}$ to $\mathbf{J_5}$", The Journal of Non-Classical Logic II-I (1983), pp. 43-48.

Kleene, S. C., "Introduction to Metamathematics", Van Nostrand, Princeton, 1952.

Buchsbaum, Arthur, "Um método automático de prova para a lógica paraconsistente", Dissertação de Mestrado, PUC/RJ, 1988.

Pequeno, Tarcísio, "Inconsistency and Non-Monotonicity in the Common-Sense Reasoning", Monografia do Departamento de Informática da PUC/RJ, 1988.