SOBRE O MÉTODO DOS TABLEAUX EM
LÓGICAS POLIVALENTES FINITÁRIAS

WALTER ALEXANDRE CARNIELLI

Orientador

Prof.Dr. Newton Carneiro Affonso da Costa

Tese apresentada ao Instituto de Matemática, Estatística e Ciência da Computação da Universidade Estadual de Campinas como requisito parcial para a obtenção do título de Doutor em Matemática.

Novembro - 1982

À memória de

Eduardo René Ferreira de Godoy,

"Dudu"

ÍNDICE

INTRODUÇÃO

SOBRE O MÉTODO DOS TABLEAUX SEMÂNTICOS EM LÓGICAS POLIVALENTES FINITÁRIAS

Dentre as diversas lógicas não clássicas, as lógicas polivalentes finitárias (isto é, aquelas que admitem um número finito n de valores de verdade, também chamadas lógicas n-valentes) talvez sejam as generalizações mais naturais da lógica clássica proposicional e da de primeira ordem, e provavelmente as que encontram um maior número de aplicações.

Podemos dizer que a lógica polivalente, como disciplina que estuda as diversas lógicas polivalentes, foi criada de modo independente por volta de 1920 por J. Łukasiewicz na Polônia e por E.M. Post nos Estados Unidos. Alguns autores (como Rescher [15]) citam N.A. Vasil'év (num artigo de 1921) como um dos lógicos que também contribuiu para a gênese da lógica polivalente, mas parece mais indicado considerá-lo como um dos precursores da lógica paraconsistente, como faz A.I. Arruda em [1] que o considera tão precursor da lógica polivalente quanto Aristóteles. De certa forma, Aristóteles pode ser considerado como um dos precursores da lógica polivalente, na medida em que, como sustentam alguns de seus comentadores, seu pensamento a respeito de certas proposições relativas aos futuros contingentes era de que tais proposições não poderiam ser nem verdadeiras, nem falsas, mas classificadas como potencialmente verdadeiras ou falsas. Assim, por exemplo, a proposição "haverá uma batalha naval amanhã" teria, pelo menos antes do evento, um valor de verdade indeterminado.

Este ponto de vista constituiu-se numa verdadeira motivação filosófica para que Łukasiewicz fosse levado ao estudo da lógica polivalente, pois a manutenção dos critérios ortodoxos da "lei da bivalência" ou "lei do terceiro excluido" nos comprometeria, no caso do futuro contingente, com alguma forma de determinismo. Assim, a lógica polivalente, na medida em que introduz vários valores de verdade, nos permite uma alternativa para os vínculos demasiado restritivos da lógica clássica.

Como observou Łukasiewicz, a lógica polivalente também poderia ser chamada de lógica não-crisipiana, pois, ao contrário de Aristóteles, Crisipo acreditava que todas as proposições deveriam ser necessariamente verdadeiras ou falsas, dentro de um esquema rigidamente determinista.

Apesar de, no princípio, ter sido cultivada em poucos centros, a lógica polivalente rapidamente se desenvolveu, tendo sido estudada por lógicos importantes, tais como, entre outros, A. Tarski, A. Heyting, K. Gödel, S.C. Kleene, S. Jáskowski, A.Church e L.E.J. Brouwer; como principais fontes de referência bibliográfica, remetemos o leitor às bibliografias compiladas por N. Rescher em [15], com trabalhos publicados até 1965, e por R. G. Wolf em [27], com trabalhos publicados de 1966 a 1974. Estes dois catálogos somam, em conjunto, cerca de 1.500 títulos.

Ao lado das aplicações da lógica polivalente em metamatemática, mecânica quântica, teoria de probabilidade e análise de circuitos eletrônicos, destacamos o volume editado por D. Rine [16], onde diversos artigos estudam o relacionamento entre teoria da computação e lógica polivalente, e o artigo de A. Rose [17], onde são discutidas duas aplicações, a nosso ver bastante sugestivas, de lógica polivalente a tabelas de confecção de horários (timetables) e ao uso eficiente de máquinas numa indústria.

Neste trabalho, tendo em vista a grande relevância atual da lógica polivalente, tencionamos mostrar como o método dos tableaux, estudados primeiramente para a lógica clássica por, principalmente, E.W. Beth [2] e R. Smullyan [24], pode ser aplicado às lógicas finitárias.

Dentro desse ponto de vista, S. Surma [25] é o único (de nosso conhecimento) estudo dos tableaux para lógicas polivalentes; o trabalho de Surma, contudo, demonstra apenas a completude do método dos tableaux para cálculos polivalentes proposicionais, com um único valor distinguido e utiliza uma definição de tableau que, por excesso de condições, praticamente trivializa as demonstrações.

O que faz com que as lógicas n-valentes sejam uma generalização natural da clássica, e que se constitui na linha mestra de nosso trabalho, é o fato de que, nestas lógicas, os valores de verdade de proposições mais complexas dependem funcionalmente dos valores de verdade de proposições elementares, através de composição algébrica de certas funções fixadas.

No caso das lógicas proposicionais, essa dependência se traduz em termos de relações entre certas classes de álgebras, através das composições das funções que descrevem os conectivos. No caso das lógicas de primeira ordem, para que esta funcionalidade continue a se manter, precisamos saber como se comporta (em função dos valores lógicos) uma fórmula com quantificadores em relação às fórmulas elementares; introduzimos, então, a noção de *quantificador de distribuição* (Cap. IV). Associada a cada quantificador, existe uma *função interpretação*, que mantém com os quantificadores uma conexão análoga à que as interpretações dos conectivos (*matrizes* ou *tábuas*) mantêm com os conectivos.

Dentro desse esquema, o trabalho se divide em cinco capítulos e dois apêndices, cujo conteúdo descrevemos a seguir.

No Capítulo I, Seção I.1, introduzimos as noções preliminares a respeito de linguagens proposicionais n-valentes; na Seção I.2, definimos os *tableaux proposicionais*, que são certas estruturas do tipo de árvore, determinadas por meio de regras de formação chamadas *regras de derivação do tableau*. Os vértices dessa árvore são *fórmulas assignadas*, isto é, fórmulas acompanhadas de um valor de verdade, hipotético ou determinado pelo tableau. Permitimo-nos o uso do anglicismo "assignação" e seus derivados (assignar, assignado) com o sentido de "assinalação" ou "sinalização".

No Capítulo II, desenvolvemos o método dos tableaux proposicionais, aplicando-o, na Seção II.1, para obter os teoremas de completude e compacidade para lógicas polivalentes, onde os valores de verdade são divididos em duas classes: a dos valores *distinguidos* (D) e a dos *não distinguidos* (ND).

A noção de *demonstração*, ou *prova*, que utilizamos, é similar à dos tableaux para a lógica clássica: uma certa fórmula é *teorema* quando se refutam todas as possibilidades de que esta fórmula não seja teorema, isto é, quando se refutam todas as possibilidades de que a fórmula possa assumir valores de verdade não distinguidos. Essa refutação á apresentada pelo próprio tableau, e nesse caso dizemos que o tableau é *fechado*.

Os tableaux proposicionais, então, se constituem em instrumentos de análise que *decidem* quais os valores que uma determinada fórmula pode assumir, independentemente da interpretação que a isso se possa dar; desse modo, pelo menos em princípio, seria possível utilizar os tableaux para um tipo de lógica polivalente proposicional infinitária (isto é, com um conjunto de valores de verdade infinito), ou ainda onde os valores de verdade fossem divididos em mais de duas classes; contudo, não abordamos neste trabalho esse tipo de problema.

No caso das lógicas polivalentes proposicionais finitárias, os tableaux são finitos, e o conjunto finito dos tableaux que, para uma dada fórmula, mostram a impossibilidade de que esta fórmula assuma valores não distinguidos, é a *demonstração* de que a fórmula em questão é um teorema.

Na Seção II.2, demonstramos uma forma do teorema da substituição e, na Seção II.3, estudamos uma versão polivalente do Princípio da Unificação, introduzido por R. Smullyan em [23]. O conteúdo matemático desse princípio não difere muito do conteúdo do teorema da completude, mas é enunciado de uma forma bem mais geral, envolvendo, em sua prova, as propriedades de caráter finito dos tableaux.

Na análise das lógicas polivalentes pelo método dos tableaux, prescinde-se dos axiomas e regras de derivação, mas seria interessante se pudéssemos dispor de um método que permitisse obter um conjunto de axiomas e regras de derivação a partir das matrizes, em relação ao qual a lógica fosse completa. Na Seção II.4, descrevemos uma versão dos tableaux proposicionais, que denominamos *tableaux sintéticos*, e que de certa forma funciona dessa maneira.

No Capítulo III, utilizamos resultados anteriores para responder à seguinte questão: dados dois cálculos proposicionais, respectivamente n-valente e m-valente (m e n iguais ou não), em que condições eles apresentam o mesmo conjunto de teoremas? Esta questão, ligada a outras do mesmo gênero, é discutida na Seção III.1 e na Seção III.2.

No Capítulo IV introduzimos as noções necessárias ao estudo das lógicas n-valentes de primeira ordem pelo ponto de vista dos tableaux. Na Seção IV.1, estabelecemos as definições preliminares, e na Seção IV.2, definimos as noções semânticas para linguagens n-valentes de primeira ordem introduzindo os quantificadores de distribuição, referidos no início desta introdução.

Queremos observar que nossa definição de quantificadores de distribuição, embora proposta como uma generalização direta dos quantificadores clássicos, como é mostrado na Seção IV.2, pode ser vista como um caso particular dos quantificadores de Rescher-

-Mostowski (ver [15]) da mesma forma como os quantificadores clássicos podem ser vistos como casos particulares dos quantificadores generalizados de Mostowski. Na Seção IV.3, introduzimos *os tableaux quantificacionais* e as noções de demonstração, consequência sintática, etc., de maneira análoga as do caso proposicional. Os tableaux quantificacionais, ao contrário dos proposicionais, não são necessariamente árvores finitas, pois na formação destas árvores podem ocorrer regras não finitárias. Na definição dos tableaux quantificacionais, contudo, está implícito um procedimento sistemático para que se possa manipular estas regras infinitárias de tal modo que pelo menos os tableaux fechados sejam finitos, e desse modo a noção de prova se torna finitária.

No Capítulo V desenvolvemos o método dos tableaux quantificacionais; nas Seções V.1, V.2 e V.3, obtemos a completude do método dos tableaux quantificacionais e estudamos algumas aplicações do mesmo; inclusive, na Seção V.2, tratamos de uma versão para primeira ordem do Princípio da Unificação desenvolvido na Seção II.3; obtemos, ainda, os teoremas de Löwenheim-Skolem e de compacidade.

No Apêndice I, investigamos um problema ligado aos quantificadores de distribuição. As questões que se colocam são as seguintes: 1) como se pode definir quantificadores a partir de outros, através dos conectivos; e 2) como se pode caracterizar aqueles quantificadores tais que a partir deles se pode definir todos os demais?

Esse problema envolve dificuldades algébricas e combinatórias consideráveis, análogas às dificuldades envolvidas no estudo da álgebra de composição das matrizes (ver [9], [13] e [21]).

No Apêndice II, consideramos, a título de ilustração, um exemplo de um procedimento computacional para um particular tableau sintético; apresentamos um programa em linguagem PASCAL que demonstra, automaticamente, teoremas do cálculo proposicional trivalente P^1 (ver [22]).

Por último, dedicamos uma seção especial a uma pequena lista de problemas em aberto, que consideramos relevantes.

Convém mencionar, por fim, alguns dos resultados paralelos obtidos durante o desenvolvimento desta tese, e que serão publicados oportunamente, (em coloboração com o Prof. Newton C.A. da Costa):

1) Pode-se desenvolver uma lógica polivalente de ordem superior, para ela estendendo-se toda a semântica "standard" e de Henkin, obtendo-se teoremas de correção, completude e incompletude como no caso bivalente.
2) Com base em lógica polivalente de ordem superior, pode-se construir lógicas paraconsistentes associadas, como se fez no caso da lógica de primeira ordem (ver [5]).
3) Pelo menos no caso das lógicas polivalentes finitárias pode-se construir teorias monádicas de ordem superior que sirvam como fundamento para teorias do conjunto polivalentes.
4) Por meio de sistemas polivalentes de ordem superior convenientes é possível construir alguns sistemas bastante fortes de lógica livre (ver [15]), incorporando-se inclusive uma teoria das descrições com características muito naturais.

Aos Professores Xavier Caicedo,da Universidade de Los Andes, Colômbia, Edgard G.K. Lópes - Escobar, da Universidade de Maryland, E.U.A., e Antonio Mário Sette, da Universidade de Campinas, meus agradecimentos pelas discussões que permitiram esclarecer alguns pontos importantes no presente trabalho.

Aos Professores do grupo de lógica e fundamentos do Departamento de Matemática da Universidade Estadual de Campinas e do Departamento de Matemática da Universidade de São Paulo, agradeço pelos ensinamentos que deles recebi, como aluno e colega de trabalho.

Ao Professor Newton C.A. da Costa, orientador de tese, meu agradecimento especial pelas lições recebidas, pela orientação da tese e pelo exemplo de atitude corajosa perante os problemas do conhecimento.

CAPITULO I

TABLEAUX PARA LÓGICAS PROPOSICIONAIS POLIVALENTES

I.1 - INTRODUÇÃO

Os métodos do tipo "tableaux analíticos" (cf. [24]), ou, ainda, "tableaux de Beth" [2] ou "tableaux de Hintikka", têm sido bastante estudados para as lógicas proposicionais e de primeira ordem clássicas (a dois valores), constituindo-se em ferramentas de análise muito elegantes e poderosas.

Em [25], S. Surma apresenta um método para a axiomatização finita de lógicas proposicionais n-valentes com um único valor distinguido, generalizando os tableaux analíticos de Smullyan [24]. Contudo, sua definição de tableau não é adequada, pois, por envolver excesso de condições, torna o teorema de completude (que é seu principal resultado) praticamente desprovido de conteúdo.

Nos dois primeiros capítulos deste trabalho, estudaremos as lógicas proposicionais n-valentes do ponto de vista dos tableaux, modificando as definições apresentadas em [25], permitindo derivar os principais teoremas da teoria das lógicas proposicionais n-valentes (tais como os teoremas de completude e de compacidade).

Por uma *linguagem proposicional* entendemos a seguinte estrutura:

L-1) Sejam $S_o = \{p_i : i \in \mathbb{N}\}$ um *conjunto de variáveis proposicionais* e $C = \{F_1, F_2, \ldots, F_r\}$ um *conjunto* (finito) *de conectivos proposicionais* m_i-*ários*; então:

L-2) A *linguagem proposicional* sobre o alfabeto $S_o \cup C$ é a álgebra abstrata S, livremente gerada pelo conjunto S_o, através dos conectivos de C, i.e.:

$$S_o = \{p_i : i \in \mathbb{N}\}$$

$$S_{r+1} = S_r \cup \{F_i(X_1, \ldots, X_{m_i}) : X_1, \ldots, X_{m_i} \in S_r\},$$

e

$$S = \cup\, \{S_r : r \in \mathbb{N}\}.$$

As fórmulas de $S_{r+1} - S_r$ são chamadas *fórmulas de nível* r+1 e as de S_o são chamadas, também, *fórmulas atômicas*.

Usaremos os seguintes resultados e definições a respeito de álgebras abstratas (cf. [14]):

Uma *álgebra abstrata de tipo finito* é um par $\langle S, \{O_i : 1 \leq i \leq r\}\rangle$, onde S é um conjunto não vazio e cada O_i é uma operação em S (i.e., uma função de $S \times S \times \ldots \times S = S^m$ em S).

Chamaremos de *aridade* da operação O_i ao natural m, e algumas vezes denotamos a álgebra pelo seu *universo* S (isto é, pelo conjunto S subjacente a ela).

Dadas as álgebras $\langle S, \{O_i : 1 \leq i \leq r\}\rangle$ e $\langle R, \{O'_i : 1 \leq i \leq r'\}\rangle$ diremos que são *similares* se r=r' e para cada $i \leq r$, O_i e O'_i têm a mesma aridade.

Dada a álgebra S, S' é *subálgebra* de S se $\langle S', \{O_i : 1 \leq i \leq r\}\rangle$ é uma álgebra, onde $S' \subseteq S$.

É fácil ver que a intersecção não vazia de uma classe de subálgebras de S é uma subálgebra de S; quando, para algum $S_o \subseteq S$, a intersecção das subálgebras que contêm S_o for a álgebra S, a álgebra S é dita *gerada* por S_o (e S_o é o *conjunto de geradores de* S).

Seja $\mathcal{A}$ uma classe de álgebras de mesma similaridade; uma álgebra S em $\mathcal{A}$ é chamada *livre na classe* $\mathcal{A}$ se ela contém um subconjunto S_o tal que S_o gera S e se toda função $\underline{h} : S_o \longrightarrow B$, onde B é uma álgebra qualquer de $\mathcal{A}$, pode ser estendida a um homomorfismo $h : S \longrightarrow B$.

Nesse caso, a álgebra S diz-se *livremente gerada* por S_o.

As provas das seguintes proposições podem ser encontradas, por exemplo, em [14]:

1) Se um homomorfismo h leva o conjunto de geradores de uma álgebra S no conjunto de geradores de uma álgebra B, então leva S em B.

2) Se S_o é o conjunto de geradores de uma álgebra S, e se uma função $\underline{h} : S_o \longrightarrow B$ pode ser estendido a um homomorfismo de S em B (onde B é uma álgebra), então esta extensão é única.

Sejam $N = \{0,1,2,\ldots,n-1\}$, $D \subseteq N$ e $A = \langle N, f_i : 1 \leq i \leq r\rangle$ uma álgebra abstrata de tipo finito de mesmo tipo de similaridade que a linguagem proposicional S (isto é, cada f_i é uma função $f_i : N^m \longrightarrow N$ e tem a mesma aridade do conectivo proposicional correspondente F_i); nesse caso, chamamos o par ordenado $\mathcal{A} = \langle A, D \rangle$ de *estrutura associada à linguagem proposicional* S ou *pré-modelo para* S.

O conjunto D é chamado de *conjunto de valores distinguidos* da estrutura $\mathcal{A}$.

Um homomorfismo v de S em A é chamado de *valorização proposicional*, e pelas observações anteriores fica claro que uma valorização proposicional está univocamente determinada se conhecermos os valores de v para cada conjunto de variáveis proposicionais S_o.

Se $n = |N|$ e $d = |D|$, denominamos o par ordenado $L = \langle S, \mathcal{A} \rangle$ de *lógica a* n *valores com* d *valores distinguidos* (ou lógica n-*valente* com d valores distinguidos).

Em particular, se os conectivos são todos unitários ou binários, as funções f_i podem ser descritas por matrizes (matrizes coluna ou matrizes quadradas). Pela facilidade de manuseio e por se constituirem na quase totalidade dos casos estudados usualmente, os exemplos do presente trabalho se restringirão a esses casos.

No que se segue, usaremos a seguinte convenção para referência aos elementos da linguagem (a menos de esclarecimento em contrário):

S, para denotar o conjunto de fórmulas da linguagem proposicional; $\mathcal{A}$, para o pré-modelo N, para o conjunto de valores, D, para o conjunto de valores distinguidos; A, para a álgebra similar a S; B,C,D (com ou sem índices), para subconjuntos de S; X,Y,Z (com ou sem índices), para elementos de S; v, para valorização proposicional; S_o, para o conjunto de fórmulas atômicas, N_o para subconjuntos de N, e ND para o subconjunto N-D (conjunto dos valores não distinguidos).

I.2 - TABLEAUX PROPOSICIONAIS.

Vamos considerar informalmente as noções de árvore, ramo, vértice, nível dos vértices, etc. (para detalhes remetemos o leitor a [8]).

Consideremos os seguintes símbolos, introduzidos em [25],

$$\frac{X}{Y_1 \wedge Y_2 \quad \ldots \wedge Y_m} \quad \text{e} \quad \frac{X}{Y_1 \vee Y_2 \vee \ldots \vee Y_m}$$

como denotando árvores dos seguintes tipos, respectivamente:

$$\begin{array}{c} X \\ | \\ Y_1 \\ | \\ Y_2 \\ | \\ \vdots \\ Y_m \end{array} \qquad \text{e} \qquad \begin{array}{c} X \\ \swarrow \;\; \downarrow \;\; \searrow \\ Y_1 \quad Y_2 \; \ldots \; Y_m \end{array}$$

e abreviaremos os símbolos acima pelas seguintes expressões, respectivamente :

$$\frac{X}{\wedge\{Y_i : i \leq m\}} \qquad \text{e} \qquad \frac{X}{\vee\{Y_i : i \leq m\}}$$

Fixemos, fora da linguagem de S, uma classe de símbolos a_i,

$0 \leq i \leq n-1$, e consideremos uma função que associa a cada $X \in S$ um símbolo a_i; se X está associado a a_i, escrevemos $a_i(X)$.

Em termos intuitivos, podemos encarar esses símbolos a_i como conectivos metalinguísticos, que afirmam: "X tem o valor de verdade de i". Chamaremos de *fórmulas assignadas* aos objetos $a_i(X)$, e denotaremos por S* o conjunto das fórmulas assignadas.

Seja $L = \langle S,A \rangle$ uma lógica proposicional fixada: definimos o *tableau proposicional* da fórmula $a_i(F(X_1,\ldots,X_m))$ como a árvore cujo primeiro vértice é a fórmula (assignada) , e cujos vértices próximos são determinados pela seguinte *regra de derivação do tableaux*.

S-1) $\dfrac{(\pi_1)}{(\pi_2)}$: $\dfrac{a_i(F(X_1,\ldots,X_m))}{\bigvee \{a_{j_1}(X_{i_1}) \wedge \ldots \wedge a_{j_t}(X_{i_t}) : j_1,\ldots,j_t < n}$ e vale a *condição proposicional* $H_i(F ; j_1,\ldots,j_t)\}$,

onde $H_i(F; j_1,\ldots,j_t)$ significa que existe um homomorfismo $h : S \longrightarrow A$, tal que: 1) $h(X_{i_k}) = j_k$ para $1 \leq k \leq t$, e 2) se f representa o conectivo F, então $f(v_1,\ldots,v_{i_1},\ldots,v_{i_2},\ldots,v_{i_t},\ldots,v_m) = i$ para toda substituição onde $v_{i_k} = j_k$ e os demais valores v_s são arbitrários.

Chamamos de *consequência de* $\pi_1/\pi_2(a_i(F(X_1,\ldots,X_m))$ a cada conjunto de vértices $\{a_{j_1}(X_{i_1}),\ldots,a_{j_t}(X_{i_t})\}$.

S-2) Os vértices são determinados pelo seguinte *procedimento sistemático:*

S-2.1) Começamos a árvore com a fórmula assignada $a_i(F(X_1,\ldots,X_m)$ (vértice 1);

S-2.2) Aplicamos o passo (S-1) à fórmula (regra π_1/π_2) e declaramos o vértice 1 *usado*;

S-2.3) Suponhamos que o vértice n tenha sido usado; se todo vértice onde ocorre uma fórmula não atômica já foi usado, o procedimento pára. Se não, tomamos como vértice n+1 o vértice de nível menor e que esteja mais à esquerda na árvore;

S-2.4) Adicionamos a cada último vértice de cada ramo que passa por n+1 a conclusão da regra π_1/π_2 para o vértice n+1 e o declaramos usado.

Devemos entender as fórmulas (assignadas) separadas pelo símbolo $\wedge$ como pertencentes ao mesmo ramo, e as separadas por V como pertencentes a ramos distintos: fica claro, também, pela definição de subfórmula e pela definição do tableau, que todo tableau proposicional para fórmulas proposicionais é finito.

Uma *subfórmula* de uma fórmula X é uma fórmula Y definida da seguinte maneira:

i) Se $X = F(Y_1,\ldots,Y_m)$ for uma fórmula, então $Y_1,\ldots,Y_m$ são *subfórmulas imediatas de* X;

ii) Subfórmulas imediatas são subfórmulas;

iii) Subfórmulas de subfórmulas de X são subfórmulas de X.

Dizemos que o tableau proposicional para a fórmula $a_k(X)$ é *fechado* se, para cada ramo, existe uma subfórmula Y de X tal que $a_i(Y)$ e $a_j(Y)$ ocorrem neste ramo com $i \neq j$, ou se S-1 não pode ser aplicado e X é não-atômica , e *aberto* caso contrário; dessa forma, se o tableau é uma árvore com um único vértice, ele será *fechado*, a menos que se trate do tableau para uma fórmula atômica; neste caso, o tableau com um único vértice é aberto.

Em termos intuitivos, o tableau de Surma para a fórmula $a_i(X)$ é fechado quando é possível afirmar que uma mesma fórmula (na verdade, uma subfórmula de X) assume valores distintos sob a mesma valorização proposicional, ou quando a fórmula X nunca assume o valor i; esta anomalia teria ocorrido por iniciarmos o tableau com $a_i(X)$. De qualquer forma, eliminamos a possibilidade de que X assuma o valor i.

Os teoremas seriam então as fórmulas tais que as possibilidades de assumir algum valor não distinguido são todas eliminadas.

Dada uma fórmula X e um valor k, aparentemente seria correto que nos referíssemos *aos tableaux proposicionais* para $a_k(X)$ ao invés de ao tableau, visto que as árvores resultantes podem não ser iguais; contudo, podemos assumir, na definição de tableau, que as árvores serão ordenadas da seguinte maneira:

1) As fórmulas separadas por $\wedge$ serão colocadas de cima para baixo segundo a ordem dos valores j_i;
2) Os ramos serão ordenados (da esquerda para a direita) pela ordem lexicográfica.

Dessa forma, é fácil mostrar (por indução, relativamente ao nível) que dados k e X, existe um único tableau para $a_k(X)$, e podemos nos referir *ao* tableau.

Definimos o tableau como *atomicamente fechado* quando cada ramo contém fórmulas $a_i(X)$ e $a_j(X)$, onde $i \neq j$ e X é uma fórmula atômica ou a regra S-1 não pode ser aplicada no ramo.

É fácil demonstrar, por indução sobre o nível de construção das fórmulas, que os conceitos de "tableau fechado" e "tableau atomicamente fechado" são equivalentes; usaremos o que for mais conveniente em cada situação:

COMENTÁRIO 1.1: Vamos considerar, como exemplo, aplicações dos tableaux proposicionais ao cálculo trivalente P_1 (ver [22]); onde os seguintes conectivos são definidos a partir de dois conectivos primitivos:

	$\neg$
0	2
1	0
2	0

$\rightarrow$	0	1	2
0	0	0	2
1	0	0	2
2	0	0	0

$\vee$	0	1	2
	0	0	0
	0	0	0
	0	0	2

&	0	1	2
	0	0	2
	0	0	2
	2	2	2

$N = \{0,1,2\}$

$D = \{0,1\}$

As regras π_1/π_2 para esse cálculo serão as seguintes (abreviando $\{a_i(X)\}$ por $a_i(X)$):

R-1) $\dfrac{a_o(\neg X)}{a_1(x) \vee a_2(x)}$

R-2) $\dfrac{a_2(\neg X)}{a_o(X)}$

R-3) $\dfrac{a_o(X \rightarrow Y)}{a_2(X) \vee a_o(Y) \vee a_1(Y)}$

R-4) $$\frac{a_2(X \to Y)}{\{a_o(X) \wedge a_2(Y)\} \vee \{a_1(X) \wedge a_2(Y)\}}$$

R-5) $$\frac{a_o(X \vee Y)}{a_o(X) \vee a_1(X) \vee a_o(Y) \vee a_1(Y)}$$

R-6) $$\frac{a_2(X \vee Y)}{\{a_2(X) \wedge a_2(Y)\}}$$

R-7) $$\frac{a_o(X \,\&\, Y)}{\{a_o(X) \wedge a_o(Y)\} \vee \{a_o(X) \wedge a_1(Y)\} \vee \{a_1(X) \wedge a_o(Y)\} \vee \{a_1(X) \wedge a_1(Y)\}}$$

R-8) $$\frac{a_2(X \,\&\, Y)}{a_2(X) \vee a_2(Y)}$$

Como exemplo, justificamos a regra R-3, a partir da definição de tableau, da seguinte maneira:

a) vale a condição proposicional $H_o(\to;2)$ para o ramo $a_2(X)$, pois $\to(2,v) = 0$ para qualquer valor de v entre $\{0,1,2\}$.

b) vale a condição proposicional $H_o(\to;0)$ para o ramo $a_o(Y)$, pois $\to(v,0) = 0$ para qualquer valor de v em $\{0,1,2\}$.

c) analogamente, vale a condição proposicional $H_o(\to;1)$ para o ramo $a_1(Y)$.

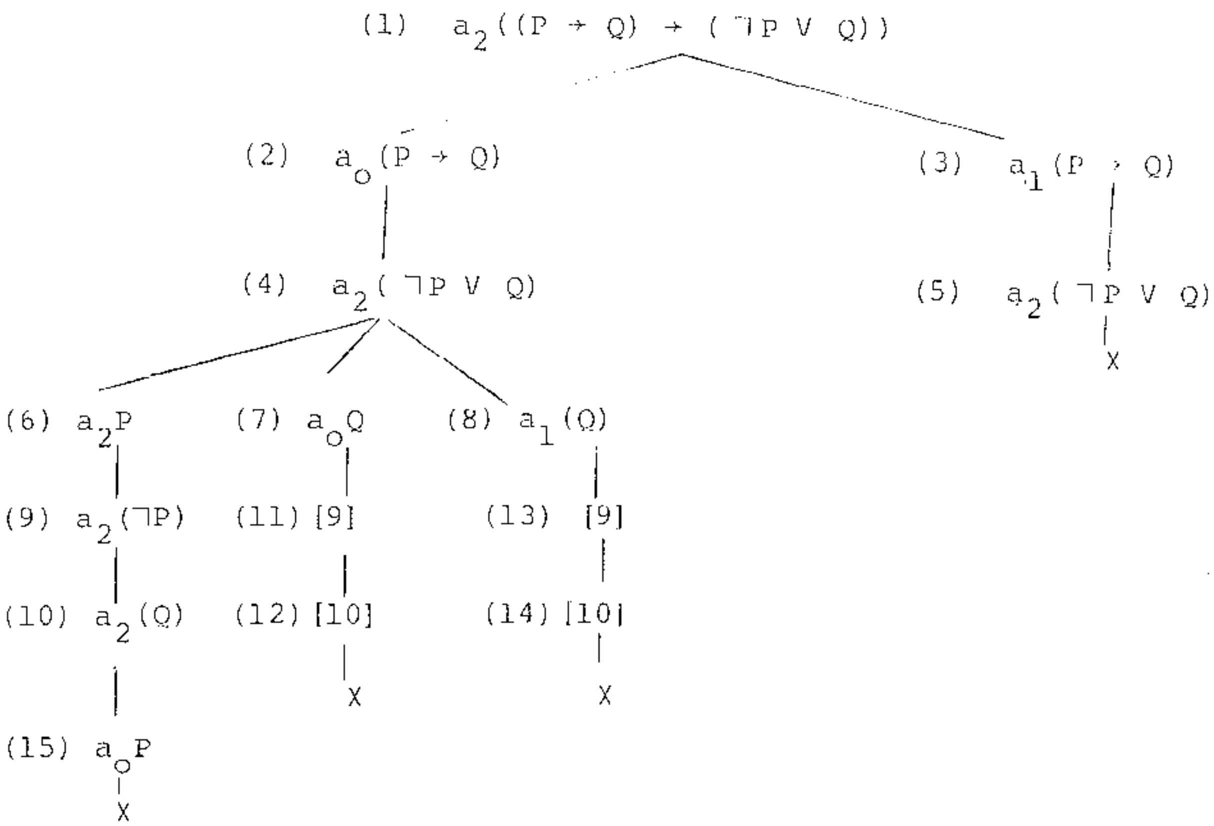
Sejam as seguintes fórmulas desse cálculo, onde P e Q são atômicas:

1) $(P \to Q) \to (\neg P \vee Q)$

2) $(P \,\&\, \neg P) \to Q$

Construiremos, a seguir, alguns dos tableaux possíveis para estas fórmulas, onde [i] significa repetição de vértice em (i) e X indica que o ramo está fechado; as regras de derivação, naturalmente, só serão aplicados a ramos abertos.

EXEMPLO 1:



1) Iniciamos o tableau com o vértice (1);

2) Os vertices (2) e (4) e (3) e (5) se obtêm de (1) por R-4; o vértice (1) está usado.

3) O vértice (2), que é o mais alto e mais à esquerda na árvore e que ainda não foi usado, dá origem aos vértices (6), (7) e (8); o vértice (2) está usado.

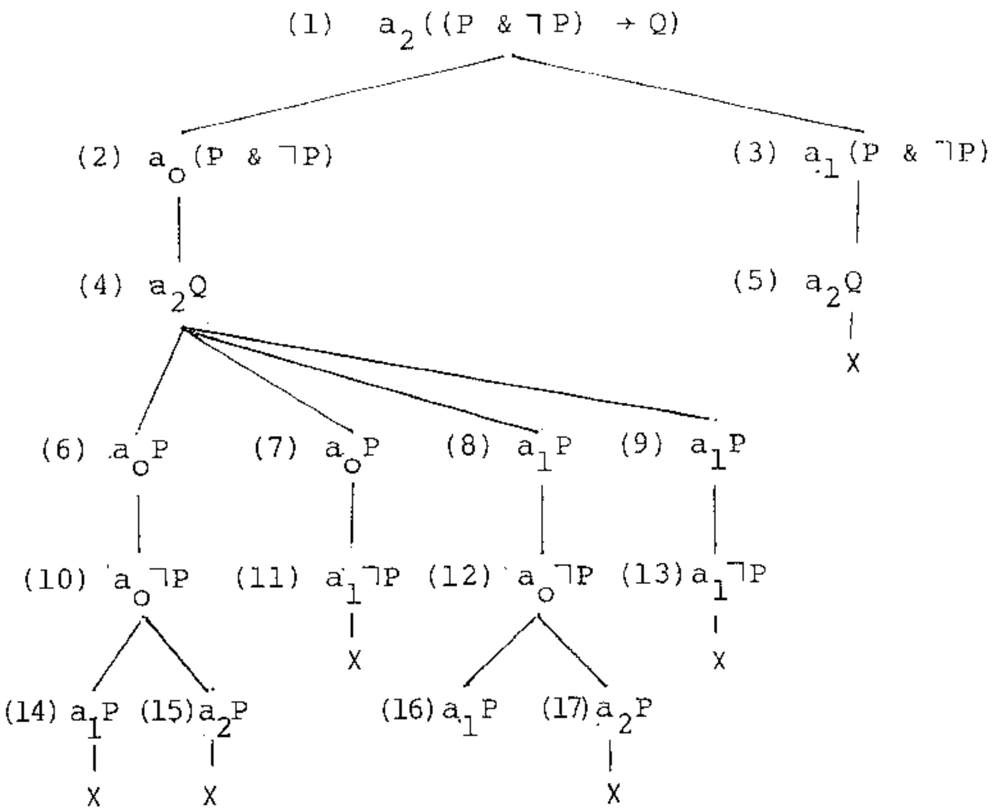
4) O próximo vértice a ser usado é (3); desde que não existe regra $\pi_1/\pi_2(a_1(P \rightarrow Q))$, o ramo que passa por (3) está fechado.

5) O vértice (4) é o próximo a ser usado; de acordo com o ítem S-2.4 da definição, adicionamos a cada último vértice de cada ramo que passa por (4) a conclusão da regra $\pi_1/\pi_2(a_2(\neg P \vee Q))$;

6) Temos, então, todos os ramos fechados, exceto o primeiro; o vértice (9) é o primeiro que ainda não foi usado e que não corresponde a uma fórmula atômica; obtemos, a partir dele, o vértice (15), e o tableau está fechado.

Como veremos no próximo parágrafo, o fato de o tableau para $a_2((P \rightarrow Q) \rightarrow (\neg P \vee Q))$ ser fechado significa que não existe valorização tal que $v((P \rightarrow Q) \rightarrow (\neg P \vee Q)) = 2$; se 2 é o único valor não distinguido do cálculo trivalente que estamos considerando, então esta fórmula é teorema, pois só assume valores distinguidos.

EXEMPLO 2: Vamos construir o tableau para $a_2(P \& \neg P) \to Q)$ e verificar que é aberto:



DESCRIÇÃO DOS PASSOS:

1) Iniciamos o tableau com o vértice (1) (S-2.1);

2) A partir de (1), obtemos (2) e (4), e (3) e (5) (S-2.2);

3) A partir de (2), obtemos (6) e (10), (7) e (11), (8) e (12), e (9) e (13) (S-2.3 e S-2.4);

4) O ramo que passa pelo vértice (3) é fechado, pois não existe regra $\pi_1/\pi_2(a_1(X \& Y))$;

5) De acordo com S-2.3, o vértice (10) é o próximo vértice a ser analisado, dando origem a (15) e (14). Analogamente, o vértice (12) dá origem a (16) e (17) (S-2.4);

6) Desde que o ramo que passa pelo vértice (16) é aberto, o tableau é aberto.

Queremos observar, como ilustra o exemplo acima, que a numeração dos vértices não coincidirá sempre com a numeração do procedimento sistemático (S-2.1 a S-2.4), uma vez que os vértices onde ocorrem fórmulas atômicas não são numerados pelo procedimento.

Seja $B = \{X_1, X_2, \ldots, X_s, \ldots,\}$ um conjunto de fórmulas (finito ou enumerável); definimos o *tableau proposicional para o conjunto* $\{a_{i_1}(X_1), \ldots, a_{i_s}(X_s), \ldots,\}$, da seguinte maneira:

T_1) Construa-se o tableau para $a_{i_1}(X_1)$;

T_2) Construido o tableau de $a_{i_s}(X_s)$, anexe-se ao último vértice de cada ramo o tableau para $a_{i_{s+1}}(X_{s+1})$;

T_3) Se não houver ramos abertos, termine o procedimento.

Denotamos por $a_i B$ o conjunto $\{a_i(X_1), \ldots, a_i(X_j), \ldots,\}$.

Queremos observar que a condição necessária, mas não suficiente, para que o tableau para um conjunto B seja aberto é que o tableau para cada elemento de B seja aberto, e que o fato do tableau para B ser aberto independe da ordem dos elementos de B; de fato, nesse caso, se o tableau para B fecha após a inclusão de n vértices por T_1 ou T_2, para qualquer outra ordem de B, esses vértices ocorrerão após um número finito m de passos. Se B for um conjunto infinito e o tableau para $a_i B$ for aberto, teremos uma árvore infinita. Nesse caso, desde que os tableaux são árvores finitamente geradas (isto é, cada vértice tem um número finito de sucessores) vale o

1.1. LEMA. (Lema de König). *Um tableau de Surma infinito tem pelo menos um ramo infinito.*

DEMONSTRAÇÃO. Ver ([8]). ∎

Nossa próxima etapa será estudar o relacionamento existente entre os possíveis valores que uma fórmula pode assumir sob uma valorização e o fechamento do tableau proposicional; desse modo, os teoremas 1.2 e 1.4 são os mais importantes do capítulo.

1.2. TEOREMA. *Seja* X *uma fórmula de* S; *se o tableau para* $a_i(X)$ *é fechado, então* X *não assume o valor* i *por nenhuma valorização proposicional.*

DEMONSTRAÇÃO. Se X é uma fórmula atômica, desde que nenhum tableau para uma fórmula atômica é fechado, o resultado é vacuamente satisfeito.

Se X não é uma fórmula atômica, suponhamos que X seja da forma $F(Y_1,\ldots,Y_m)$ e que exista uma valorização h tal que $h(X) = i$, $h(Y_k) = i_k$, para $1 \leq k \leq m$. Raciocinando por indução sobre o nível de construção das fórmulas, existe um tableau aberto para cada uma das subfórmulas $a_{i_k}(Y_k)$, $1 \leq k \leq m$.

Dado que h é uma valorização e que o valor de cada subfórmula Y_k é i_k, vale pelo menos a condição proposicional $H_i(F;i_1,\ldots,i_m)$; pela definição de tableau, portanto, existe alguma regra de derivação da forma $\dfrac{a_i\, F(Y_1 \ldots Y_m)}{\vee\{a_{j_1}(Y_{k_1}) \wedge \ldots \wedge a_{j_t}(Y_{k_t})\}}$, e o tableau para $a_i(F(Y_1,\ldots,Y_m))$ é aberto, pois desde que por hipótese de indução todos os tableaux para $a_{i_k}(Y_k)$ são abertos basta anexar ao primeiro vértice $a_i(X)$ os tableaux para $a_{j_r}(Y_{k_r})$, $1 \leq r \leq t$. ∎

Antes de demonstrarmos a recíproca do Teorema 1.2, necessitamos das seguintes definições e de uma versão do Lema de Hintikka similar à apresentada em [24].

DEFINIÇÃO. (Conjunto de Hintikka). Seja HT um conjunto de fórmulas assignadas, satisfazendo às seguintes condições:

C-1) Para todo $j,k \leq n$, $j \neq k$, se $a_j(Y) \in HT$ então $a_k(Y) \notin HT$, onde Y é uma variável proposicional;

e

C-2) Se $Y = a_i(F(X_1,\ldots,X_m)) \in HT$, então alguma consequência de $\pi_1/\pi_2(Y)$ está contida em HT.

A um tal conjunto HT damos o nome de *conjunto de Hintikka.*

Um conjunto de Hintikka $\overline{HT}$ será dito um *conjunto de Hintikka saturado* quando satisfizer às condições:

$\overline{C\text{-}1}$) Para toda variável proposicional Y existe $j < n$ tal que $a_j(Y) \in \overline{HT}$;

e

$\overline{C\text{-}2}$) Se $Y = a_i(F(X_1,\ldots,X_m))$, então $Y \in \overline{HT}$ see alguma consequência de $\pi_1/\pi_2(Y)$ está contida em $\overline{HT}$.

1.3. LEMA.(Hintikka). *Todo conjunto de Hintikka* HT *pode ser estendido a um conjunto de Hintikka saturado* $\overline{HT}$.

DEMONSTRAÇÃO.

Se HT é um conjunto de Hintikka, construimos um conjunto $\overline{HT}$ da seguinte maneira: consideremos

a) $HT_o = HT \cup \{a_i Y$: Y é uma variável proposicional que não ocorre em HT_o e $i \in N$ é um valor qualquer$\}$.

b) Se $G_1, G_2, \ldots, G_t, \ldots$, é uma enumeração das fórmulas não atômicas, definimos o conjunto $HT_{\alpha+1}$ como:

$$HT_{\alpha+1} = HT_\alpha \cup \{a_i(F(X_1,\ldots,X_m)),\ \text{se } G_\alpha = F(X_1,\ldots,X_m)$$

tal que $a_i(F(X_1,\ldots,X_m))$ não contrarie a condição C-1 e satisfaça a condição $\overline{C-2}\}$ ou $HT_{\alpha+1} = HT_\alpha$, se isto não é possível.

c) $\overline{HT} = \bigcup_\alpha HT_\alpha$.

Afirmamos que $\overline{HT}$ é um conjunto de Hintikka saturado; em primeiro lugar, $\overline{HT}$ é um conjunto de Hintikka, pois a cláusula a) da construção precedente garante que nenhuma variável proposicional ocorre em $\overline{HT}$ com assignações diferentes, isto é, vale a condição C-1. A condição C-2 é obviamente satisfeita.

A validade das condições $\overline{C-1}$ e $\overline{C-2}$ pode ser verificada por indução sobre o nível de construção das fórmulas. ∎

1.4. TEOREMA. (Recíproca do Teorema 1.2). *Se* $X \in S$ *não assume valor* i *por nenhuma valorização* h, *o tableau proposicional para* a_i (X) *é fechado.*

DEMONSTRAÇÃO. A idéia da demonstração é construir uma valorização onde X assuma o valor i_o, supondo que o tableau para a_{i_o} (X) seja aberto; esta valorização se obtém da seguinte maneira:

a) Se o tableau para $a_{i_o}(X)$ é aberto, existe pelo menos um ramo R aberto;

b) Se o ramo R é aberto, valem:

(i) Se $a_j(Y) \in R$, então $a_k(Y) \notin R$, $j \neq k$, $j,k \leq n$ onde Y é subfórmula de X;

(ii) Se $Y = a_k(F(Y_1,\ldots,Y_m)) \in R$, alguma conclusão de $\pi_1/\pi_2(Y)$ está contida em R.

c) R é, então, um conjunto de Hintikka, e pelo Lema 1.3, pode ser estendido a $\overline{R}$;

d) Seja u uma função definida como se segue:

$u(X) = i$ see $a_i(X) \in \overline{R}$, $X \in S_o$;

e) u é bem definida, pois $\overline{R}$ é um conjunto de Hintikka saturado, e u pode ser estendida a S(pois S é absolutamente livre); Dessa forma, $u(F(X_1,\ldots,X_m))=f(u(X_1),\ldots,u(X_m))$;

f) Vamos mostrar que, para uma fórmula qualquer $Y = F(X_1,\ldots,X_m)$ de S, $u(Y) = i$ see $a_iY \in \overline{R}$;

g) Se $u(F(X_1,\ldots,X_m) = i$, então $a_i(F(X_1,\ldots,X_m)) \in \overline{R}$.

Por indução, suponhamos que se $u(X_k) = i$, então $a_i(X_k) \in \overline{R}$ i.e., $a_{u(X_k)}(X_k) \in \overline{R}$. Se $a_i(F(X_1,\ldots,X_m) \notin \overline{R}$ e $u(F(X_1,\ldots,X_m))= = i$, vale $H_i(F, u(X_1),\ldots,u(X_m))$ e, daí, como $\overline{R}$ é saturado, é fácil ver que existe pelo menos um X_k tal que

$$a_{u(X_k)}(X_k) \notin \overline{R} \text{ , o que é absurdo.}$$

h) Se $a_i(F(X_1,\ldots,X_m)) \in \overline{R}$, então $u(F(X_1,\ldots,X_m) = i$.
Se $a_i(F(X_1,\ldots,X_m)) \in \overline{R}$, desde que $\overline{R}$ é um conjunto de Hintikka, temos que vale a propriedade $H_i(F,j_1,\ldots,j_t)$ para alguns $j_1,\ldots,j_t$, e para todo $1 \leq k \leq t$,

$$a_{j_k}(X_{i_k}) \in \overline{R}.$$

Assumindo, por hipótese de indução, que se $a_{j_k}(X_{i_k}) \in \overline{R}$, então $u(X_{i_k}) = j_k$, por definição da propriedade $H_i(F, j_1,\ldots,j_t)$ temos: $f(u(X_1),\ldots,u(X_m)) = i$, isto é,

$$u(F(X_1,\ldots,X_m)) = i \quad ;$$

i) Se, por hipótese, o tableau para $a_{i_o}(X)$ é aberto, por (a), (b) e (c) temos que $a_{i_o}(X) \in \overline{R}$ e, por h), que $u(X) = i_o$, e obtemos a valorização procurada; a prova, então, está completa. ∎

O próximo resultado, cuja idéia central foi usada no teorema 1.4, generaliza a noção clássica de que uma valorização v e seu "conjunto verdade" (i.e., as fórmulas que assumem valor distinguido sob v) são interdefiníveis.

1.5. TEOREMA. *Dado um conjunto de Hintikka saturado* $\overline{HT}$, *existe uma única valorização proposicional associada a* $\overline{HT}$, *e reciprocamente.*

DEMONSTRAÇÃO. Dado $\overline{HT}$, pela condição $\overline{C\text{-}1}$, para cada fórmula X de S_o, exatamente uma fórmula assignada $a_i(X)$ pertence a $\overline{HT}$; tomamos a seguinte função v de S_o em N:

$$v(X) = i \quad \text{see} \quad a_i(X) \in \overline{HT}, \text{ onde } X \text{ é uma variável proposicional.}$$

A condição $\overline{C\text{-}1}$ garante que v é uma função bem definida para o conjunto de geradores S_o da álgebra S ; logo, v pode ser estendida a um homomorfismo v de S em A; este homomorfismo é único.

Reciprocamente, dada a valorização proposicional v, definimos $\overline{HT}$ como se segue:

Para cada $X \in S$, $a_i(X) \in \overline{HT}$ see $v(X) = i$ e $\overline{HT}$ é, claramente, um conjunto de Hintikka saturado. ∎

Dizemos que uma fórmula X é i-*valorizável* se existe uma valorização proposicional v tal que $v(X) = i$.

1.6. COROLÁRIO. *Dada* $X \in S$, X *é* i-*valorizável see o tableau proposi* *cional para* $a_i(X)$ *é aberto; e dado* $B \subset S$, *existe uma valorização proposicional* h *para* B *tal que* $h(X)=a_{i(X)}(X)$, *para* $X \in B$, *see existe um tableau aberto para o conjunto* $\{a_{i(X)}(X) : X \in B\}$.

DEMONSTRAÇÃO. A primeira parte é imediata, após os teoremas 1.2 e 1.4. Para a segunda parte, se existe uma valorização h para B, então para cada valor de verdade h(X), o tableau para a fórmula $a_{h(X)}(X)$ é aberto. Suponhamos que o tableau para o conjunto $\{a_{h(X)}(X) : X \in B\}$ feche; nesse caso, uma fórmula de B (ou subfórmula de fórmula de B) teria ocorrido no tableau com assignações distintas, e é fácil ver, por indução no nível de construção de fórmulas, que isto é impossível, desde que h é uma valorização.

Por outro lado, se há um tableau aberto para o conjunto $\{a_{i(X)}(X): X \in B\}$, este tableau tem pelo menos um ramo aberto (se B é finito, este fato é óbvio; se B é infinito e o tableau é aberto, então o próprio tableau é uma árvore infinita, e pelo Lema 1.1 tem pelo menos um ramo infinito, logo aberto , pela cláusula T-3 da definição de tableau para um conjunto.

Este ramo aberto é um conjunto de Hintikka, e pelo Lema 1.3 pode ser estendido a um conjunto de Hintikka saturado. Pelo teorema 1.5, existe a valorização indicada. ∎

CAPÍTULO II

APLICAÇÕES DOS TABLEAUX PROPOSICIONAIS

II.1 - TEOREMAS DE COMPLETUDE E COMPACIDADE

Pelo Corolário 1.6, temos um mètodo mecânico para verificar, dada uma fórmula X, quais são os valores (dentre $0,1,\ldots,n-1$) que são proibidos a esta fórmula por alguma valorização proposicional; estes valores proibidos são exatamente aqueles índices i para os quais o tableau proposicional para $a_i(X)$ é fechado. No cálculo trivalente considerado no Comentário 1.1 (Seção I.2), por exemplo, o valor "1" é proibido para toda fórmula dos tipos, $\neg X$, $X \to Y$, $X \vee Y$ e $X \& Y$, mas não para fórmulas atômicas, uma vez que tableaux com um único vértice são abertos se a fórmula é atômica (ver definição de tableau proposicional, Seção I.2).

Vamos, agora, definir as noções de *satisfação*, *consequência sintática*, *demonstração* e *consequência semântica*, e obter o teorema de completude.

Dada a lógica $L = \langle S, A \rangle$ e dado $X \in S$, dizemos que X é *demonstrável* (ou é *teorema*) em L se, para todo $i \in N_D$, os tableaux proposicionais para $a_i(X)$ são fechados; dizemos que o conjunto dos tableaux fechados é uma *demonstração* para X. Denotamos o fato de X ser teorema em L por $\vdash_L X$.

Para definir *consequência sintática*, necessitamos da seguinte definição a respeito dos tableaux: se B for um conjunto de fórmulas e N_o um subconjunto de N, chamamos um *tableau para o conjunto* $a_{N_o}(B)$ a um tableau para um conjunto $\{a_{i_1}(X_1), a_{i_2}(X_2), \ldots, a_{i_n}(X_n), \ldots,\}$, onde $X_j \in B$ e $i_j \in N_o$. (ver Seção I.2 para a definição de "tableau para um conjunto").

Observamos que a notação $a_{N_o}(B)$ é ambígua, pois serve para denotar todos os tableaux para conjuntos com fórmulas em B e índices em N_o; no entanto, na maior parte dos casos, essa ambiguidade não trará problemas: quando necessário, esclarecemos o índice com que cada fórmula de B comparece no tableau.

Dizemos que X é *consequência sintática* de B, onde $B \cup \{X\} \subset S$, se para cada tableau aberto para um conjunto $a_D(B)$ (isto é, para cada tableau aberto cujos vértices são fórmulas de B assignadas por valores distinguidos), o tableau para $a_D(B) \cup \{a_i(X)\}$ é fechado, para todo $i \in ND$.

Denotamos por $B \vdash_L X$ o fato de X ser consequência sintática de B.

Dada a lógica $L = \langle S, A\rangle$ e $X \in S$, dizemos que X é *satisfatível* em A, ou que v *satisfaz* X, se existe uma valorização proposicional v tal que $v(X) \in D$; dizemos que X é uma *tautologia* se for satisfeita para toda valorização proposicional.

Dados $B \cup \{X\} \subset S$, dizemos que X é *consequência semântica* de B se, para cada valorização proposicional v, X é satisfeita por v quando os elementos B forem satisfeitos por v.

O corolário 1.6 permite uma forma fraca do teorema de Completude, relacionando as noções de demonstração e satisfação definidas acima; no teorema abaixo, obtemos uma forma forte do teorema de completude:

2.1. TEOREMA. (Completude, forma forte). *Sejam* $X \in S$ *e* $B \subset S$; *então*, $B \vdash_{L} X$ see $B \models_{L} X$.

DEMONSTRAÇÃO. Suponhamos que $B \vdash_{L} X$; se v é uma valorização que a cada $B_j \in B$ assigna um valor distinguido i_j, então existe um tableau aberto para o conjunto $K = \{a_{j_1}(B_1), \ldots, a_{j_n}(B_n), \ldots\}$. Se $v(X) = i_o$ é um valor não distinguido, pelo colorário 1.6 é claro que o tableau para o conjunto $K \cup \{a_{i_o}(X)\}$ é aberto, contrariando a definição de consequência sintática. Portanto, $v(X) \in D$, e $B \models_{L} X$.

Por outro lado, se $B \models_{L} X$, suponhamos que existe um tableau aberto para $a_D(B)$; se existir algum valor i_o não distinguido tal que o tableau para $a_D(B) \cup \{a_{i_o}(X)\}$ seja aberto, pelo corolário 1.6 teremos uma valorização v tal que $v(B) \in D$ e $v(X) \notin D$, contrariando a definição de $B \models_{L} X$; logo $B \vdash_{L} X$. ■

COMENTÁRIO 2.1. É fácil ver que $\vdash_{L}$ satisfaz as propriedades estruturais do símbolo clássico $\vdash$ (leis de Gentzen); as demonstrações, envolvendo apenas as propriedades dos tableaux, são rotineiras:

(1) $X \vdash_{L} X$

(2) Se $B \vdash_{L} X$, então $B,A \vdash_{L} X$

(3) Se $B,Y,Y \vdash_{L} X$, então $B,Y \vdash_{L} X$

(4) Se $A,Y,Z \vdash_{L} X$, então $A,Z,Y \vdash_{L} X$

(5) Se $A \vdash_{L} Y$ e $Y \vdash_{L} X$, então $A \vdash_{L} X$

Para a prova do teorema de compacidade necessitamos o seguinte:

2.2. LEMA. (Propriedade de caráter finito dos tableaux): *Seja $B \subset S$, B finito ou enumerável; se $N_o \subset N$, então existe um tableau aberto para algum conjunto $a_{N_o}(B)$ see para todo subconjunto finito B_o de B existe um tableau aberto para algum conjunto $a_{N_o}(B_o)$.*

DEMONSTRAÇÃO. Seja $a_{N_o}(B) = \{a_i(X) : X \subset B \text{ e } i \in N_o\}$; se existe um tableau aberto para $a_{N_o}(B)$, seja $B_o = \{X_1,\ldots,X_r\} \subset B$ e sejam $a_{i_1}(X_1),\ldots,a_{i_r}(X_r)$ as assignações com as quais os elementos de B_o comparacem no tableau para $a_{N_o}(B)$.

Pela definição de tableau para um conjunto (Seção I.2), é óbvio que o tableau para o conjunto $\{a_{i_1}(X_1),\ldots,a_{i_r}(X_r)\}$ é aberto, e $i_1,\ldots,i_r$ pertencem a N_o.

Por outro lado, se para todo subconjunto finito B_o de B existe um tableau aberto para algum $a_{N_o}(B_o)$, seja $X_1,X_2,\ldots,X_n,\ldots$, uma enumeração de B; consideremos os subconjuntos finitos de B da forma

$$B_n = \{X_1,\ldots,X_n\}.$$

Suponhamos que B é enumerável (o caso B finito é óbvio a partir deste); construímos, então, um tableau infinito, indutivamente, como segue:

1) Iniciamos com um tableau aberto para $a_{N_o}(B_1)$;

2) Se temos um tableau aberto para $a_{N_o}(B)$, acrescentamos a cada ramo aberto deste o tableau para $a_{N_o}(X_{n+1})$, se o tableau resultante for aberto; em caso contrário, substituímos todo o tableau obtido por um tableau aberto $a_{N_o}(B_{n+1})$, que existe, por hipótese.

Desse modo, conseguimos um tableau aberto e infinito com fórmulas assignadas em N_o; pelo Lema de Konig (Lema 1.1), existe pelo menos um ramo aberto, e este ramo obviamente é um tableau aberto para $a_{N_o}(B)$. ∎

Pelo teorema da Completude e pelo lema 2.2, podemos obter, então, duas formas do teorema de Compacidade:

2.3. TEOREMA. (Compacidade, 1ª forma): *Seja* $B \cup \{X\} \subset S$; *então*, $B \models_L X$ *see existe* B_o *finito contido em* B *tal que* $B_o \models_L X$.

DEMONSTRAÇÃO. Se para cada B_o finito contido em B existe uma valorização proposicional v tal que $v(Y) \in D$ (para todo $Y \in B$) e $v(X) \notin D$, então usando o corolário 1.6 e o lema 2.2, temos que existe um tableau aberto para o conjunto

$$\{a_i(Y): Y \in B_o \text{ e } i \in D\} \cup \{a_i(X): i \in N - D\}.$$

Aplicando a construção do lema 2.2, obteremos um tableau aberto para o conjunto $\{a_i(Z): Z \in B \text{ e } i \in D\} \cup \{a_i(X): i \in N - D\}$ e, daí, existe uma valorização proposicional tal que $v(Z) \in D$, ($D \in \in B$) e $v(X) \notin D$; demonstramos, portanto, que se $B \models_L X$, então existe $B_o \subset B$ finito tal que $B_o \vdash X$.

A recíproca é óbvia. ∎

2.4. COROLÁRIO. (Compacidade, $2^{\underline{a}}$ forma). *Seja* $B \cup \{X\} \subset S$; *então*, $B \vdash_L X$ *see existe* $B_o \subset B$, B_o *finito, tal que* $B_o \vdash_L X$.

DEMONSTRAÇÃO. Pelo teorema anterior e pelo teorema da completude. ∎

OBSERVAÇÃO. Pode-se verificar facilmente que se definirmos o conjunto $Cn(B) = \{X : B \vdash_L X\}$, onde $B \cup \{X\} \subset S$, Cn é um *operador de fecho*, no sentido de Tarski [14]; isto é,

F-1) Para todo $B, C \subset S$, $B \subset C$ implica $Cn(B) \subset Cn(C)$;

F-2) Para todo $B \subset S$, $B \subset Cn(B)$;

F-3) $Cn(Cn(B)) = Cn(B)$, para todo $B \subset S$.

Dizemos que T é uma *teoria não trivial maximal* se $T \subset S$, $Cn(T) \neq S$ e se $T \subset T'$, então $T' = T$ ou $T' = S$.

Vale, então, o seguinte resultado:

2.5. TEOREMA. *Seja* $B \subset S$: *então* $Cn(B) = \cap \{T \subset S : B \subset T$ *e* T *é uma teoria não trivial maximal*$\}$.

DEMONSTRAÇÃO. Por um lado, se $X \in Cn(B)$, $X \in T$, para todo T na definição do lado direito da igualdade, pois se existe T tal que $X \notin T$, então $T \cup \{X\} \supset T$ e como T é maximal, $T \cup \{X\} = T$ ou $T \cup \{X\} = S$.

Se $T \cup \{X\} = S$, desde que $B \vdash_L X$, temos que $T \vdash_L X$ e $Cn(T) = S$, o que é impossível desde que $Cn(T) \neq S$; logo, $X \in T$.

Por outro lado, se X pertence à intersecção das teorias não-triviais maximais que contém B, e se $B \vdash_L X$, podemos estender B a uma teoria maximal não trivial (por um processo similar à de-demonstração clássica do teorema de Lindenbaum):

Tomando uma enumeração da fórmula, definimos:

$$B_1 = B \quad \text{e}$$

$$B_{n+1} = B_n \cup \{Y_n\} \quad \text{se} \quad B_n \cup \{Y_n\} \nvdash_L X$$

ou

$$B_{n+1} = B_n \quad \text{caso contrário}$$

Tomando $\overline{B} = \bigcup_{n \in \mathbb{N}} B_n$ temos que $\overline{B} \supset B$ e $\overline{B}$ é uma teoria não trivial maximal, logo $X \in \overline{B}$, absurdo. ∎

II.2 - TEOREMA DE SUBSTITUIÇÃO

Daqui para diante, para amenizar a notação, passaremos a considerar somente conectivos binários ou unários, embora a generalização para o caso dos conectivos m-ários seja imediata.

Se S é uma linguagem proposicional n-valente e S_o é o conjunto de fórmulas atômicas de S, chamamos de *pré-substituição* a toda função $s : S_o \longrightarrow S$.

Desde que S é uma álgebra livremente gerada por S_o (ver Seção I.1), a função s pode ser estendida a um único homomorfismo Sb_s: S $\longrightarrow$ S; chamamos de *substituição relativa a* s a este homomorfismo SB_s. Desse modo, temos, por exemplo, que se F é um conectivo binário em S e G é um conectivo unário, então:

$$Sb_s(F(X,Y)) = F(SB_s(X) , Sb_s(Y)) \quad e$$

$$Sb_s(G(X)) = G(Sb_s(X)).$$

Nesse caso, se $Sb_s(X) = X_1$ e $Sb_s(Y) = Y_1$, dizemos que $F(X_1,Y_1)$ é o resultado da substituição em F(X,Y) de X por X_1 e de Y por Y_1; analogamente, $G(X_1)$ é o resultado da substituição em G(X) de X por X_1.

O teorema da substituição, que demostraremos a seguir, garante que, se uma fórmula for teorema em $L = \langle S,A \rangle$, então qualquer fórmula obtida desta como resultado de uma substituição também será teorema.

2.6. TEOREMA (Teorema da Substituição). *Seja* Sb_s *uma substituição relativa a* s, X *uma fórmula em* S, $L=\langle S,A\rangle$ *uma lógica proposicional* n-*valente; se* $\vdash_L X$, *então* $\vdash_L Sb_s(X)$.

DEMONSTRAÇÃO. Seja v uma valorização proposicional qualquer: então, $\overline{v} = v \circ Sb_s : S \longrightarrow S$ também é uma valorização proposicional, pois:

$\bar{v}(X) = v(s(X))$, se X é atômica, e

$\bar{v}(F(X,Y)) = v(Sb_s(F(X,Y))$, se F é um conectivo binário, e

$\bar{v}(G(X)) = v(Sb_s(G(X))$, se G é um conectivo unário.

Desde que v é uma valorização, tem-se:

$$\bar{v}(F(X,Y)) = v(Sb_s(F(X,Y)) = v(F(Sb_s(X),\ Sb_s(Y)) =$$

$= f(v(Sb_s(X)),\ v(Sb_s(Y))) = f(\bar{v}(X),\ \bar{v}(Y))$, por indução sobre o nível de construção das fórmulas, e $\bar{v}$ é uma valorização proposicional (o caso dos conectivos unários é análogo).

Se $Sb_s(X)$ não é um teorema de L, então, pelo teorema de completude, existe uma valorização proposicional v tal que $v(Sb_s(X)) \notin D$. Desde que $v(Sb_s(X)) = (v \circ Sb_s)(X) = \bar{v}(X)$ e $\bar{v}$ é uma valorização, isto implica que $\bar{v}(X) \notin D$, e, pelo teorema de completude, X não é teorema em L. ∎

II.3 - PRINCÍPIO DA UNIFICAÇÃO PARA LÓGICAS PROPOSICIONAIS

O Princípio da Unificação para o cálculo de predicados clássico de 1ª ordem foi introduzido por R. Smullyan (ver [23]) e [24]) e permite obter simultaneamente diversos resultados da teoria clássica da quantificação (como teoremas de completude, compacidade, Löwenheim-Skolen e da interpolação).

No que se segue, vamos estudar uma generalização desse princípio para as linguagens proposicionais n-valentes (no Capítulo IV, estudaremos uma versão desse princípio para linguagens n-valentes de primeira ordem).

Seja Γ um conjunto de conjuntos de fórmulas assignadas tal que Γ tem *caráter finito* (isto é, B pertence a Γ see todo subconjunto finito B_o de B pertence a Γ). Se $K \in \Gamma$, dizemos que K *tem a propriedade* Γ. Se B é um conjunto de fórmulas não assignadas (i.e., $B \subset S$), dizemos que B é Γ-*consistente* se todas as fórmulas de B ocorrem num elemento K de Γ com assignações por valores distinguidos (isto é, $X \in B$ implica $a_i(X) \in K$, com $i \in D$).

Dizemos que Γ é uma *propriedade de consistência analítica proposicional* (abreviadamente, PCAP) quando todo conjunto K que tem a propriedade Γ satisfaz às seguintes condições:

P_o) K não contém $a_i(X)$ e $a_j(X)$, se $i \neq j$ e X é uma fórmula atômica de S;

P_1) Se $Y = a_i F(Y_1, Y_2) \subset K$, existe alguma conclusão C de $a_1/a_2(Y)$ tal que $K \cup C$ tem a propriedade Γ.

A expressão "B é Γ- consistente" deve ser entendida, informalmente, como significando que Γ garante que B é consistente, isto é, B é satisfatível ou tem modelo; o teorema 2.7 deixa claro este ponto.

COMENTÁRIO 2.2. Em termos intuitivos, Γ é uma PCAP quando todos seus elementos são "consistentes", isto é, não ocorrem fórmulas atômicas com assignações distintas nos elementos de Γ, e se uma certa fórmula assignada ocorre num elemento de Γ, então ocorrem nos elementos de Γ subfórmulas que garantem a presença dessa fórmula em Γ.

2.7. TEOREMA. (Princípio da Unificação para lógicas proposicionais n-valentes). *Seja Γ uma PCAP, e $B \subset S$; se B é Γ-consistente, então todos os elementos de B são satisfatíveis.*

DEMONSTRAÇÃO. Desde que o conjunto S^* de todas as fórmulas assignadas é enumerável, seja $Z_1, Z_2, \ldots, Z_n, \ldots$, uma enumeração de S^*; construimos, então, indutivamente, conjuntos C_n da seguinte maneira:

$C_1 = a_D(B)$ tal que $a_D(B) \in \Gamma$, (que existe pelo fato de B ser Γ-consistente).

$C_{n+1} = C_n \cup \{Z_n\}$ se $C_n \cup \{Z_n\}$ tem a propriedade Γ,

ou

$C_{n+1} = C_n$ em caso contrário.

É claro que os conjuntos C_n construidos desse modo tem a propriedade Γ, e que $C_1 \subset C_2 \subset \ldots \subset C_{n+1} \ldots$.

Seja $M = \cup \{C_i : i \in \mathbb{N}\}$; vamos mostrar que M é um conjunto de Hintikka saturado:

1) M tem a propriedade Γ, pois para cada $K \subset M$, K finito, existe C_n tal que $K \subset C_n$; como C_n tem a propriedade Γ, então K tem a propriedade Γ, pois Γ tem caráter finito.

Portanto, como cada subconjunto finito K de M tem a propriedade Γ e Γ tem caráter finito, M tem a propriedade Γ.

2) Se $M \cup \{Z_i\}$ tem a propriedade Γ, então $Z_i \in M$; de fato, nesse caso, o conjunto $C_i \cup \{Z_i\}$ tem a propriedade Γ, pois é um subconjunto de $M \cup \{Z_i\}$; logo, $Z_i \in C_{i+1} \subset M$.

3) Por (1) e (2), então, é fácil ver que para cada fórmula atômica $X \in S$, exatamente uma fórmula assignada $a_i(X)$ pertence a M, e que $a_i(F(X_1,X_2)) \in M$ see alguma consequência C de $\pi_1/\pi_2(a_i(F(X_1,X_2)))$ pertence a M.

Pelo teorema 1.5, existe, então, uma valorização proposicional v que satisfaz todos os elementos de B. ∎

COMENTÁRIO 2.3. A demonstração do teorema anterior, no fundo, é uma aplicação do lema de Tukey (caso enumerável), e imita a construção clássica para o teorema de Lindenbaun.

Do teorema 2.7, podem-se obter diretamente os teoremas de completude e compacidade demonstrados na Seção II.2.

2.8. COROLÁRIO. (Completude). *Seja* $B \subset S$; *então,* B *é satisfatível see existe um tableau aberto para algum conjunto* $a_D(B)$.

DEMONSTRAÇÃO. Se B é satisfatível, existe, claramente, um tableau aberto para $a_D(B)$ (ver Corolário 1.6, Seção I.1).

Por outro lado, se existe tableau aberto para $a_D(B)$, seja $\Gamma(B)$ o conjunto dos vértices do tableau para $a_D(B)$, e seja Γ o conjunto formado por todos os conjuntos $\Gamma(B)$.

Pelo lema 2.2, Γ tem caráter finito, e pela definição de tableau proposicional, valem as cláusulas P_o e P_1 da definição de PCAP; portanto, B é satisfatível. ∎

2.9. COROLÁRIO. (Compacidade). *Seja* $B \subset S$; *então* B *é satisfatível see todo subconjunto finito* B_o *de* B *é satisfatível.*

DEMONSTRAÇÃO. Suponhamos que todo $B_o \subset B$, B_o finito, é satisfatível; pelo corolário anterior, para cada B_o existe um tableau aberto para $a_D(B_o)$; utilizando diretamente o lema 2.2, e o corolário anterior, B é satisfatível. ∎

A recíproca é óbvia.

II.4 - UM ALGORÍTMO PARA AXIOMATIZAÇÃO DE LÓGICAS PROPOSICIONAIS.

Embora, em [25] os tableaux sejam considerados algorítmos para axiomatização de lógicas finitas, e essa denominação seja razoável devido ao teorema de completude para os tableaux, essa noção de axiomatização difere da noção "hilbertiana" usual no aspecto de que os tableaux são *analíticos* (isto é, dada uma fórmula, o tableau a decompõe para decidir se a fórmula é ou não teorema), enquanto que a noção hibertiana de axiomatização, com axiomas (ou postulados) explícitos e regras de derivação, é *sintética* (isto é, gera as fórmulas que são teoremas a partir dos postulados e através das regras).

Nesta seção, apresentamos um conjunto de procedimentos, estreitamente ligado ao método dos tableaux proposicionais, que funciona como um algoritmo para a axiomatização de qualquer lógica proposicional n-valente. Este método é uma adaptação da noção de *produção* introduzida por Post (ver [19]), e o denominamos *tableau sintético*, por suas semelhanças com as noções hibertianas (e "gentzenianas", também).

Os procedimentos ligados ao tableau sintético, que apresentamos a seguir, podem ser interpretados como instruções a uma máquina, que, partindo de um suprimento inicial de objetos, vai juntando a estes certos objetos produzidos a partir dos anteriores; o método consiste basicamente no seguinte:

1) Temos um suprimento inicial de objetos;
2) Temos instruções que possibilitam produzir novos objetos, juntando-os ao suprimento inicial;
3) Dispomos de instruções que permitem suprimir certos objetos do suprimento disponível;
4) As instruções são executadas em seqüência;
5) Cada instrução só é executada depois de esgotadas todas as possibilidades de execução de instrução anterior.

Utilizamos as seguintes notações para o tableau sintético:

1) $X_o, X_1, \ldots, X_n, \ldots$, são as variáveis proposicionais (i.e., fórmulas atômicas) em alguma enumeração fixada;
2) $Y, Y_1, Y_2, \ldots, Y_n, \ldots$, denotam fórmulas quaisquer (inclusive atômicas);
3) PY é interpretado como "Y é uma proposição";

4) $A_i Y$, $0 \leq i \leq n-1$, é interpretado como o "valor i é permitido para a fórmula Y"

5) $B_i Y$, $0 \leq i \leq n-1$, é interpretado como o "valor i é proibido para a fórmula Y";

6) $\longmapsto$ é interpretado como "anexe ao suprimento disponível os objetos à direita do símbolo, se ocorrem neste suprimento os objetos que estão à esquerda";

7) $\vdash\!\!\!-\!\!<$ é interpretado como "apague do suprimento disponível os objetos à direita, se ocorrem os que estão à esquerda";

8) TY é interpretado como "Y é teorema";

9) Q e C_i são símbolos auxiliares, cuja finalidade é evitar que o procedimento fique ocupado indefinidamente.

Seja L uma lógica proposicional n-valente, $F_1,\ldots,F_r$ os conectivos binários de L (os conectivos unários podem ser interpretados como casos particulares dos conectivos binários), f_1, $,\ldots,f_r$ as interpretações destes conectivos, $N = \{0,1,\ldots,d,\ldots,n-1\}$, o conjunto dos valores-verdade de L, e $D = \{0,1,\ldots,d-1\}$ o conjunto dos valores-verdade distinguidos; o *tableau sintético para* L, TS(L), é o seguinte conjunto de procedimentos:

$P_o)$ PX_o

$\downarrow$

$P_1)$ $PX_j \longmapsto A_i X_j \; , \; 0 \leq i \leq n-1$

$\downarrow$

$P_2)$ $PY_1, PY_2 \longmapsto QF_i(Y_1,Y_2) \; , \; 1 \leq i < r$

$\downarrow$

$P_3)$ $QY \longmapsto PY$

$\downarrow$

$P_4)$ $PY \longmapsto B_i Y \; , \; 0 \leq i \leq n-1$

$\downarrow$

$P_5)$ $\{A_{i_k} Y_k\} \longmapsto C_i F(Y_1,Y_2)$ para cada $\{a_{i_k} Y_k\}$ que é consequência de $\pi_1/\pi_2(a_i F(Y_1,Y_2))$

$\downarrow$

$P_6)$ $C_i Y \longmapsto A_i Y$

$\downarrow$

$P_7)$ $A_i Y \vdash\!\!-\!\!< B_i Y$

$\downarrow$

$P_8)$ $B_d Y, B_{d+1} Y, \ldots, B_{n-1} Y \longmapsto TY$

$\downarrow$

$P_9)$ $PX_j \longmapsto PX_{j+1}$

As setas indicam a ordem dos procedimentos; a descrição de cada passo é a seguinte:

$P_o)$ Inicie com a fórmula atômica X_o e a declare uma proposição; o conjunto unitário PX_o é o suprimento inicial;

$P_1)$ Se X_j é uma proposição, anexe ao suprimento de objetos $A_o X_j, A_1 X_j, \ldots, A_{n-1} X_j$ (significando que os valores $0, 1, 2, \ldots, n-1$ são todos permitidos às fórmulas atômicas);

$P_2)$ Se F_i é um conectivo e Y_1, Y_2 são proposições, produza $F_i(Y_1,Y_2)$ mas não a declare uma proposição; ao invés disso, rotule-a com o prefixo Q (a finalidade desse rótulo como mencionado, é evitar que o procedimento fique indefinidamente ocupado);

P_3) Após o término de P_2, declare como proposições as fórmulas rotuladas com Q;

P_4) Para todas as proposições obtidas, anexe ao suprimento os objetos B_oY, $B_1Y, \ldots, B_{n-1}Y$, significando que, a priori, todos os valores-verdade são proibidos para todas as proposições;

P_5) Se os valores i_k são permitidos para Y_k e $\{a_{i_k} Y_k\}$ é uma consequência $\pi_1/\pi_2(a_i(F(Y_1,Y_2))$ acrescente ao suprimento os objetos $C_iF(Y_1,Y_2)$ para cada conectivo F.

P_6) Após o término de P_5, anexe A_iY ao suprimento, para cada C_iY;

P_7) Se A_iY pertence ao suprimento, apague B_iY do suprimento (isto é, corrigimos nossa hipótese de que o valor i é proibido para Y);

P_8) Se, após P_7, restaram os objetos B_iY para $i \in N-D$ (isto é, todos os valores não-distinguidos são realmente proibidos para Y e Y é uma proposição), então Y é teorema;

P_9) Anexe ao suprimento o objeto PX_{j+1}, se PX_j pertence ao suprimento, e volte a P_1.

A relação entre os tableux proposicionais e os tableaux sintéticos é dada pelo seguinte teorema:

2.10. TEOREMA. *Seja* $L = \langle S, A \rangle$ *uma lógica proposicional* n-*valente e* $Y \in S$; *então existe um tableau proposicional aberto para* $a_i(Y)$ *see* $A_i Y$ *é produzido em* $TS(L)$.

DEMONSTRAÇÃO. Suponhamos que exista um tableau proposicional aberto para $a_i(Y)$; se Y é atômica, $A_i Y$ é produzido em $TS(L)$ por P_1. Se Y não é atômica, utilizando indução no nível de construção das fórmulas, suponhamos que Y seja da forma $F(Y_1, Y_2)$: por definição de tableau, desde que existe um tableau aberto para $a_i(F(Y_1, Y_2))$, existe um tableau aberto para algum conjunto $C = \{a_{i_k}(Y_k)\}$, onde C é consequência de $\pi_1/\pi_2(a_i(F(Y_1, Y_2))$. Por hipótese de indução, temos que $\{A_{i_k} Y_k\}$ é produzido, e por P_5 e P_6, $A_i F(Y_1, Y_2)$ é produzido.

Por outro lado, se $A_i Y$ é produzido em $TS(L)$, e se Y é atômica, existe um tableau aberto para $a_i(Y)$, que é exatamente o tableau formado pelo único vértice $a_i(Y)$ (lembramos que, para fórmulas atômicas, tableaux compostos por um único vértice são abertos, por definição).

Se Y não é atômica, suponhamos que Y seja da forma $F(Y_1, Y_2)$; evidentemente, $A_i F(Y_1, Y_2)$ só pode ter sido produzido por P_5 e P_6 a partir de objetos $\{A_{i_k} Y_k\}$ onde $\{a_{i_k}(Y_k)\}$ é uma consequência de $\pi_1/\pi_2(Y)$.

Utilizando novamente indução no nível de construção, existem, então, por hipótese de indução, tableaux proposicionais abertos para $a_{i_k}(Y_k)$, e pela definição de tableau, existe um tableau aberto para $a_i(F(Y_1,Y_2))$. ■

Do teorema 2.10 segue-se a completude de TS(L):

2.11. COROLÁRIO. (Completude para TS(L). *Se* $L = \langle S, A \rangle$ *é uma lógica proposicional* n-*valente e* $Y \in S$, *então* TY *é produzido em* TS(L) *see* $\vdash_{\overline{L}} Y$.

DEMONSTRAÇÃO. TY é produzido em TS(L) see $A_i Y$ não é produzido, para todo $i \in ND$ (por P_8), see todo tableau proposicional para $a_i(Y)$, $i \in ND$, é fechado (pelos teoremas 2.10 e corolário 1.6), see $\vdash_{\overline{L}} Y$ (pelo teorema de completude para tableaux proposicionais). ■

COMENTÁRIO 2.4. Em termos intuitivos, obter a produção de TY em TS(L) equivale ao fato de que é possível percorrer todos os tableaux proposicionais abertos para $a_i(Y)$, $i \in D$, de baixo para cima, mas não é possível percorrer nenhum tableau para $a_j(Y)$ onde $j \in ND$.

EXEMPLOS. Vamos considerar dois exemplos de aplicação de TS(L):

a) Para o cálculo proposicional clássico C, e

b) Para o cálculo proposicional trivalente P^1 descrito em [22].

a) Para o cálculo clássico, os tableaux para a implicação $\longmapsto$ e para a negação $\sim$ (onde $N = \{0,1\}$ e $D = \{0\}$) são:

1) $\dfrac{a_o(X \to Y)}{a_1 X \vee a_o Y}$

2) $\dfrac{a_1(X \to Y)}{a_o X \wedge a_1 Y}$

3) $\dfrac{a_o(\sim X)}{a_1 X}$

4) $\dfrac{a_1(\sim X)}{a_o X}$

<u>TS(C)</u>

P_o) PX_o

P_1) $PX_j \longmapsto A_o X_j,\ A_1 X_j$

P_2) $PY_1, PY_2 \longmapsto Q(Y_1 \to Y_2)$

$PY \longmapsto Q(\sim Y)$

P_3) $QY \longmapsto PY$

$P_4)$ $PY \longmapsto B_oY,\ B_1Y$

$P_5)$ $A_1X \longmapsto C_o(X \to Y)$

$A_oY \longmapsto C_o(X \to Y)$

$A_oX, A_1Y \longmapsto C_1(X \to Y)$

$A_1X \longmapsto C_o(\sim X)$

$A_oX \longmapsto C_1(\sim X)$

$P_6)$ $C_iY \longmapsto A_iY$

$P_7)$ $A_iY \longmapsto B_iY$

$P_8)$ $B_1Y \longmapsto TY$

$P_9)$ $PX_j \longmapsto PX_{j+1}$

Se chamamos *ciclo* a uma aplicação completa de procedimento, desde P_1 até P_9, e D_i ao conjunto de objeto disponíveis após aplicação do passo P_i, do primeiro ciclo resultam os seguintes objetos:

$D_0 = \{PX_0\}$

$D_1 = D_0 \cup \{A_0X_0, A_1X_0\}$

$D_2 = D_1 \cup \{Q(X_0 \to X_0) \ , \ Q(\sim X_0)\}$

$D_3 = D_2 \cup \{P(X_0 \to X_0), \ \ P(\sim X_0)\}$

$D_4 = D_3 \cup \{B_0(X_0 \to X_0), \ B_1(X_0 \to X_0) \ ,$

$\quad B_0(\sim X_0), \ B_1(X_0), \ B_0(X_0), \ B_1(X_0)\}$

$D_5 = D_4 \cup \{C_0(X_0 \to X_0), \ C_1(\sim X_0)\}$

$D_6 = D_5 \cup \{A_0(X_0 \to X_0), \ A_1(\sim X_0)$

$D_7 = D_6 - \{B_0(X_0 \to X_0), \ B_1(\sim X_0), \ B_0(X_0), \ B_1(X_0)\}$

$D_8 = D_7 \cup \{T(X_0 \to X_0)\}$

$D_9 = D_0 \cup \{PX_1\}$

Do primeiro ciclo, então, resultou o único teorema $X_0 \to X_0$; do segundo ciclo, resultarão os teoremas $X_1 \to X_1$, $\sim X_0 \to \sim X_0$, e $X_0 \to (X_0 \to X_0)$.

b) Para o cálculo P^1 (cf. [22]), utilizando os tableaux descritos na Seção I.2, temos (utilizamos o símbolo / para evitar repetições de linhas).

$\underline{TS(P^1)}$

$P_o)$ PX_o

$P_1)$ $PX_j \longmapsto A_oX_j,\ A_1X_j,\ A_2X_j$

$P_2)$ $PY_1,\ PY_2 \longmapsto Q(Y_1 \supset Y_2)$

$PY \longmapsto Q(\neg Y)$

$P_3)$ $QY \longmapsto PY$

$P_4)$ $PY \longmapsto B_oY,\ B_1Y,\ B_2Y$

$P_5)$ $A_1X/A_2X \longmapsto C_o(\neg X)$

$A_oX \longmapsto C_2(\neg X)$

$A_2X/A_oY/A_1Y \longmapsto C_o(X \rightarrow Y)$

$\{A_oX,A_2Y\}/\{A_1X,A_2Y\} \longmapsto C_2(X \rightarrow Y)$

$A_oX/A_1X/A_oY/A_1Y \longmapsto C_o(X \vee Y)$

$\{A_2X,A_2Y\} \longmapsto C_2(X \vee Y)$

$\{A_oX,A_oY\}/\{A_oX,A_1Y\}/\{A_1X,A_oY\}/\{A_1X,A_1Y\} \longmapsto C_o(X\ \&\ Y)$

$A_2X/A_2Y \longmapsto C_2(X\ \&\ Y)$

$P_6)$ $C_iY \longmapsto A_iY$

$P_7)$ $A_iY \longmapsto B_iY$

$P_8)$ $B_2Y \longmapsto TY$

$P_9)$ $PX_j \longmapsto PX_{j+1}$

No Apêndice II, mostramos um programa PASCAL que, seguindo as instruções de $TS(P^1)$, demonstra automaticamente todos os teoremas de P^1, desde que se disponha de tempo suficiente.

CAPÍTULO III

CARACTERIZAÇÃO DE LÓGICAS PROPOSICIONAIS POR CONFIGURAÇÕES DE CONECTIVOS

III.1 - CONFIGURAÇÕES DE CONECTIVOS

Este capítulo constitui uma espécie de digressão à nossa linha do trabalho, cuja sequência natural seria apresentar as lógicas n-valentes de primeira ordem do ponto de vista dos tableaux; preferimos apresentá-lo aqui, contudo, por sua aplicação imediata aos cálculos proposicionais, embora os resultados obtidos não sejam necessários aos capítulos posteriores.

Uma das características mais interessantes, a ao mesmo tempo mais embaraçosa, dos cálculos n-valentes (proposicionais) é a enorme quantidade de conectivos disponíveis; permanece em aberto, por exemplo, para $n \geq 4$, a classificação dos conectivos em relação aos quais as lógicas proposicionais são funcionalmente completas (cf.[12],[18],[21]) ou a classificação dos conectivos "vazios",i.é, dos conectivos tais que não existe nenhum teorema onde somente esse conectivo ocorra (ver [10]). No cálculo proposicional clássico, por exemplo, os conectivos $\wedge$ e $\vee$ são vazios, mas a implicação não é vazia ($(X \supset X)$ é teorema).

Uma questão que se coloca, também relacionada com os conectivos, é a seguinte: para uma mesma linguagem proposicional, variando-se as interpretações dadas aos conectivos, como se relacionam as fórmulas que são teoremas nas lógicas obtidas?

Em termos formais, a questão pode ser formulada da seguinte maneira: Seja S uma linguagem proposicional fixada, e sejam $F_1, F_2, \ldots, F_s$ os conectivos de S. Sejam $A = \langle N, \{f_i : 1 \leq i \leq s\}\rangle$ e $A' = \langle N', \{f_i' : 1 \leq i \leq s\}\rangle$ álgebras abstratas de mesmo tipo de similaridade de S tais que $\mathcal{A} = \langle A, D\rangle$ e $\mathcal{A}' = \langle A', D'\rangle$ são pré-modelos para S (ver Seção I.1) onde N e N' podem ser distintos, e D e D' são subconjuntos quaisquer de N e N' respectivamente. Se considerarmos as lógicas $L = \langle S, \mathcal{A}\rangle$ e $L' = \langle S, \mathcal{A}'\rangle$, qual é a relação entre os teoremas de L e de L'?

Neste capítulo veremos que esta questão pode ser respondida com base em certas relações de equivalência entre os valores-verdade e entre as interpretações dos conectivos. Para facilitar as discussões, chamaremos de *conectivos* às interpretações $\{f_i : 1 \leq i \leq s\}$ dos conectivos propriamente ditos $\{F_i : 1 \leq i \leq s\}$.

Nesta parte do trabalho vamos, então, estudar certas classes de conectivos que chamaremos de *configurações de conectivos lógicos*; dadas as funções $f : N \times N \to N$ e $g : N \to N$, que representam conectivos em L, onde $N = D \cup ND$ é o conjunto de valores-verdade, D são os valores distinguidos e ND são os não distinguidos, seja $P_k = \{C_0, C_2, \ldots, C_{k-1}\}$ uma partição de N tal que para cada i, $0 \leq i < k$, $C_i \neq \phi$ e $C_i \subset D$ ou $C_i \subset ND$; se $C_i \subset D$, dizemos que C_i é uma *classe distinguida*, e se $C_i \subset ND$ dizemos que C_i é uma *classe não distinguida*.

Dizemos que P_k é uma *partição proposicional* para L se, além destas condições, as classes são *fechadas pelos conectivos*, isto é, para todo $f, g : f(C_i, C_j) \subset C_r$, $g(C_i) \subset C_s$ para todo $i, j < k$ (onde $r, s < k$).

Seja $L = \langle S, \langle A, D \rangle \rangle$ uma lógica proposicional que admite uma partição proposicional P_k, onde

$$A = \langle N, \{f_i : 1 \leq i \leq s,\ g_j : i \leq j \leq s'\} \rangle$$

são tais que f_i, $1 \leq i \leq s$ são conectivos binários e g_j, $1 \leq j \leq s'$, são conectivos unários.

É fácil verificar que a relação de equivalência definida pela partição proposicional P_k sobre o conjunto N é uma *congruência* com respeito a todas as operações em A (ver [14]); de fato, se $x, y \in N$ e definimos $x \sim_{P_k} y$ see $x, y \in C_i$, temos que

1) $x_1 \sim_{P_k} x_2$, $y_1 \sim_{P_k} y_2$ implica $f_i(x_1, y_1) \sim_{P_k} f_i(x_2, y_2)$

e

2) $x_1 \sim_{P_k} x_2$ implica $g_j(x_1) \sim_{P_k} g_j(x_2)$,

pela definição de partição proposicional.

Seja $N/\sim_{P_k} = \{[x] : x \in N\} = P_k$ o conjunto das classes de equivalência $[x]$ pela relação $\sim_{P_k}$.

Definimos a *álgebra quociente* $A/\sim_{P_k}$ como a álgebra abstrata:

$$A/\sim_{P_k} = \langle \langle N/\sim_{P_k}, \phi_{f_i}, \Gamma_{g_j} : 1 \leq i \leq s,\ 1 \leq j \leq s' \} \rangle, D_k \rangle$$

onde:

1) D_k é o conjunto das classes distinguidas de $N/\sim_{P_k}$

e

2) para cada f_i, g_j em A,

$$\phi_{f_i}([x], [y]) = [f_i(x,y)]$$

$$\Gamma_{g_j}([x]) = [g_j(x)]$$

É claro que a ágebra $A/\sim_{P_k}$ é de mesma similaridade que a álgebra A.

Seja S uma linguagem proposicional fixada e Λ uma classe de pré-modelos $A_i = \langle A_i, D_i \rangle$ para S, onde A_i são álgebras de mesma similaridade; seja também λ a classe de lógicas proposicionais $L_i = \langle S, A_i \rangle$ correspondente a Λ.

Se $L = \langle S, A \rangle$ e $L' = \langle S, A' \rangle$ são lógicas em λ, respectivamente n-valente e m-valente, e N e M são os respectivos conjuntos de valores-verdade, dizemos que as partições proposicionais

$$P_k = \{C_0, \ldots, C_{k-1}\} \text{ de } N$$

e

$$P'_{k'} = \{C'_0, \ldots, C'_{k'-1}\} \text{ de } M \quad \text{(a menos de permutação dos índices)}$$

são *partições similares* se as seguintes condições são satisfeitas:

P-1) $k = k'$ e

P-2) Para todo $0 \leq i \leq k.1$, C_i e C'_i são ambas classes distinguidas ou classes não distinguidas.

P-3) Se f,g são operações em A, existem $f'g'$ em A' tais que $f(C_i, C_j) \subset C_r$ see $f'(C'_i, C'_j) \subset C'_r$, e $g(C_i) \subset C_s$ see $g'(C'_i) \subset C'_s$.

Temos, então, o seguinte:

3.1. TEOREMA. *Seja* λ_o *a subclasse das álgebras em* λ *que admitem partições similares; se* A, A' *estão em* λ_o *com partições* P_k *e* P_k', *então* $A/\sim_{P_k}$ *e* $A'/\sim_{P_k'}$ *são isomorfas.*

DEMONSTRAÇÃO. Seja $\rho : P_k \longrightarrow P_k'$ uma bijeção entre P_k e P_k' tal que $\rho(C_i) = C_{\rho(i)}$ é uma classe distinguida em P_K' see C_i é uma classe distinguida em P_k.

Sejam $x,y \in N$, e f,g operações em A tais que $[x] = C_i$, $[y] = C_j$, $[f(x,y)] = C_r$, $[g(x)] = C_s$; sejam f',g' operações correspondentes em A'.

Vamos mostrar que ρ é um isomorfismo entre $A/\sim_{P_k}$ e $A'/\sim_{P_k'}$; temos, por hipótese, que

(1) $\rho([f(x,y)]) = \rho(\phi_f([x],[y]))$, ou ainda

(2) $\rho(C_r) = C_{\rho(r)} = \rho(\phi_f(C_i,C_j))$

Pela definição de partições similares,

(3) $\phi_f(C_i,C_j) = C_r$ see $\phi_{f'}(C_{\rho(i)}, C_{\rho(j)}) = C_{\rho(r)}$ e desde que, por hipótese, $\phi_f(C_i,C_j) = C_r$, temos $\rho(\phi_f(C_i,C_j)) = \phi_{f'}(\rho(C_i), \rho(C_j))$, de (2) e (3) e ρ é um homormorfismo. (a prova para funções unárias é análoga). Como ρ é bijetora, ρ é um isomorfismo. ∎

Seja π' a *função projeção* de A' em $A'/\sim_{P_k}$ tal que $\pi'(x) = [x]$, e A uma álgebra em λ_o fixada; é claro que, para toda álgebra A' em λ_o, existe um homomorfismo de A' em $A/\sim_{P_k} = A_k$ (este homomorfismo é π, se $A' = A$, ou $\pi = \rho \cdot \pi'$ em caso contrário, como ilustra o diagrama.

$$\begin{tikzcd}
A' \arrow[d, "\pi'"'] \arrow[dr] & \\
A'/\sim_{P_k} \arrow[r, leftrightarrow, "\rho"] & A_k
\end{tikzcd}$$

Se Φ_f, Γ_g são operações em A_k, dizemos que Φ_f, Γ_g são *configurações de conectivos*, e que Φ_f, Γ_g *representam* os conectivos f', g' de uma álgebra qualquer A' de λ_o.

Dizemos que o par $L_k = \langle S, A_k \rangle$ é a *lógica representante* das lógicas $L \in \lambda_o$.

III.2 - CARACTERIZAÇÃO DAS LÓGICAS PROPOSICIONAIS n-VALENTES

A motivação para as definições precedentes é a seguinte: se tomamos certas classes de lógicas proposicionais e partimos os valores-verdade dividindo os valores distinguidos em algumas classes e os valores não distinguidos em outras classes (mas sem misturar valores), poderemos definir "matrizes lógicas" cujas "entradas" serão as classes de equivalência.

Essas "matrizes", como uma nova interpretação de S, formarão um "cálculo proposicional" que permite comparar, de certo modo, o poder de dedução das lógicas dessa classe.

Um exemplo desse procedimento é o seguinte:

Sejam L_1 o cálculo proposicional clássico, com o conjunto $N_1 = \{0,1\}$ de valores-verdade, $D_1 = \{0\}$, e L_n, $n \geq 2$ uma sequência de cálculos proposicionais n-valentes com $D = \{0,1,\ldots,d\}$, $d < n-1$. Seja f_1 a matriz para a implicação clássica:

f_1	0	1
0	0	1
1	0	0

f_2	0	1	2
0	0	1	2
1	1	0	2
2	1	1	0

$N = \{0,1,2\}$

$D = \{0,1\}$

e sejam f_n funções tais que:

$f_n(D \times D) \subset D$, $f_n(D \times ND) \subset ND$, $f_n(ND \times D) \subset D$, $f_n(ND \times ND) \subset D$.
(ver exemplo acima).

Se definimos as partições $C_0 = D$, $C_1 = ND$ para cada cálculo, teremos duas classes: T_0 e T_1 e podemos definir a função projeção π tal que $\pi(x) = T_0$ se $x \in D$, e $\pi(x) = T_1$, se $x \in ND$.

Neste caso, se definimos ϕ como a matriz

$$\begin{array}{c|cc} & T_o & T_1 \\ \hline T_o & T_o & T_1 \\ T_1 & T_o & T_o \end{array} ,$$

teremos $\phi(\pi(x), \pi(y)) = \pi(f_n(x,y))$, para todo n. É óbvio que neste caso todos os teoremas clássicos que envolvam somente o conectivo de implicação continuarão a valer em todas as lógicas L_n; isto está diretamente ligado ao fato de que a configuração de conectivos ϕ é exatamente a implicação clássica. Os próximos resultados formalizam este ponto de vista, e o resultado principal deste capítulo mostra que a recíproca também é verdadeira, isto é, em nenhum desses cálculos haverá novos teoremas.

Para facilitar as demonstrações, omitiremos o caso dos conectivos unários nos teoremas seguintes.

3.2. TEOREMA (Caracterização de lógicas proposicionais n-valentes). *Sejam* $L = \langle S,A \rangle$ *uma lógica tal que* $L \in \lambda_o$ *e* $F(X,Y)$ *uma fórmula de* L; *então* $\vdash_L F(X,Y)$ *see* $\vdash_{L_k} F(X,Y)$.

DEMONSTRAÇÃO. i) Suponhamos, em primeiro lugar, que X e Y são fórmulas atômicas, e seja v uma valorização proposicional para L. Se v é tal que $v(F(X,Y)) \in D$, então, $\pi(v(F(X,Y))=T_i$, $T_i \in D_k$, see $\pi(f(v(X), v(Y))) = T_i$ see $\phi_f(\pi(v(X)), \pi(v(Y))) = T_i$ pela definição de função projeção.

Considerando-se a função $\pi \circ v$, é fácil ver que $\pi \circ v$ é uma valorização proposicional para L_k, e temos que $v(F(X,Y)) \in D$ see $\pi \circ v(F(X,Y) \in D_k$.

Desde que L e L_k são completas (pois L_k é uma lógica proposicional que satisfaz claramente todas as condições das lógicas proposicionais consideradas nos capítulos I e II) resulta que $\vdash_L F(X,Y)$ see $\vdash_{L_k} F(X,Y)$.

ii) Se e não são atômicas, basta raciocinar por indução no nível das fórmulas. ∎

O teorema precedente caracteriza conectivos da seguinte forma: definimos a *valência* de um conectivo F em uma lógica n-valente L como o *menor* k tal que existe uma configuração ϕ_f em L_k que representa F; desse modo um cálculo é n-*valente estrito* quando tem pelo menos um conectivo com valência n.

3.3. TEOREMA. *Um cálculo proposicional é* n-*valente estrito see a única partição proposicional para* N *é a partição trivial* $P_n=\{\{0\},\{1\},\ldots,\{n-1\}\}$.

DEMONSTRAÇÃO. Se L é um cálculo n-valente estrito, então existe pelo menos um conectivo F em L tal que F tem valência n, isto é, o menor k tal que existe configuração ϕ_f em L_k é n; pela definição de L_n, então, a única partição dos valores de N em n classes é a partição trivial.

Por outro lado, se a única partição proposicional para N com respeito a L é a partição trivial, existe algum conectivo F em L tal que, se P_k é uma partição com alguma classe C_i contendo dois elementos pelo menos, $f|C_i \times C_j$ não está contido nos elementos da partição P_k, para algum $j < k$, se $k < n$.

Portanto, n é o menor valor k tal que existe ϕ_f em L_k e, então, por definição de valência, F tem valência n, e L é n-valente estrito. ∎

Por exemplo, o cálculo 3-valente descrito em [6] é 3-valente estrito, pois os conectivos são descritos pelas tábuas:

$\underline{C}$	0	1	2
0	0	1	2
1	0	0	1
2	0	0	0

	$\underline{N}$
0	2
1	1
2	0

$N = \{0,1,2\}$

$D = \{0,1\}$

e as únicas partições possíveis de N tais que $C_i \subset D$ ou $C_i \subset ND$ são $P_2 = \{\{0,1\},\{2\}\}$ e $P_3 = \{\{0\}, \{1\},\{2\}\}$.

No primeiro caso, se $C_o = \{0,1\}$ e $C_1 = \{2\}$, embora $\underline{C}(C_o, C_o) \subset C_o$, $\underline{C}(C_1, C_o) \subset C_o$, $\underline{C}(C_1, C_1) \subset C_o$, como $\underline{C}(C_o, C_1)$ não está contido nem em C_o nem em C_1, P_2 não é uma partição proposicional para N nesse cálculo, e portanto P_3 é a única partição proposicional possível.

Segue-se imediatamente do teorema 3.3 e da definição de valência o seguinte corolário:

3. . COROLÁRIO. *Se* L *é um cálculo* n-*valente tal que todos os seus conectivos têm valência* 2, *então todo teorema de* L *é um teorema do cálculo proposicional clássico, e reciprocamente*

DEMONSTRAÇÃO. $\vdash_L F$ see $\vdash_{L_2} F$, pelo teorema 3.3, e L_2 é o cálculo proposicional clássico. ∎

Como ilustração do método, temos, por exemplo, que as duas interpretações de um conectivo F em uma lógica 4-valente, descri - tas pelas matrizes:

f_1	0	1	2	3
0	1	0	2	3
1	0	1	2	3
2	1	0	1	3
3	0	1	3	3

$D = \{0,1\}$

f_2	0	1	2	3
0	0	0	2	3
1	0	0	2	3
2	0	0	0	3
3	0	0	3	3

$D = \{0,1\}$

ϕ	T_o	T_1	T_2
T_o	T_o	T_1	T_2
T_1	T_o	T_o	T_2
T_2	T_o	T_2	T_2

são equivalente, pois para f_1 e f_2 podemos tomar $P_3 = \{C_o, C_1, C_2\}$ onde $C_o = \{0,1\}$, $C_1 = \{2\}$ e $C_2 = \{3\}$ e ambas são representadas pela configuração ϕ.

Se considerarmos interpretações g_1 e g_2 de conectivos unários descritos abaixo, essas interpretações também serão equivalentes, pois são descritas pela configuração Γ.

g_1	
0	2
1	2
2	0
3	1

g_2	
0	2
1	2
2	1
3	0

Γ	
T_o	T_1
T_1	T_o
T_2	T_o

CAPÍTULO IV

LÓGICAS POLIVALENTES DE PRIMEIRA ORDEM

IV.1 - DEFINIÇÕES PRELIMINARES

Seja S uma linguagem para uma lógica proposicional L n-valente com d valores distinguidos.

Por uma *linguagem de primeira ordem* S^1 para uma lógica n-valente de primeira ordem L^1 (associada à linguagem proposicional S) entendemos uma estrutura construida com os seguintes símbolos:

ALFABETO DE S^1

a) os símbolos de S que não sejam variáveis proposicionais;

b) Um conjunto finito com pelo menos dois elementos $QT = \{Q_1, Q_2, \ldots, Q_q\}$; os membros de QT chamam-se *quantificadores;*

c) Um conjunto VI enumerável e não vazio de símbolos denominados *variáveis individuais;* $VI = \{v_1, \ldots, v_n, \ldots\}$;

d) Um conjunto CI enumerável e não vazio de símbolos, denominados *constantes individuais* ou *parâmetros;* $CI = \{c_1, \ldots, c_n, \ldots\}$, tal que $CI \cap VI = \phi$;

e) Para cada $m \in \mathbb{N} - \{0\}$, um conjunto enumerável PR_m de símbolos chamados *predicados m-ários.*

Usaremos como variáveis metalinguísticas: x,y,z (com ou sem índices) para denotar variáveis; a,b,c (com ou sem índices), para denotar constantes; P,R,S (com ou sem índices), para denotar símbolos de predicados; e Q, para denotar quantificadores. As variáveis não-linguísticas usadas na parte proposicional continuarão a ser utilizadas.

Uma *linguagem de primeira ordem* S^1 *construida com o alfabeto de* S^1 é um conjunto construido da seguinte maneira:

1) Uma *fórmula atômica em* S^1 é uma n+1-upla $Pr_1r_2 \ldots r_n$, onde $P \in PR_n$ e $r_i \in VI \cup CI$;

2) Uma *fórmula* é um elemento de S^1 definido da seguinte maneira:

F-1) $S_o^1 = \{X : X$ é fórmula atômica$\}$;

F-2) $S_{n+1}^1 = S_n^1 \cup \{F(X_1, \ldots, X_p) : X_i \in S_n^1$ e F são conectivos proposicionais p-ários$\} \cup \{(Q_j x)A : A \in S_n^1$, $x \in VI$ e $Q_j \in QT\}$

F-3) $S^1 = \bigcup_{n \in \omega} S_n^1$.

OBSERVAÇÃO: De modo rigoroso, deveríamos escrever $FX_1 \ldots X_p$ em lugar de $F(X_1, \ldots, X_p)$, desde que formalmente nossa linguagem não inclue símbolos auxiliares, além do que a primeira forma é mais eficiente do ponto de vista sintático para evidenciar a unicidade da escrita de uma fórmula. No entanto, para facilidade da leitura, vamos utilizar símbolos auxiliares como vírgulas, parênteses,etc.

O símbolo de igualdade = é usado sempre em sentido metalinguístico.

Uma *fórmula pura* é uma fórmula que não contém constantes individuais.

Pelo *nível* de uma fórmula de primeira ordem entendemos a quantidade de conectivos e quantificadores que a fórmula contém; definindo indutivamente:

n_o) $n(X) = 0$, se X é fórmula atômica;

n_1) $n(F(X_1,\ldots,X_p)) = \sum\limits_{i=1}^{p} n(X) + 1$;

n_2) $n((Qx)A) = n(A) + 1$.

Notamos que $n(X) = r$ see $X \in S^1_r - S^1_{r-1}$.

Definimos a noção de *substituição por constantes* como se segue: Se A é uma fórmula, x uma variável, e a uma constante, então:

S_1) Se A é atômica, $[A]^x_a$ é o resultado da substituição de todo x que ocorre em A por a;

S_2) $[F(X_1,\ldots,X_m)]^x_a = F([X_1]^x_a,\ldots,[X_m]^x_a)$;

S_3) $[(Qx)A]^x_a = (Qx)A$;

S_4) $[(Qy)A]^x_a = (Qy)[A]^x_a$ se y é diferente de x.

Uma fórmula A é uma *fórmula fechada* ou uma *sentença* quando para toda variável x e toda constante a, $[A]^x_a = A$; a fórmula $[A]^x_a$ é chamada uma *instância* de A. Denotamos por $\overline{S}^1$ o conjunto das sentenças em S^1.

Definimos indutivamente o símbolo $[A]^{x_1,\dots,x_r}_{a_1,\dots,a_r}$ como $[A^{x_1,\dots,x_{r-1}}_{a_1,\dots,a_{r-1}}]^{x_r}_{a_r}$, onde $x_1 \neq x_2 \neq x_3 \neq \dots \neq x_r$.

Uma *subfórmula* de uma fórmula X é definida da seguinte maneira:

SF_1) Se $X = F(Y_1,\dots,Y_m)$, então $Y_1,\dots,Y_m$ são subfórmulas (imediatas) de X;

SF_2) Se $X = (Qx)A$, então, para toda constante a, $[A]^x_a$ é subfórmula (imediata) de X;

SF_3) Subfórmulas de subfórmulas de X são subfórmulas de X.

Por uma *árvore de subfórmulas* para uma fórmula definimos uma árvore AS tal que:

AS_1) Todos os vértices finais de AS são fórmulas atômicas:

AS_2) Se $F(X_1,\dots,X_m)$ é um vértice de AS, então $X_1,\dots,X_m$ são os únicos sucessores imediatos de $F(X_1,\dots,X_m)$;

AS_3) Se $(Qx)A$ é um vértice de AS, então se $a_1,\dots,a_r,\dots$ é uma enumeração das constantes individuais, todas as fórmulas $[A]^x_{a_1},\dots,[A]^x_{a_r}\dots$ são sucessores imediatos de $(Qx)A$.

COMENTÁRIO 4.1. As árvores de subfórmulas têm todos os ramos finitos, mas não são finitamente geradas (isto é, um vértice pode ter infinito sucessores).

Definimos uma *árvore de subformulas para a fórmula* X, AS(X), como a árvore de subfórmula onde X é o vértice de nível zero.

4.1. TEOREMA. *Se* AS(X) *é a árvore de subfórmulas para* X, *então:*

i) Y *é subfórmula de* X *see* Y *ocorre em algum vértice de* AS(X);

e

ii) *O nível de* X *é o comprimento do ramo de comprimento máximo.*

DEMONSTRAÇÃO. Por indução sobre o nível das fórmulas. ■

Por um *universo* entendemos um conjunto não vazio U; uma *fórmula com constantes em* U ou uma *U-fórmula* é uma fórmula construida como anteriormente, mas trocando-se CI pelo conjunto U. Formalmente,

U-1) Uma *U-fórmula atômica* é uma (m+1)-upla

$Pu_1u_2 \ldots u_m$, onde $P \in PR_m$ e $u_i \in U \cup VI$;

U-2) Uma *U-fórmula* é definida como anteriormente, a partir das U-fórmulas atômicas.

Definimos, de maneira análoga as noções de U-substituição , U-subfórmula, etc. Para cada universo U, definimos $S^1(U)$ como sendo o conjunto das U-fórmulas, e denotamos por X^U uma fórmula correspondente, em $S^1(U)$, à fórmula X em S^1. É claro, a partir destas definições, que as fórmulas puras são as mesmas em S^1 e $S^1(U)$.

Definimos a *parte proposicional da linguagem de primeira ordem* S^1, denotada por S', como o subconjunto de S^1 formado pelas fórmulas sem quantificadores . É fácil ver que S', assim como S, é uma álgebra livremente gerada, na classe das álgebras de mesma similaridade, a partir das fórmulas atômicas (S_o^1) pelos co-conectivos de S^1.

É evidente que podemos tratar S' como linguagem proposicional; usaremos esse ponto de vista para definir valorizações de primeira ordem (ver Seção 4.2).

Definimos uma *pré-substituição em primeira ordem* como uma função

$$s : VI \longrightarrow VI \cup CI \ .$$

Se X é uma fórmula atômica de S', X de forma $Pr_1r_2...r_n$, podemos estender s a uma função $s' : S_o^1 \longrightarrow S_o^1$, tal que $s'(Pr_1r_2...r_n) = P(s(r_1))s(r_2)...s(r_n))$, e como S_o^1 gera S', a função s' pode ser estendida a um único homomorfismo $Sb : S' \longrightarrow S'$, chamado *substituição em primeira ordem*, definido como:

$Sb(X) = s'(X)$, se X é atômica, e

$Sb(X) = F(Sb(X_1), Sb(X_2))$, se X é da forma $F(X_1,X_2)$.

A noção de substituição por constantes, definida anteriormente, pode ser vista como caso particular de substituição em primeira ordem.

IV.2 - SEMÂNTICA PARA LINGUAGENS n-VALENTES DE PRIMEIRA ORDEM

A noção de valorização proposicional foi introduzida na Seção I.1 como um homomorfismo especial, estabelecendo a conexão entre o valor de verdade de uma fórmula e o valor de verdade de suas subfórmulas. Para que uma conexão análoga se mantenha para fórmulas de primeira ordem, necessitamos estabelecer a relação entre o valor de verdade de uma fórmula quantificada e o valor de verdade de suas subfórmulas. É claro que essa dependência não poderá ser um homomorfismo usual, pois uma fórmula quantificada possui infinitas subfórmulas (ver Seção IV.1).

Se dividirmos os valores de verdade das subfórmulas em um certo número finito de classes, poderemos ter uma solução para o problema das fórmulas quantificadas se soubermos qual é a relação entre as classes e o valor da fórmula; definimos essa relação como sendo a *interpretação* do quantificador.

Introduzimos, então, a *função distribuição* de uma fórmula A, relativamente à variável x e à valorização v, como a função que distribue os valores de verdade das subfórmulas de uma fórmula quantificada em n classes, cada uma correspondendo a um dos valores-verdade, e que informa se cada classe é vazia ou não. A *função interpretação* para o quantificador, desse modo, é uma função cujo domínio é o conjunto das sequências de comprimento n formadas pelos símbolos "vazio" (**0**) e "não vazio" (**1**), denotada por V_2^n, e cujo contradomínio é o conjunto de valores de verdade, e as valorizações de primeira ordem são definidas, para fórmulas quantificadas, como composições algébricas de funções distribuição e funções interpretação.

Estas noções, introduzidas como generalização do que ocorre no caso clássico (ver discussão abaixo, a respeito dos quantificadores clássicos) podem ser vistas, de certa forma, como casos particulares da definição de quantificadores de Rescher - Mostowski (ver [15], pp. 197-206); contudo, as motivações para as definições, neste caso, são totalmente distintas.

Seja V_2^n o conjunto de todas as sequências de comprimento n, formadas pelos símbolos **0** e **1**, que serão interpretados como *vazio* e *não vazio* respectivamente, com pelo menos um símbolo **1**; sejam A uma fórmula e x variável livre de A, Q um quantificador; definimos uma *valorização de primeira ordem* para as fórmulas fechadas de $S^1(U)$, denotado por $S^1(\overline{U})$, como sendo uma função

$$v : S^1(\overline{U}) \longrightarrow N \ , \qquad \text{tal que:}$$

V_1) v é uma valorização proposicional para S'; e

V_2) $v(Qx)A = \sigma_q(D(A,x,v))$,

onde $\sigma_Q : V_2^n \longrightarrow n$ é a *função interpretação do quantificador* Q e D(A,x,v) é o valor da *função distribuição de* A *relativa a* x definida como $D_{x,v} : S^1(U) \longrightarrow V_2^n$ tal que

$$D_{x,v}(A) = D(A,x,v) = (d(0),d(1),\ldots,d(n-1)) \ , \text{ onde } \ d(i) = \mathbf{1}$$

se existe $k \in U$ tal que $v([A]_k^x) = i$, e $d(i) = \mathbf{0}$, em caso contrário.

Assumimos que todo quantificador Q de S^1 possue uma função interpretação σ_Q correspondente a Q, e denominamos o terno $L^1 = \langle S^1, A, \{\sigma_Q : Q$ é um quantificador em $S^1\}\rangle$ uma *lógica n-valente* *de primeira ordem associada à lógica proposicional* $L = \langle S, A \rangle$.

A título de ilustração, vamos considerar as funções interpretação $\sigma_\forall$ e $\sigma_\exists$ no caso do cálculo clássico; nesse caso, V_2^2 é o conjunto das duplas $\{(\mathbf{0},1), (\mathbf{1},0)$ e $(\mathbf{1},\mathbf{1})\}$ e as funções interpretação, respectivamente para os quantificadores $\forall$ e $\exists$, são:

0 1	$\sigma_\forall$
$(\mathbf{0},\mathbf{1})$	1
$(\mathbf{1},\mathbf{0})$	0
$(\mathbf{1},\mathbf{1})$	1

0 1	$\sigma_\exists$
$(\mathbf{0},\mathbf{1})$	1
$(\mathbf{1},\mathbf{0})$	0
$(\mathbf{1},\mathbf{1})$	0

(0 = verdadeiro)

(1 = falso)

Intuitivamente, os símbolos **0** e **1** devem ser interpretados como ausência e presença de valorizações (para as instâncias de uma fórmula) com valor 0 na primeira coluna e 1 na segunda. Assim, $\sigma_\forall$ significa que $(\forall x)A(x)$ tem valor 0 see todas as instanciações recebem valor 0 por todas as valorizações.

COMENTÁRIO 4.2. Desde que, para cada n, $|V_2^n| = 2^n-1$, a cardinalidade do conjunto de funções σ_Q é exatamente $n^{(2^n-1)}$; esta é a razão de termos considerando apenas um número finito de quantificadores em nossas linguagens de primeira ordem. No caso clássico as funções σ_Q são as seguintes:

	σ_{Q_1}	σ_{Q_2}	σ_{Q_3}	σ_{Q_4}	σ_{Q_5}	σ_{Q_6}	$\sigma_{\forall}$	$\sigma_{\exists}$
(0,1)	1	1	0	0	0	0	1	1
(1,0)	1	1	1	1	0	0	0	0
(1,1)	1	0	1	0	1	0	1	0

e podemos interpretá-los da seguinte maneira:

$(Q_1x)A \equiv (\forall x)A \wedge (\forall x)\neg A$

$(Q_2x)A \equiv (\exists x)A \wedge (\exists x)\neg A$

$(Q_3x)A \equiv \neg(\exists x)A$

$(Q_4x)A \equiv \neg(\forall x)A$

$(Q_5x)A \equiv \neg(Q_2x)A \equiv (\exists x)A \rightarrow (\forall x)A$

$(Q_6x)A \equiv \neg(Q_1x)A.$

COMENTÁRIO 4.3. É claro que em alguns casos poderemos ter um único quantificador primitivo (como no caso clássico) e os demais definidos a partir dele, ou ainda um conjunto de quantificadores primitivos.

No Apêndice I introduzimos as noções de quantificador primitivo e de quantificador definido num cálculo de predicados n-valente e estudamos alguns casos mais simples ($n = 2$ e $n = 3$).

Dizemos que um quantificador $\mathbb{V}$ em uma lógica n-valente é um *quantificador universal*, quando sua função interpretação tem a seguinte propriedade:

$$\sigma_{\forall}(\mathbf{a}_o,\ldots,\mathbf{a}_{n-1}) \in D \quad \text{see} \quad \mathbf{a}_i = \mathbf{0} \quad \text{para} \quad i \in ND$$

(isto é, $v((\forall x)A) \in D$ see para todo $k \in U$, $v(|A|^x_k) \in D$).

Similarmente, $\exists$ é um *quantificador existencial* quando tem a seguinte propriedade:

$$\sigma_{\exists}(\mathbf{a}_o,\ldots,\mathbf{a}_{n-1}) \in D \quad \text{see} \quad \mathbf{a}_i = \mathbf{1} \quad \text{para algum} \quad i \in D.$$

Uma *valorização atômica de primeira ordem* para $S^1(\bar{U})$ é uma assignação de valores-verdade às sentenças atômicas de $S^1(\bar{U})$; assim como no caso proposicional, vale o seguinte

4.2. TEOREMA. *Uma valorização atômica para* $S^1(\bar{U})$ *pode ser estendida a uma única valorização de primeira ordem para* $S^1(\bar{U})$.

DEMONSTRAÇÃO. Por indução sobre o nível de construção das fórmulas, usando as definições V-1 e V-2 para valorizações de primeira ordem e o fato de que cada quantificador possui uma função interpretação. ∎

Seja S^1 o conjunto das fórmulas (não necessariamente fechadas) de L^1; uma *interpretação semântica* de L^1 em um universo $U \neq \emptyset$ é uma função I tal que associa:

I-1) A cada constante individual de S^1 um elemento de U;

I-2) A cada símbolo de predicado m-ário P uma partição do produto cartesiamo U^m e n classes $P^*_0,\ldots,P^*_{n-1}$ correspondentes aos valores verdade chamada P-*partição* de U.

Chamamos ao par $\mathcal{U} = \langle U, I \rangle$ uma *estrutura de primeira ordem* para L^1.

DEFINIÇÃO. Dizemos que uma lógica n-valente de primeira ordem é *regular* quando entre seus quantificadores temos um quantificador universal e um quantificador existencial.

Nossos próximos passos serão as definições de *verdade sob uma interpretação semântica*, *validade* e *satisfação numa estrutura*. O plano que pretendemos seguir é: 1) definir a noção de verdade de uma U-sentença, numa interpretação semântica I; 2) mostrar que equivalem as noções de estrutura e valorização de primeira ordem; 3) definir as noções de satisfação e validade para sentenças em S^1, com ênfase nas sentenças puras; e 4) mostrar que numa lógica regular as definições de satisfação e validade para sentenças em geral se reduzem ao caso das sentenças puras. Estes resultados permitem simplificar bastante os procedimentos do tipo tableau para primeira ordem (que definiremos adiante).

Dada a estrutura $\mathcal{U} = \langle U, I \rangle$, dizemos que uma U-sentença atômica $Pu_1 \ldots u_m$ (isto é, uma sentença atômica de $S^1(U)$) *tem valor* i em $\mathcal{U}$ se a m-upla $(u_1, \ldots, u_n)$ pertence a P^*_i (onde $I(P) = (P^*_o, \ldots, P^*_i, \ldots, P^*_n)$. Se o valor i é um valor distinguido, então dizemos que $Pu_1 \ldots u_m$ é *verdadeira* em $\mathcal{U}$, ou que $\mathcal{U}$ é um *modelo* para $Pu_1 \ldots u_m$.

4.3. TEOREMA. *Uma estrutura* $\mathcal{U} = \langle U, I \rangle$ *define uma única valorização atômica de primeira ordem para* $S^1(\bar{U})$ *e reciprocamente.*

DEMONSTRAÇÃO. Dada a estrutura $\mathcal{U} = \langle U, I \rangle$, podemos construir o conjunto das U-fórmulas fechadas $S^1(\bar{U})$ e pela definição precedente temos uma valorização atômica para $S^1(\bar{U})$; por outro lado, dada uma valorização atômica para $S^1(\bar{U})$, construimos a interpretação semântica I de L^1 para U da seguinte maneira:

I-1) Desde que o conjunto das sentenças $\bar{S}^1$ é enumerável, ordenamos $S^1(\bar{U})$ (eventualmente usando o axioma da escolha, pois U pode ser um conjunto qualquer) e para cada constante que ocorra numa fórmula em um dado nível de construção escolhemos para imagem dessa constante o elemento de U que ocorre na fórmula correspondente; percorrendo todos os níveis, associamos a cada constante um elemento de U;

I-2) Para cada símbolo de predicado m-ário P, construimos a P-partição I(P) do seguinte modo:

$$(u_1, \ldots, u_m) \in P_i^* \quad \text{see} \quad v(Pu_1 \ldots u_m) = i. \blacksquare$$

Dizemos que uma sentença qualquer de $S^1(\bar{U})$ (são necessariamente atômica) tem valor i em $\mathcal{U}$ see assume valor i pela valorização induzida pela valorização atômica; a sentença é verdadeira se $i \in D$.

Seja SP o conjunto das *sentenças puras* de S^1; é claro que SP está contido em $S^1(\overline{U})$, para todo universo U. Dizemos que uma sentença de SP é *satisfatível* se é verdadeira em alguma estrutura $\mathcal{U}$ (equivalentemente, após o teorema 4.3, se existe alguma valorização de primeira ordem v tal que $v(A) \in D$); dizemos que A é *válida* se é verdadeira em toda estrutura $\mathcal{U}$.

De maneira geral, um *subconjunto* S *de* SP *é satisfatível* se existe uma estrutura tal que cada elemento de S é satisfatível nessa estrutura. Dizemos que uma sentença A é *satisfatível num universo* U (respectivamente, *válida num universo* U) se é satisfatível (válida) em toda estrutura que tem U como universo.

Se $A(c_1,\ldots,c_r)$ é uma sentença em $\overline{S}^1$ - SP contendo exatamente as constantes $c_1,\ldots,c_r$, dizemos que $A(c_1,\ldots,c_r)$ é *satisfatível* (*válida*) se existe uma estrutura $\mathcal{U} = \langle U, I\rangle$ tal que a sentença $B = A(I(c_1),\ldots,I(c_r))$ correspondente a $A(c_1,\ldots,c_r)$ em $S^1(\overline{U'})$ (onde $U' = I(CI) \subset U$) é verdadeira (respectivamente, se B é verdadeira para toda estrutura $\mathcal{U}$).

O próximo teorema, cuja prova é omitida por ser rotineira e totalmente análoga à do caso clássico, deixa registrado o fato de que basta considerar sentenças em SP quando se trata de validade e satisfação para lógicas regulares (e mostra que as definições dos quantificadores universal e existencial fazem sentido):

4.4. TEOREMA. *Seja* L^1 *uma lógica* n-*valente regular; então, se* $x_1,\ldots,x_r$ *são variáveis individuais distintas que não ocorrem em*

$A(c_1,\ldots,c_r)$, *e se* $A(x_1,\ldots,x_r)$ *indica a substituição de* c_1 *por* $x_1,\ldots,x_r$ *por* x_r, *temos que:*

1) $A(c_1,\ldots,c_r)$ *é satisfatível see a sentença de* SP $(\exists\!|\, x_1)\ \ldots\ (\exists\!|\, x_r)(A(x_1,\ldots,x_r)$ *é satisfatível; e*

2) $A(c_1,\ldots,c_r)$ *é válida see a sentença de* SP $(\forall\!\!\!\forall\, x_1)\ \ldots\ (\forall\!\!\!\forall\, x_r)(A(x_1,\ldots,x_r))$ *é válida.* ∎

IV.3 - TABLEAUX QUANTIFICACIONAIS

Seja $\bar{S}^1$ o conjunto das sentenças em L^1; denotamos por $\bar{S}^{1*}$ o conjunto das sentenças assignadas de $\bar{S}^1$, isto é,

$$\bar{S}^{1*} = \{a_i(X) : X \in \bar{S}^1 \text{ e } 0 \leq i \leq n-1\}.$$

Se Q é um quantificador de L^1 e σ_Q sua função interpretação dizemos que a m+1-upla $(i, j_1,\ldots,j_m)$ de valores de N *satisfaz a condição quantificacional* para Q (em símbolos, vale a condição $D_i(Q,j_1,\ldots,j_m)$) se a n-upla $\langle d(0),\ldots,d(j_1),\ldots,d(j_m),\ldots,d(n-1)\rangle$, onde $n = |N|$ e $m < n$, definida como $d(t) = 1$ see $t = j_r$ e $d(t) = \mathbf{0}$ caso contrário, pertence a $\sigma_Q^{-1}(i)$.

Em outras palavras, $(i,j_1,\ldots,j_m)$ satisfaz a condição quantificacional para Q se a n-upla que consiste das coordenadas **1** nas posições j_r e **0** nas demais tem imagem i por σ_Q .

A definição de condição quantificacional é fundamental para a definição dos tableaux quantificacionais; vamos discutir, como motivação, estas noções no caso clássico.

Notamos, em primeiro lugar, que da definição das funções $\sigma_\forall$ e $\sigma_\exists$ da Seção V.2, temos:

a) $\sigma_\forall^{-1}(0) = \{(\mathbf{1},\mathbf{0})\}$

b) $\sigma_\forall^{-1}(1) = \{(\mathbf{0},1),(1,1)\}$

c) $\sigma_\exists^{-1}(0) = \{(\mathbf{1},\mathbf{0}),(\mathbf{1},1)\}$

d) $\sigma_\exists^{-1}(1) = \{(\mathbf{0},1)\}$

e, portanto, valem, respectivamente, as seguintes condições quantificacionais:

a) $D_0(\forall;0)$

b) $D_1(\forall;1)$, $D_1(\forall,0,1)$,

c) $D_0(\exists;0,1)$, $D_0(\exists,0)$,

d) $D_1(\exists;1)$,

que significam, do ponto de vista intuitivo, o seguinte:

a) ∀ é verdadeiro somente quandos todas as instâncias o são;

b) ∀ é falso quando todas as instâncias são falsas ou quando tem instâncias verdadeiras e falsas;

c) ∃ é verdadeiro quando todas as instâncias são verdadeiras ou quando tem instâncias verdadeiras e falsas;

d) ∃ é falso quando todas suas instâncias são falsas.

Assim como no caso proposicional, vamos também nos restringir aos conectivos binários, embora a única razão para esta restrição seja a facilidade de leitura e de escrita.

Seja L^1 uma lógica n-valente de primeira ordem fixada; definimos o *tableau quantificacional* da fórmula assignada $a_i(X)$ como a árvore cujo primeiro vértice é $a_i(X)$, construida como se segue:

T-1) Se X é da forma $F(X_1,X_2)$, estendemos a árvore como no caso proposicional, trocando "variável proposicional" por "fórmula atômica" (isto é, usamos as regras π_1/π_2 com as modificações óbvias);

T-2) Se X é da forma (Qx)A, estendemos a árvore de acordo com a regra:

$$\lambda_1/\lambda_2 : \frac{a_i((Qx)A)}{V\{a_{j_1}([A]^x_{c_1}) \wedge a_{j_2}([A]^x_{c_2}) \wedge \cdots \wedge a_{j_m}([A]^x_{c_m})\}},$$

onde a) $j_1,\ldots,j_m < n$ e vale a condição quantificacional $D_i(Q;j_1,\ldots,j_m)$ e

b) se $m \geq 2$ nenhuma das constantes c_r ocorreu anteriormente.

(esta última condição será referida como *substituição cuidadosa*, e sua razão de ser é evitar confronto de instanciações; este cuidado se reduz, no caso $n = 2$, às derivações de $[A]^x_c$ e $\sim[A]^x_c$ a partir de $(\exists x)A$ e $\sim(\forall x)A$).

Os símbolos V e $\wedge$ devem ser entendidos como no caso proposicional;

T-3) As regras π_1/π_2 e λ_1/λ_2 são aplicadas de acordo com o seguinte *procedimento sistemático*:

T-3.1) Iniciamos a árvore com $a_i(X)$; este é o vértice 1.

T-3.2) Se temos o vértice n aplicamos T-1 ou T-2 (isto é, aplicamos as regras π_1/π_2 ou λ_1/λ_2, de acordo com a fórmula X).

T-3.3) Desenvolvemos o tableau proposicional resultante, considerando as fórmulas do tipo $a_j((Qx)Ax)$ como *atômicas* do ponto de vista proposicional, e declaramos o vértice n *usado*.

T-3.4) Caso a regra λ_1/λ_2 tenha sido utilizada em T-3.2, adicionamos o vértice n ao último vértice de cada ramo que passa por n;

T-3.5) Suponhamos que o vértice n tenha sido usado; se todo vértice onde ocorre uma fórmula não atômica já foi usado, o procedimento pára. Caso contrário, tomamos como vértice n+1 o vértice não usado de menor nível e mais à esquerda na árvore.

Dizemos que um ramo de um tableau quantificacional para $a_i(X)$ é *fechado* se existe alguma subfórmula Y de X tal que $a_j(Y)$ e $a_k(Y)$ com $j \neq k$, ocorrem neste ramo ou T-1 ou T-2 não podem ser aplicados. Um tableau quantificacional é *fechado* quando cada ramo for fechado.

Dizemos que uma sentença X de $\bar{S}^1$ é *demonstrável* (ou é *teorema* se para cada $i \in ND$, o tableau quantificacional para $a_i(X)$ é fechado; nesse caso, o conjunto dos tableaux fechados para $a_i(X)$, $i \in ND$, é chamado de *demonstração* para X. Denotamos o fato de X ser teorema em L^1 por $\vdash_{L^1} X$.

Se $B \cup \{X\} \subset \bar{S}^1$, definimos as noções de *consequência sintática* e *consequência semântica* (para primeira ordem) da mesma maneira que as do caso proposicional, substituindo "tableau" por "tableau quantificacional" e "valorização" por "valorização de primeira ordem", respectivamente, e definindo a noção de *tableau quantificacional para um conjunto* $a_{N_O}(B)$ de maneira similar à do caso proposicional com as modificações óbvias. Se X é consequência sintática (semântica) de B, denotamos essa relação entre B e X por $B \vdash_{L^1} X$ $(B \models_{L^1} X)$.

COMENTÁRIO 4.4. EXEMPLOS DE APLICAÇÃO DOS TABLEAUX QUANTIFICACIONAIS.

Vamos considerar exemplos de aplicações dos tableaux para a lógica clássica de primeira ordem e para o cálculo trivalente de primeira ordem utilizado por D'Ottaviano em [7]; é óbvio que não é necessário utilizar as regras de derivação de tableau para ramos fechados, de acordo com a definição de demonstração.

Abreviamos, nos tableaux, as expressões $|A|^x_a$ por Aa, para facilidade de leitura.

EXEMPLO 1 - CASO CLÁSSICO. Sejam $Q_1 = \forall$, $Q_2 = \exists$ e $0 =$ verdadeiro; vamos dar uma demonstração para

$(\forall x)(P(x) \wedge Q(x) \supset (\exists x)(P(x) \vee Q(x))$ (P,Q atômicas)

É fácil ver, pelo ítem T-2 da definição de tableau quantificacional, que as regras λ_1/λ_2 nesse caso são:

R-1) $$\frac{a_o((\forall x)A)}{a_o(Ac)}$$

R-2) $$\frac{a_1((\forall x)A)}{\{a_1(Ac)\} \vee \{a_o(Ac_2) \wedge a_1(Ac_3)\}}$$

R-3) $$\frac{a_1((\exists x)A)}{a_1(Ac)}$$

R-4) $$\frac{a_o((\exists x)A)}{\{a_o(Ac)\} \vee \{a_o(Ac_2) \wedge a_1(Ac_3)\}}$$

onde nenhuma das constantes indexadas ocorreu anteriormente.

A demonstração consiste em mostrar que o tableau quantificacional para $a_1(X)$ é fechado, onde X é a sentença em questão:

(1) $a_1 [(\forall x)(Px \wedge Qx) \supset (\exists x)(Px \vee Qx)]$

(2) $a_0 [(\forall x)(Px \wedge Qx)]$ (de (1), aplicando T-3.2)

(3) $a_1 [(\exists x)(Px \vee Qx)]$ (idem)

(4) $a_0 [Pa \wedge Qa]$ (de (2), aplicando T-3.2)

(5) $a_0 [Pa]$ (de (4), aplicando T-3.3)

(6) $a_0 [Qa]$ (idem)

(7) [2] (de (2), aplicando T-3.4)

(8) $a_1 [Pb \vee Qb]$ (de (3), aplicando T-3.2)

(9) $a_1 [Pb]$ (de (8), aplicando T-3.3)

(10) $a_1 [Qb]$ (idem)

(11) $a_0 [Pb \wedge Qb]$ (de (7), aplicando T-3.2)

(12) $a_0 [Pb]$ (de (11), aplicando T-3.3)

(13) $a_0 [Qb]$ (idem).

X

OBSERVAÇÕES. A regra R-1 foi aplicada em (2) e (7), gerando os vértices (4) e (11); a regra R-3 foi aplicada em (3), gerando o vértice (8) .

O tableau é fechado em virtude de (9) e (12) (e também de (10)e(13)).

EXEMPLO 2. Vamos demonstrar outro teorema, no caso clássico:

$$(\forall y)((\forall x)Px \supset Py)$$

Nesse caso, o tableau terá ramificações por envolver as condições quantificacionais $D_1(\forall;1)$, $D_1(\forall;0,1)$.

O símbolo [i] indica repetição do vértice em (i)

(1) $a_1((\forall y)((\forall x)Px \supset Py))$

(2) $a_1((\forall x)Px \supset Pa)$ (3) $a_0((\forall x)Px \supset Pb)$

(4) $a_1((\forall x)Px \supset Pc)$

(5) $a_0((\forall x)Px)$

(6) $a_1(Pa)$

(7) [1]

(14) $a_0(Pa)$

X

(9) $a_1((\forall x)Px)$

(8) $a_0(Pb)$ (11) $a_0((\forall x)Px)$

X

(10) $a_0((\forall x)Px)$

(12) $a_1(Pc)$

(13) [1]

(15) $a_0(Pc)$

X

DESCRIÇÃO DOS PASSOS DO TABLEAU:

a) Iniciamos o tableau com o vértice (1);

b) Aplicamos R-2 a (1); obtemos (2), (3) e (4);

c) A partir de (2), por T-1, obtemos (5) e (6); o vértice (2) está usado;

d) O vértice (7) vem de (1), por T-3.4;

e) Os vértices (8) e (9) vem de (3), que é o primeiro vértice não usado;

f) Os vértices (10), (11), (12) vêm de (4), por T-1;

g) O vértice (13) vêm de (1), por T-3.4;

h) Os vértices (14) e (15) vêm de (7) e (13), respectivamente, por R-1.

EXEMPLO 3. Seja J_3 o cálculo trivalente de primeira ordem estudado em [7], cujos conectivos e quantificadores são descritos pelas seguintes funções:

$\vee$	0	1	2
0	0	1	2
1	1	1	2
2	2	2	2

$\supset$	0	1	2
0	2	2	2
1	0	1	2
2	0	1	2

$\wedge$	0	1	2
0	0	0	0
1	0	1	1
2	0	1	2

	∇
0	0
1	2
2	2

	$\urcorner$
0	2
1	1
2	0

$N = \{0,1,2\}$

$D = \{1,2\}$

0	1	2	$\exists$	$\forall$
1	**0**	**0**	0	0
0	**1**	**0**	1	1
0	**0**	**1**	2	2
1	**1**	**0**	1	0
1	**0**	**1**	2	0
0	**1**	**1**	2	1
1	**1**	**1**	2	0

isto é, $\exists$ e $\forall$ são interpretados pelas seguintes funções, respectivamente:

$$\sigma_{\exists}(\langle d(0),\ d(1),\ d(2)\rangle) = \max\{i : d(i) \neq \mathbf{0}\}$$

$$\sigma_{\forall}(\langle d(0),\ d(1),\ d(2)\rangle) = \min\{i : d(i) \neq \mathbf{0}\}$$

e as condições quantificacionais são as seguintes (agrupadas para referência nos tableaux):

$\exists_0$: $D_o(\exists;0)$

$\exists_1$: $D_1(\exists;1)$, $D_1(\exists;0,1)$

$\exists_2$: $D_2(\exists;2)$, $D_2(\exists;0,2)$, $D_2(\exists;1,2)$, $D_2(\exists;0,1,2)$

$\forall_0$: $D_o(\forall;0)$, $D_o(\forall;0,1)$, $D_o(\forall;0,2)$, $D_o(\forall;0,1,2)$

$\forall_1$: $D_1(\forall;1)$, $D_1(\forall;1,2)$

$\forall_2$: $D_2(\forall;2)$.

e, portanto, as regras λ_1/λ_2 para J_3 são as seguintes:

J-1) $$\frac{a_o((\exists x)A)}{a_o(Ac)}$$

J-2) $$\frac{a_1((\exists x)A)}{\{a_1(Ac)\} \vee \{a_o(Ac_1) \wedge a_1(Ac_2)\}}$$

J-3) $$\frac{a_2((\exists x)A)}{\{a_2A_c\} \vee \{a_oAc_1 \wedge a_2Ac_2\} \vee \{a_1Ac_3 \wedge a_2Ac_4\} \vee \{a_oAc_5 \wedge a_1Ac_6 \wedge a_2Ac_7\}}$$

J-4) $$\frac{a_o((\forall x)A)}{\{a_oAc\} \vee \{a_oAc_1 \wedge a_1Ac_2\} \vee \{a_oAc_3 \wedge a_2Ac_4\} \vee \{a_oAc_5 \wedge a_1Ac_6 \wedge a_2Ac_7\}}$$

J-5) $$\frac{a_1((\forall x)A)}{\{a_1(Ac)\} \vee \{a_1(Ac_1) \wedge a_2(Ac_2)\}}$$

J-6) $$\frac{a_2((\forall x)A)}{a_2(Ac)}$$

onde nenhuma das constantes indexadas ocorrem anteriormente.

As regras π_1/π_2 são claras a partir das tábuas.

Vamos dar uma demonstração para:

$(\forall x)\ \nabla(Ax \wedge Bx) \supset (\exists x)\ (\nabla Ax \wedge \nabla Bx)$, ($A$ e B atômicas):

(1) $a_o((\forall x)\nabla(Ax \wedge Bx) \supset (\exists x)(\nabla Ax \vee \nabla Bx)$

(2) $a_1((\forall x)\nabla(Ax \wedge Bx)$ (3) $a_2((\forall x)\nabla(Ax \wedge Bx))$

(4) $a_o((\exists x)(\nabla Ax \vee \nabla Bx))$ (5) [4]

(6) $a_1(\nabla(Aa \wedge Ba)$ (7) $a_1(\nabla(Ab \wedge Bb))$

X

(8) $a_2(\nabla(Ac \wedge Bc))$

X

(9) $a_2(\nabla(Aa \wedge Ba))$

(10) $a_1(Aa \wedge Ba)$ (11) $a_2(Aa \wedge Ba)$

(12) $a_1(Aa)$ (13) $a_1(Aa)$ (14) $a_2(Aa)$ (18) $a_o(Aa)$

(15) $a_1(Ba)$ (16) $a_2(Ba)$ (17) $a_1(Ba)$ (19) $a_2(Ba)$

(20) [3] [3] [3] [3]

(24) $a_o(\nabla Aa \vee \nabla Ba)$ [24] [24] [24]

(28) $a_o(\nabla Aa)$ [28] [28] [28]

(32) $a_o(\nabla Ba)$ [32] [32] [32]

(36) $a_o(Aa)$ [36] [36] [36]

(40) $a_o(Ba)$ [40] [40] [40]

X X X X

DESCRIÇÃO DO TABLEAU

a) Iniciamos o tableau com o vértice 1;

b) Os vértices (2), (3), (4) e (5) se obtém de (1), por T-3.2;

c) Os vértices (6), (7) (8) se obtem de (2), por T-3.2, aplicando a regra J-5;

d) Os ramos que passam por (6) e (7) são fechados porquê não existe condição proposicional $H_1(V;i)$;

e) O vértice (9) vem de (3), por T-3.2 e pela regra J-6;

f) Os vértices (10) a (19) vêm de (9), por T-3.3;

g) Os vértices (20) a (23) vêm de (3), por T-3.4;

h) O vértice (24) vem de (5), por T-3.2 e regra J-1;

i) Os demais vértices se obtêm de (24), por T.3.3;

j) Os vértices (15) e (40) fecham o ramo que passa por (40); os demais ramos são fechados por (16) e (41), (17) e (42) e (18) e (43).

EXEMPLO 4. Vamos considerar, neste exemplo, um tableau que não fecha: se tentamos demonstrar, em J_3, $(\exists x)(Ax \wedge VAx)$, temos o seguinte tableau:

(1) $a_o((\exists x)(Ax \wedge \nabla Ax))$

(2) $a_o(Aa \wedge \nabla Aa)$

(3) $a_o(A_a)$

(6) [1]

.
.
.

$a_o(Ab)$

.
.
.

[1]

.
.
.

(4) $a_o(\nabla Aa)$

(5) $a_o(Aa)$

(7) [1]

.
.
.

O vértice (2) se obtém de (1), por T-3.2 e J-1; os vértices (3) a (5) vêm de (2) por T-3.3. O vértice (6) e (7) vêm de (1), por T-3.4. Os dois ramos que passam por (6) e (7) são abertos. Fica claro que a repetição de (1) permitirá obter $a_o(Ac)$, para toda constante c.

Esta é a razão da aplicação de T-3.4: conforme veremos adiante, a repetição de vértices permite obter, no caso de tableaux abertos, uma valorização de primeira ordem que dá o valor 0 à fórmula testada. Como 0 é um valor não distinguido, a fórmula não será teorema.

CAPÍTULO V

APLICAÇÕES DOS TABLEAUX QUANTIFICACIONAIS

V.1 - TEOREMAS DE COMPLETUDE E COMPACIDADE.

Definimos *conjunto de Hintikka quantificacional para um universo* U como sendo um conjunto H de U-sentenças assignadas (isto é, $H \subset S^1(\bar{U})^*$), tais que as seguintes condições são satisfeitas:

C_1) Para toda fórmula atômica X de $S^1(\bar{U})$, e para todo $k, j < n$, $k \neq j$, se $a_k(X) \in H$, então $a_j(X) \notin H$;

C_2) Se $Y = a_i(F(Y_1, Y_2) \in H$, então alguma consequência de $\pi_1/\pi_2(Y)$ está contido em H .

C_3) Se $a_i(Qx)A \in H$, então existem $j_1, \ldots, j_m < n$ tais que:

a) Vale a condição quantificacional $D_i(Q, j_1, \ldots, j_m)$.

b) Cada fórmula $[A]^x_c$, $c \in U$, está em H assignada por algum símbolo a_{j_r} , $1 \leq r \leq m$.

c) Para cada j_r, $1 \leq r \leq m$, existe c em U tal que $a_{j_r}([A]^x_c) \in H$.

COMENTÁRIO 5.1. Para enfatizar a dependência do valor de n (conjunto de valores-verdade) nesta definição, poderíamos chamar o conjunto H de conjunto de Hintikka quantificacional *n-dimensional;*

nesse caso, o leitor poderá se convencer de que esta definição generaliza propriamente o caso clássico (2-dimensional) se considerar as condições quantificacionais para $\forall$ e $\exists$.

Se B é um conjunto de U-fórmulas asssignadas, definimos o *traço* de B, T(B), como o seguinte subconjunto de N : $T(B)=\{i \in N:$ existe X tal que $a_i(X) \in B\}$.

Se A é um conjunto de fórmulas não assignadas, dizemos que um elemento X de A é i-*valorizável* se existe uma valorização de primeira ordem v tal que $v(X) = i$; dizemos que A é N_o-*valorizável* se existe uma valorização tal que cada X em B é i-valorizável para $i \in N_o \subset N$.

Se A é um conjunto de fórmulas não assignadas, denotamos por $A_{N_o}(A)$, $N_o \subset N$, o conjunto de assignações de fórmulas de A tal que:

1) $T(a_{N_o}(A)) = N_o$; e

2) Se $a_i(X)$ e $a_j(X) \in a_{N_o}(A)$, $i = j$

(dito de outra maneira, $a_{N_o}(A)$ é a imagem de uma função $F : N_o \times A \longrightarrow S^{1*}$ tal que $F(i,X) = a_i(X)$.

Se B é um conjunto de fórmulas assignadas chamamos de *suporte* de B ao conjunto $SP(B) = \{X \in S^1$: existe $i < n$ tal que $a_i(X) \in B\}$.

5.1. TEOREMA. *Sejam* H *um conjunto de* U*-sentenças assignadas com traço* T(H), *e* H_o *o suporte de* H; *se* H *é um conjunto de Hintikka, existe uma valorização* v *tal que* $v(X) = i$ *se* $a_i(X) \in H$ (*e consequentemente* H_o *é* T(H)*-valorizável*).

DEMONSTRAÇÃO. Construimos uma valorização v por indução sobre o nível de construção das fórmulas da seguinte maneira:

1) Se X é uma fórmula atômica de $S^1(\bar{U})$ e se $a_i(X) \in H$, então $v(X) = i$; se $X \notin SP(H)$ assignamos um valor arbitrário a $v(X)$;

2) Pelo teorema 4.2, esta valorização atômica pode ser estendida a uma única valorização (e, portanto, podemos denotar a extensão pelo mesmo símbolo v);

3) Por indução no nível de construção, temos, para fórmulas X em H_o:

3.1) Se X é de nível 0 (atômica) é claro que X é T(H)-valorizável, e que $v(X) = i$ se $a_i(X) \in H$.

3.2) Suponhamos que X tenha nível positivo e que toda fórmula de H_o de nível menor é tal que $v(Y) = i$ se $a_i(Y) \in H$:

a) Se X é $F(Y_1, Y_2)$ e $a_i(X) \in H$, pela definição de conjunto de Hintikka quantificacional existe alguma conclusão de $\pi_1/\pi_2\ (a_i(X))$ da forma $A\{a_{j_i} Y_i\}$ contida em H; por hipótese de indução, $v(Y_i) = j_i$ e pela definição da regra π_1/π_2 do tableau, e pela cláusula V-1 da definição de valorização de primeira ordem, $v(F(Y_1, Y_2)) = i$.

b) Se X é (Qx)A, e $a_i(X) \in H$, pela condição C_3 vale a condição quantificacional $D_i(Q, j_1, \ldots, j_m)$, para cada j_r existe uma fórmula $a_{j_r}([A]^x_c)$ em H (e, portanto, por hipótese de indução, uma fórmula tal que $v([A]^x_c) = j_r$) e ainda todas as fórmulas $a_{j_r}([A]^x_c)$ estão em H (e, portanto, por hipótese de indução todas as fórmulas desse tipo têm valores j_r).

Pela cláusula V_2 da definição de valorização de primeira ordem, então, $v(X) = i$.

É claro que H_o é T(H)-valorizável. ∎

Vamos demonstrar, em seguida, o teorema de completude e consistência para lógicas n-valentes de primeira ordem; antes, porém, demonstraremos um resultado um pouco mais geral relacionando existência de tableaux abertos e existência de valorizações de primeira ordem. O teorema de completude e consistência obtêm-se como caso particular, à semelhanca do que ocorre no caso proposicional (Capítulo II, Seção II.1).

5.2. TEOREMA. a) *Se X é uma sentença em* $S^1(\overline{U})$, *então X é i-valorizável see existe um tableau quantificacional aberto para* $a_i(X)$;

b) *Seja B um conjunto de sentenças em* $S^1(\overline{U})$; *se* $N_o \subset N$ *e f e i são funções tais que*

$$f(X)=a_{i(X)}(X), \text{ onde } X \in B \text{ e } i(X) \in N_o$$

então existe uma valorização de primeira ordem v tal que $v(X)=i(X)$, *para todo* X *em* B *see existe um tableau quantificacional aberto para o conjunto* $\{a_{i(X)}(X): X \in B\}$.

DEMONSTRAÇÃO. a) Suponhamos que exista uma valorização de primeira ordem tal que $v(X) = i$; se construirmos a árvore de subfórmulas para X, AS(X) (nesse caso, tomando o cuidado de tomar as constantes em U), a existência da valorização nos permitirá assignar cada vértice com a_j se o valor do vértice pela valorização é j; é claro que dessa forma cumprem-se as condições para aplicação das regras de derivação π_1/π_2 e λ_1/λ_2 e conseguimos um tableau aberto para $a_i(X)$ (que será, dependendo das regras π_1/π_2, uma subárvore de AS(X)).

Por outro lado, se existe um tableau quantificacional aberto para $a_i(X)$, é fácil ver que o conjunto dos vértices deste tableau é um conjunto de Hintikka quantificacional, devido à cláusula T-3.4 da definição de tableau quantificacional. Logo, pelo teorema 5.1, existe uma valorização de primeira ordem v tal que $v(X) = i$.

b) Se existe uma valorização v nas condições acima, então pela parte (a) existe um tableau aberto para cada fórmula assignada $a_i(X)$, $X \in B$; suponhamos que o tableau para o conjunto $\{a_i(X) : X \in B\}$ seja fechado; nesse caso, existe uma fórmula de B (ou subfórmula de fórmula de B) que ocorre no tableau com assignações distintas; desde que v é uma valorização, isto é impossível, logo o tableau é aberto.

Reciprocamente, se existe um tableau quantificacional aberto para o conjunto $\{a_i(X) : X \in B\}$, então existe pelo menos um ramo aberto que contém como vértices todos os elementos desse conjunto;

este ramo, claramente, é um conjunto de Hintikka quantificacional, e pelo teorema 5.1 obtém-se uma valorização v tal que $v(X) = i(X)$ para todo $X \in B$. ∎

5.3. COLORÁRIO. (Teorema da Consistência e Completude). *Seja* $B \cup \{X\}$ *um conjunto de sentenças puras; então* $B \vdash_{L^1} X$ *see* $B \models_{L^1} X$.

DEMONSTRAÇÃO. Suponhamos que $B \not\models_{L^1} X$; se D é o conjunto de valores distinguidos de L^1, seja v uma valorização de primeira ordem tal que $\{v(Y) : Y \in B\} \subset D$ e $v(X) = i \notin D$.

Pelo teorema 5.2, existe um tableau aberto para o conjunto $\{a_{v(Y)}(Y) : Y \in B\} \cup \{a_i(X)\}$ e portanto $B \not\vdash_{L^1} X$.

Por outro lado, se $B \not\vdash_{L^1} X$, existe um tableau aberto para algum conjunto $a_D \cup \{a_i(X)\}$, onde $i \in ND$ e $a_D = \{a_j(Y) : Y \in B$ e $j \in D\}$.

Pelo teorema 5.2, há uma valorização v tal que $v(B) \subset D$ e $v(X) \notin D$, logo $B \not\models_{L^1} X$. ∎

COMENTÁRIO 5.2. Como consequência do teorema anterior, obtemos que toda fórmula válida tem uma prova finita; de fato, se X é válida, para todo $i \in ND$, o tableau para $a_i(X)$ é fechado, ou seja, tem todos os ramos fechados, e, portanto, finitos. Pelo lema de König, o tableau não pode ser infinito, pois teria um ramo infinito.

Como aplicações do Teorema da Completude, podemos mostrar que o valor de verdade que uma sentença assume numa lógica n-valente de primeira ordem não depende das particulares variáveis que a sentença contenha:

5.4. COROLÁRIO. (Teorema da Variante). *Seja* $s : VI \longrightarrow VI$ *uma pré-substituição que leva variáveis em variáveis, e* Sb_s *a substituição em primeira ordem relativa a* s *(ver Seção IV.I) e sejam* $A' = Sb(A)$ *e* $x' = s(x)$, *onde* x *é uma variável e* A *uma fórmula pura; então*

$$\vdash_{L}^{1} (Qx)A \quad \textit{see} \quad \vdash_{L}^{1} (Qx')A' .$$

DEMONSTRAÇÃO. De acordo com o teorema de Completude, basta mostrar que, para toda valorização de primeira ordem v, $v((Qx)A) \in D$ see $v((Qx')A') \in D$; em primeiro lugar, é fácil ver, por indução sobre a construção das fórmulas, que

$$[A]_a^x = [A']_a^{x'}$$

para toda constante a num universo U. Desde que $v((Qx)A)=\sigma_Q(D(A,x,v))$, onde $D(A,x,v) = \langle d(0), d(1), \ldots, d(n-1) \rangle$ e $d(i) = \mathbf{1}$ see existe $k \in U$ tal que $v([A]_k^x) = i$, e $d(i) = \mathbf{0}$ em caso contrário (conforme definição de valorização de primeira ordem, Seção 4.2), pela observação anterior temos que $D(A,x,v) = D(A',x',v')$ e, portanto,

$$v((Qx)A) = v((Qx')A') ,$$

para toda valorização de primeira ordem v, e pelo teorema de completude, a demonstração está completa. ∎

V.2 - PRINCÍPIO DA UNIFICAÇÃO PARA LÓGICAS DE PRIMEIRA ORDEM

No Capítulo II, Seção II.3, descrevemos o Princípio da Unificação para lógicas proposicionais n-valentes, e mostramos como os teoremas de completude e compacidade, no caso proposicional, podem se obter diretamente a partir deste Princípio. Neste parágrafo vamos demonstrar uma versão do Princípio de Unificação para lógicas de primeira ordem e derivar os teoremas de compacidade e Löwenheim-Skolem para lógicas regulares; de acordo com o teorema 4.4, Seção 4.2, é suficiente considerar sentenças puras (para questões que envolvam as noções de validade e satisfação) em lógicas regulares; por esta razão, as hipóteses dos teoremas a seguir referem-se somente a conjuntos de sentenças puras. Sugerimos ao leitor que neste ponto, compare as definições a seguir com as do caso proposicional (Seção II.3).

Seja Γ um conjunto de conjuntos de sentenças em $S^1(\bar{U})^*$, (isto é, de sentenças assignadas com parâmetros em um universo U) e suponhamos que Γ tem caráter finito. Se $K \in \Gamma$, dizemos que K *tem a propriedade* Γ.

Se B é um conjunto de sentenças em $S^1(\bar{U})$ (de sentencas sem assignação) dizemos que B é Γ-*consistente* se existe $K \in \Gamma$ tal que $K = \{a_i(X) : X \in B \text{ e } i \in D\}$.

Assim como no caso do Princípio da Unificação proposicional, (Seção II.3), a expressão "B é Γ-consistente" significa, do ponto de vista intuitivo, que Γ garante que B é consistente, como demonstramos no teorema 5.6 a seguir. Assim, Γ será uma Propriedade de Consistência Analítica quando; 1) não contém sentenças atômicas com valores "contraditórios", e 2) se uma sentença A ocorre em Γ com um certo valor de verdade, suas subsentenças ocorrem também em Γ com os valores tais que justificam o valor com que A comparece em Γ.

Dizemos que Γ é uma *propriedade de consistência analítica* (abreviadamente, PCA) quando todo conjunto K que tem a propriedade Γ satisfaz às seguintes condições:

P_o) K não contém $a_i(X)$ e $a_j(X)$, se $i \neq j$ e X é uma sentença atômica de $S^1(\overline{U})$;

P_1) Se $a_i(F(Y_1,Y_2)) \in K$, existe alguma consequência $C = \pi_1/\pi_2(a_iF(Y_1,Y_2))$ tal que $K \cup C$ tem a propriedade Γ.

P_2) Se $a_i(Qx)A \in K$, existem $j_1,\ldots,j_m < n$ tais que vale a condição quantificacional $D_i(Q;\ j_1,\ldots,j_m)$ e o conjunto $Z = \{a_{j_r}([A]^x_c) : c \in U\}$ (onde para cada j_r, $1 \leq r \leq m$, existe pelo menos um $c \in U$ tal que $a_{j_r}([A]^x_c) \in Z$) é tal que $K \cup Z \in \Gamma$.

5.5. TEOREMA (Princípio da Unificação para Lógicas n-Valentes de Primeira Ordem). *Seja* B *um conjunto de sentenças puras*, Γ *uma* PCA; *se* B *é* Γ-*consistente, então todos os elementos de* B *são satisfatíveis num universo enumerável.*

DEMONSTRAÇÃO. Suponhamos, em primeiro lugar, que B seja enumerável (o caso finito é trivial a partir do caso enumerável).

Desde que B é Γ-consistente, existe um conjunto $a_D(B)=\{a_i(X): X \in B$ e $i \in D\}$ que tem a propriedade Γ.

Como o conjunto $a_D(B)$ é enumerável, seja $Z_1, Z_2, \ldots, Z_n, \ldots,$ uma enumeração para esse conjunto. O plano da demonstração é construir um conjunto de Hintikka quantificacional M, definindo a sequência M_i como segue:

1) Tomamos $M_1 = Z_1$; este é o primeiro passo na construção de M_1;

2) Se temos o termo M_n, onde os termos da sequência formam um conjunto finito que tem a propriedade Γ, suponhamos que Y é o n-ésimo termo:

   a) Se Y é da forma $a_i(F(Y_1,Y_2))$, estendemos a sequência colocando $Y, \wedge\{a_{j_i}(Y_i\}, Z_{n+1}$ se $C = \wedge\{a_{j_i}(Y_i)\}$ é uma consequência de $\pi_1/\pi_2(a_iF(Y_1,Y_2))$ tal que P-1 é satisfeita; a sequência resultante tem pelo menos n+1 termos, e tem a propriedade Γ, por P_1.

   b) Se Y é da forma $a_i(Qx)A$ e se $j_1,\ldots,j_m$ são tais que vale a condição P_2 da definição de PCA, estendemos a sequência anexando os seguintes termos: Y, $a_{j_1}(Aa_1)$, $a_{j_2}(Aa_2),\ldots,a_{j_m}(Aa_m), Y, Z_{n+1}$, onde $a_1,a_2,\ldots,a_m$ são constantes tais que:

b-1) Se $m \geq 2$, $a_1, a_2, \ldots, a_m$ não ocorreram anterior - mente na sequência;

b-2) Se $m = 1$, Aa_1 não ocorreu anteriormente na sequência.

O conjunto M dos termos da sequência, obviamente, tem a propriedade Γ, desde que cada M_i tem a propriedade Γ e Γ tem caráter finito.

É fácil ver, também, que M é um conjunto de Hintikka quantificacional: de fato, desde que cada M_i tem a propriedade Γ , M_i não pode conter fórmulas com assignações distintas, e M satisfaz a condição C_1 da definição de conjunto de Hintikka quantificacional, para o universo U das constantes envolvidas no processo de construção de M.

As outras duas condições (C_2 e C_3) da definição de conjunto de Hintikka quantificacional são claramente satisfeitas.

Pelo teorema 5.1 (Seção V.1), se M_o é o suporte de M, existe uma valorização v para M_o, e esta valorização, obviamente, satisfaz todas as fórmulas de $B \subset M_o$, (desde que as assignações com que B comparece no conjunto M são distinguidas) no universo enumerável das constantes individuais. ∎

COMENTÁRIO 5.3. A construção envolvida no Teorema 5.6 é a construção do tableau quantificacional aberto, com pequenas modificações (ver Seção IV.3).

A partir do Princípio da Unificação (Teorema 5.5), podemos obter outra prova para o Teorema da Completude:

5.6. COROLÁRIO. (Completude). *Seja* B *um conjunto de sentenças puras; então,* B *é satisfatível see existe um tableau quantificacional aberto para algum* $a_D(B)$.

DEMONSTRAÇÃO. Por um lado, se B é satisfatível, é fácil ver que existe um tableau quantificacional aberto para $a_D(B)$; basta tomar cada fórmula X de B assignada por a_i onde $i = v(X)$; pela definição de valorização (de primeira ordem) e de tableau quantificacional, a construção é imediata,

Por outro lado, seja $S^{1*}(U)$ o conjunto das U-fórmulas assignadas, e seja Γ o seguinte conjunto: $\Gamma = \{K \subseteq S^{1*}(U)$: existe tableau quantificacional para o conjunto $K\}$.

É claro que tem caráter finito e que é uma PCA; se existe um tableau aberto para $a_D(B)$, B é Γ-consistente, e, pelo Princípio da Unificação, B é satisfatível (num universo enumerável). ∎

5.7. COROLÁRIO. (Teorema da Compacidade). *Se* B *for um conjunto de sentenças puras, todo subconjunto finito* B_O *de* B *é satisfatível see* B *é satisfatível num universo enumerável.*

DEMONSTRAÇÃO. Suponhamos que todo subconjunto finito B_O de B seja satisfatível; pelo Teorema da completude, existe um tableau aberto para algum conjunto $a_D(B_O)$; utilizando o mesmo argumento do

Lema 2.2 (Seção II.1), temos que existe um tableau aberto para $a_D(B)$ see para cada B_o finito contido em B existe um tableau aberto para $a_D(B_o)$.

O resultado segue-se, então, diretamente do teorema da Completude. ∎

COMENTÁRIO 5.4. Considerando a construção similar à do lema 2.2 referida na demonstração do corolário anterior, e o Teorema da Completude, é claro que valem, também, as seguintes formas do Teorema da Compacidade (a prova é totalmente análoga à do caso proposicional): Se B é um conjunto de sentenças puras, e X é uma sentença pura, $B \vdash_{\overline{L}}^{1} X$ see existe $B_o \subset B$, B_o finito, tal que $B_o \vdash_{\overline{L}}^{1} X$. (e $B \models_{\overline{L}}^{1} X$ see existe $B_o \subset B$, B_o finito, tal que $B_o \models_{\overline{L}}^{1} X$).

A partir do Corolário 5.8, deduz-se imediatamente o Teorema de Löwenheim-Skolem:

5.8. COROLÁRIO. (Teorema de Löwenheim-Skolem). *Seja* B *um conjunto de sentenças puras: se* B *for satisfatível,* B *é satisfatível num universo enumerável.*

DEMONSTRAÇÃO. Se B for satisfatível, todo subconjunto finito B' de B é satisfatível; pelo Corolário 5.7, B é satisfatível num universo enumerável. ∎

V.3 - COMPARAÇÃO ENTRE LÓGICAS n-VALENTES DE PRIMEIRA ORDEM

Seja n fixado e S^1 uma linguagem n-valente de primeira ordem; sejam L e L' dois cálculos de predicados n-valentes de primeira ordem determinados pela linguagem S^1 e por interpretações distintas para algum (ou todos) de seus conectivos e quantificadores ; isto é, (ver Seção IV.2):

$$L = S^1, A, \langle \{\sigma_Q : Q \text{ é um quantificador em } S^1\} \rangle$$

e

$$L' = \langle S^1, A', \{\sigma'_Q : Q \text{ é um quantificador em } S^1\} \rangle$$

Nesta seção, retomando, de certa forma, o tema do Capítulo III, mostramos que, se as interpretações dos conectivos e quantificadores nas duas lógicas guardam certas relações a respeito dos velores de verdade, as lógicas podem ser comparadas no seguinte sentido: se $X \in S^1$, então $\vdash_L X$ see $\vdash_{L'} X$.

Seja $F_k = \{C_o, C_1, \ldots, C_{k-1}\}$ uma partição de N tal que $C_i \subset D$ ou $C_i \subset ND$; se Q,F,G são, respectivamente, quantificadores, conectivos binários e conectivos unários na linguagem, definimos os vetores $\langle a_o, a_1, \ldots, a_{k-1} \rangle$, onde $a_i = \square$ ou $a_i = \bot$:

1) $\langle a_o, a_1, \ldots, a_{k-1} \rangle = \{\langle d(0), d(1), \ldots, d(n-1) \rangle$: para cada $a_i \neq \square$ existe j em C_i tal que $d(j) \neq \mathbf{0}\}$;

Definimos, então, extensões $\sigma_Q\langle a_o, a_1, \ldots, a_{k-1} \rangle$, $f(C_i, (j)$ e $g(C_i)$ das funções σ_{Q_1} , f e g como segue:

2) $\sigma_Q\langle a_o, a_1, \ldots, a_{k-1}\rangle = \{\sigma_Q(s) : s \in \langle a_o, a_1, \ldots, a_{k-1}\rangle\}$;

3) $f(C_i, C_j) = \{f(x,y) : x \in C_i \text{ e } y \in C_j\}$;

4) $g(C_i) = \{g(x) : x \in C_i\}$.

Se $R,S \subset N$, escrevemos $R \sim S$ para denotar que existe i tal que $R \subset C_i$ e $S \subset C_i$; se $R = \{i\}$ e $S = \{j\}$ abreviamos $R \sim S$ por $i \sim j$. Dizemos que P_k é uma *partição quantificacional para* N *com respeito a* L *e* L' se:

1) Para cada Q em S^1, $\sigma_Q\langle a_o, a_1, \ldots, a_{k-1}\rangle \sim \sigma'_Q\langle a_o, a_1, \ldots, a_{k-1}\rangle$ para toda sequência de símbolos $\Box$ e 1 com comprimento k;

2) Para cada F em S^1, $f(C_i, C_j) \sim f'(C_i, C_j)$ para todo i,j entre 0 e $k-1$, onde $f \in A$ e $f' \in A'$; e

3) Para cada G em S^1, $g(C_i) \sim g'(C_i)$, para todo i entre 0 e $k-1$, onde $g \in A$ e $g' \in A'$.

Se valem estas relações, escrevemos $\sigma_Q \simeq \sigma'_Q$, $f \simeq f'$, e $g \simeq g'$, respectivamente, e dizemos que estes conectivos e quantificadores são *congruentes por* P_k.

5.9. TEOREMA. *Se* $X \in S^1$ *e se existe uma partição quantificacional para* N *com respeito a* L *e* L', *então* $\vdash_L X$ *see* $\vdash_{L'} X$.

DEMONSTRAÇÃO. Seja P_k a partição quantificacional para N; vamos mostrar primeiramente que, dada qualquer valorização de primeira ordem v para L, existe uma valorização de primeira ordem v' para

L' tal que $v(X) \sim v'(X)$.

De fato, dada a valorização v para L, relativa a um universo U, seja v uma valorização atômica para L' tal que $v'(Y) = v(Y)$ para toda fórmula atômica Y em S^1. Pelo Teorema 4.2 (Seção IV.2) v' pode ser estendida a uma valorização de primeira ordem, e temos que:

1) Se X é atômica, obviamente $v(X) \sim v'(X)$;

2) Se X é da forma $F(X_1,X_2)$, $v(F(X_1,X_2)) = f(v(X_1),v(X_2))$; por hipótese de indução no nível de construção das fórmulas, temos

$$v(X_1) \sim v'(X_1) \text{ e } v(X_2) \sim v'(X_2).$$

Portanto , $f(v(X_1),\ v(X_2)) \sim f'(v'(X_1),\ v'(X_2))$, pois $f \simeq f'$, por hipótese, e daí $v(F(X_1,X_2)) \sim v'(F(X_1,X_2))$.

3) Se X é da forma G(X) a prova á análoga.

4) Se X é da forma (Qx)A, por hipótese de indução, temos $v([A]^x_k) \sim v'([A]^x_k)$, para todo $k \in U$. Como $\sigma_Q \sim \sigma'_Q$, é fácil ver que $v((Qx)A) \sim v'((Qx)A)$.

Portanto, pelo Teorema de completude, se $\nvdash_L X$, existe v para L tal que $v(X) \in ND$, e então, existe v' para L' tal que $v'X) \in ND$, pela definição de partição quantificacional; resulta, usando novamente o teorema de completude que $\nvdash_{L'} X$.

Demonstramos, então, que $\vdash_{L'} X$ implica $\vdash_L X$; a recíproca é análoga. ■

Como exemplo, consideremos o cálculo trivalente de primeira ordem J_3 descrito em [7], cujos conectivos e quantificadores dados pelas seguintes funções:

$\vee$	0	1	2
0	0	1	2
1	1	1	2
2	2	2	2

	∇	$\neg$
0	0	2
1	2	0
2	2	0

0	1	2	$\sigma_\exists$	$\sigma_\forall$
1	0	0	0	0
0	1	0	1	1
0	0	1	2	2
1	1	0	1	0
1	0	1	2	0
0	1	1	2	1
1	1	1	2	0

$N = \{0,1,2\}$

$D = \{1,2\}$

Seja $S^{1'}$ a sublinguagem de J_3 consistindo de todos os símbolos da linguagem de J_3 exceto $\neg$, e sejam $\vee'$, ∇', $\sigma'_\exists$, $\sigma'_\forall$ novas interpretações para os símbolos de $S^{1'}$ dados da seguinte maneira:

$\vee'$	0	1	2
0	0	1	1
1	1	1	1
2	1	1	1

	∇'
0	0
1	1
2	1

0	1	2	$\sigma'_\exists$	$\sigma'_\forall$
1	0	0	0	0
0	1	0	1	1
0	0	1	1	1
1	1	0	1	0
1	0	1	1	0
0	1	1	1	1
1	1	1	1	0

$N = \{0,1,2\}$

$D = \{1,2\}$

Se designamos por J_3^o e $J_3^{o'}$ as lógicas de primeira ordem obtidas, e se X é uma fórmula de J_3 onde não ocorre o símbolo $\neg$, então $\vdash_{J_3^o} X$ see $\vdash_{J_3^{o'}} X$; pelo Teorema 5.9, basta tomar a partição $P_2 = \{\{0\}, \{1,2\}\}$ e mostrar que é uma partição quantificacional para N com respeito a J_3^o e $J_3^{o'}$; de fato, temos que

1) $(\perp,\Box) = \{(1,0,0)\}$

$(\Box,\perp) = \{(0,1,0), (0,0,1), (0,1,1)\}$;

$(\perp,\perp) = \{(1,1,0), (1,0,1), (1,1,1)\}$;

e $\sigma_\exists(a,b) \sim \sigma'_\exists(a,b)$, $\sigma_\forall(a,b) \sim \sigma'_\forall(a,b)$ para todo

$(a,b) \in \{(\perp,\Box), (\Box,\perp), (\perp,\perp)\}$;

2) $V(C_i,C_j) \sim V'(C_i,C_j)$ para todo $i,j \in \{0,1\}$;

3) $\nabla(C_i) \sim \nabla'(C_i)$ para todo $i \in \{0,1\}$.

APÊNDICE I

COMPLETUTE QUANTIFICACIONAL E GERAÇÃO DE QUANTIFICADORES.

AI-1) COMPLETUDE QUANTIFICACIONAL

No capítulo IV, ao definirmos a contrapartida semântica dos quantificadores de nossas linguagens n-valentes de primeira ordem como as *funções de distribuição*, notamos que o fato de haver um quantificador primitivo tal que os demais possam ser definidos a partir dele constitui uma propriedade que pode se estender, ademais do caso clássico n = 2, para as lógicas n-valentes de primeira ordem em geral.

Em termos formais, se L^1 é uma lógica n-valente de primeira ordem fixada e Q é um quantificador em L, podemos identificar o quantificador Q com sua função distribuição σ_Q; desse modo, se K é um conjunto de quantificadores, dizemos que Q é *gerado pelo conjunto* K se existem funções unárias $f : N \longrightarrow N$ e binárias $g : N \times N \longrightarrow N$ (i.e., conectivos unários e binários de L^1) tais que $(Qx)A$ pode ser escrito de uma das seguintes maneiras, onde A é uma fórmula e Q_1, Q_2 pertencem a K:

a) $(Qx)A = f(Q_1x)A$

b) $(Qx)A = g((Q_1x)A, (Q_2x)A)$

c) $(Qx)A = (Q_1x)(fA)$.

Para simplificar a notação, escreveremos, respectivamente,

a) $Q = fQ_1$

b) $Q = g(Q_1, Q_2)$

c) $Q = Q_1 f$.

Se o conjunto K gera todos os quantificadores de L dizemos que K é um *conjunto completo de quantificadores*; se $\{Q_1\}$ é um conjunto completo de quantificadores, dizemos que Q_1 é um *quantificador completo*. L é *quantificacionalmente completo* quando possui um quantificador completo.

O problema da completude quantificacional consiste em determinar quantos e quais são os quantificadores completos; além de ser um problema de interesse próprio, a presença de um quantificador completo assegura quantificadores existenciais e universais numa lógica n-valente de primeira ordem, isto é, assegura que a lógica seja *regular* (ver definição na Seção IV.2).

A caracterização dos quantificadores completos, contudo, parece ser um problema de dificuldade semelhante à dos conectivos completos, resolvido para o caso das lógicas trivalentes por N.M. Martin (ver [12] e [13]). Para os demais casos, o problema da completude funcional está resolvido apenas parcialmente, envolvendo técnicas de natureza algébrica e combinatória bastante sofisticadas; dois dos trabalhos que consideramos dentre os mais importantes nessa área, são ([18] e [21]), e que dão uma idéia clara das dificuldades envolvidas.

Desde que o número de conectivos (unários e binários) numa lógica n-valente é n^{n^2} e o número de quantificadores é $n^{(2^n-1)}$, a dificuldade no caso dos quantificadores não é surpreendente, já que este último número cresce mais rapidamente que o primeiro.

Neste parágrafo estudaremos os casos n = 2 e n = 3 (parcialmente) do problema de completude quantificacional; vamos demonstrar, antes, um lema para o caso geral; lembramos que Q é uma função de V_2^N em N (isto é, Q é uma função que associa a cada n-upla de **0**'s e **1**'s um valor entre 0 e n-1, exceto à n-upla 000...00, onde Q não está definido); recordamos também que se v é uma valorização de primeira ordem, A uma fórmula e x uma variável,

$$D_{x,v}(A) = \langle d(0), d(1), \ldots, d(n-1)\rangle \in V_2^N ,$$

onde $d(i) = \mathbf{1}$ se existe k tal que $v([A]^x_k) = i$ e

$d(i) = \mathbf{0}$ caso contrário;

e que $v((Qx)A) = Q(D_{x,v}(A))$.

De acordo com a definição de IV.2, Capítulo IV, deveríamos escrever $\sigma_Q(D_{x,v}(A))$ ao invés de $Q(D_{x,v}(A))$; estamos usando, porém, o mesmo símbolo para Q e σ_Q conforme observação no início da Seção AI-1.

Definimos o *comprimento* de um vetor w em V_2^N, l(w), como o número de coordenadas de w que são iguais a **1**. Se V_r é o conjunto de vetores de comprimento r, podemos escrever V_2^N como a união disjunta $V_2^N = \bigcup_{r=1}^{N} V_r$.

O próximo lema diz respeito às propriedades do ítem (c) da definição de quantificadores gerados a partir de outros quantificadores:

1. LEMA. *Seja* Q *um quantificador*, f *um conectivo unário; então:*

a) Qf *é um quantificador dado por*

$Qf = Q(\lambda_f) : V_2^N \longrightarrow N$ onde $\lambda_f : V_2^N \longrightarrow V_2^N$ *é tal que*

$\lambda_f\langle d(0),\ d(1),\ldots,d(n-1)\rangle = \langle b_o,\ldots,b_{n-1}\rangle$ *onde*

$b_i = \mathbf{1}$ *see existe* j *tal que* $f(j) = i$ *e* $d(j) = \mathbf{1}$;

b) *Se* $w \in V_2^N$, *então* $1(\lambda_f(w)) \leq 1(w)$, *e a igualdade vale para todo* w *see* f *é uma permutação dos valores de* N;

e

c) *Para cada* $V_r \subset V_2^N$, $Qf(V_r) \subset \bigcup_{s\leq r} Q(V_s)$.

DEMONSTRAÇÃO: a) Desde que, pela parte (c) da definição, temos que $(Qfx)A = (Qx)fA$, se $\langle d(0),d(1),\ldots,d(n-1)\rangle = \langle b_o,b_1,\ldots,b_{n-1}\rangle$, então $b_i = \mathbf{1}$ see existe k tal que $v(f[A]_x^k) = i$, see $f(v([A]_x^k) = i$, see existe j tal que $f(j) = i$ e $v([A]_x^k) = j$, see $d(j) = \mathbf{1}$.

b) Seja $w \in V_r \subset V_2^N$; se $d(i_1),\ldots,d(i_r)$ são as coordenadas de w iguais a $\mathbf{1}$, se todos os valores $f(i_1),f(i_2),\ldots,f(i_r)$ forem distintos, então $\lambda_f(w)$ terá r coordenadas iguais a $\mathbf{1}$, pela parte a), e $1(\lambda_f(w)) = 1(w)$; se nem todos esses valores forem distintos, $1(\lambda_f(w)) < 1(w)$.

É claro que vale a igualdade para todo w see f é uma permutação.

c) Óbvio, a partir de (b). ∎

Para exemplo de cálculos de quantificadores utilizando o lema 1, ver AI.2.

2. TEOREMA. *Seja* $n = 2,3$ *ou* 4, Q *um quantificador em* L, *e suponhamos que* L^1 *seja funcionalmente completo; então, se existe* $f : N \longrightarrow N$ *tal que para todo* $w \in V_2^N$ *os pares ordenados* $(Q(w)),\ Qf(w))$ *são todos distintos,* Q *é um quantificador completo.*

DEMONSTRAÇÃO. Sejam $(a_i, b_i) = (Q(w_i),\ Qf(w_i))$, para $i=1,2,\ldots,2^n-1$, onde $n = 2,3$ ou 4. Se $\bar{Q} : V_2^N \longrightarrow N$ é um quantificador, como L^1 é funcionalmente completo existe uma função binária $g : N \times N \longrightarrow N$ tal que $g(a_i, b_i) = \bar{Q}(w_i)$ pois para $n = 2,\ 3,$ e 4 temos que $2^n - 1 < n^2$; portanto, $g(Q, Qf) = \bar{Q}$ e Q é um quantificador completo. ∎

Como exemplo, suponhamos que L^1 seja uma lógica trivalente funcionalmente completa e Q é um quantificador dado por:

	0	1	2		Q
V_1	1	0	1	$\longrightarrow$	0
	0	1	0		1
	0	1	0		2
V_2	1	1	0		0
	1	0	1		2
	0	1	1		1
V_3	1	1	1		2

Se f é o conectivo unário em L^1 dado pela função

	f
0	0
1	2
2	1

temos, pelo lema 1, que o quantificador Qf é descrito por:

$$\begin{array}{llll} 1\ 0\ 0 & \xrightarrow{Q_f} & 0 & \text{e os pares } (Q(w),\ Qf(w)) \text{ são } (0,0) \\ 0\ 1\ 0 & & 2 & (1,2) \\ 0\ 0\ 1 & & 1 & (2,1) \\ 1\ 1\ 0 & & 2 & (0,2) \\ 1\ 0\ 1 & & 0 & (2,0) \\ 0\ 1\ 1 & & 1 & (1,1) \\ 1\ 1\ 1 & & 2 & (2,2) \end{array}$$

Como L^1 é funcionalmente completa, podemos escolher conecti — vos binários adequados para gerar, como imagem destes pares, todas as funções de V_2^N em N, onde $N = \{0,1,2\}$.

Vamos introduzir uma notação especial para $n = 3$, da seguinte maneira: seja $V_2^N = V_1 \cup V_2 \cup V_3$, onde $N = \{0,1,2\}$ e $V_1 = \{1\ 0\ 0,\ 0\ 1\ 0,\ 0\ 0\ 1\}$ $V_2 = \{1\ 1\ 0,\ 1\ 0\ 1,\ 0\ 1\ 1\}$ $V_3 = \{1\ 1\ 1\}$ e $Q : V_2^N \longrightarrow N$ um quantificador; nesse caso,as restrições de Q a V_1, V_2 e V_3 serão denotadas por, respectivamente, $Q'V_1$, $Q'V_2$ e $Q'V_3$, e diremos que $Q'V_i$ $(i = 1,2)$ tem *tipo* I se a imagem de $Q'V_i$ apresenta apenas um valor; *tipo* II, se apresenta dois valores distintos; e *tipo* III, se apresenta três valores distintos. Diremos que Q é de tipo I-I, I-II, I-III, II-I, II-II, II-III, III-I, III-II, ou III-III conforme sejam os tipos de $Q'V_1$ e $Q'V_2$.

3. TEOREMA. *Se Q é um quantificador de tipo III-III numa lógica trivalente funcionalmente completa, Q é um quantificador completo.*

DEMONSTRAÇÃO. Se a,b,c indicam os valores 0,1,2 numa permutação qualquer, então os quantificadores podem ser classificados dentro das seguintes possibilidades:

				A	B	C	D	E	F
	1	**0**	**0**	a	a	a	a	a	a
V_1	**0**	**1**	**0**	b	b	b	b	b	b
	0	**0**	**1**	c	c	c	c	c	c
	1	**1**	**0**	a	a	c	b	c	b
V_2	**1**	**0**	**1**	b	c	b	a	a	c
	0	**1**	**1**	c	b	a	c	b	a
V_3	1	1	1	d	d	d	d	d	d

Consideremos as seguintes funções unárias de N em N:

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8
0	0	1	1	2	2	1	0	0
1	2	2	0	1	0	0	1	0
2	1	0	2	0	1	0	0	1

Utilizando o Lema 1, se Q é um quantificador de uma das formas A - F (que chamaremos de $Q_A, Q_B, \ldots, Q_F$), podemos escolher f_i adequados e calcular $Q_X f_i$, onde $X \in \{A, B, \ldots, F\}$ e $i \in \{1, 2, \ldots, 8\}$, de maneira tal que para todo $w \in V_2^N$, os pares $(Q_X(w), Q_X f_i(w))$ sejam distintos; pelo Teorema 2, cada Q_X será um quantificador completo.

Resumimos, abaixo, a escolha dos f_i e o cálculo de cada $Q_X f_i$ (subentendendo V_1, V_2 e V_3 com seus elementos na mesma ordem em que estão colocados no início da demonstração):

$Q_A f_2$ (d qualquer)	$Q_B f_1$ (para $d = c$)	$Q_B f_3$ (para $d = b$)	$Q_B f_4$ (para $d = a$)
b	a	b	c
c	c	a	b
a	b	c	a
c	c	a	b
a	a	b	c
b	b	c	a
d	c	b	a

$Q_C f_6$ (para $d = a$)	$Q_C f_7$ (para $d = b$)	$Q_C f_8$ (para $d = c$)
b	a	a
a	b	a
a	a	b
c	c	a
c	a	c
a	c	c
c	c	c

$Q_D f_1$ (para $d=b$)	$Q_D f_3$ (para $d=a$)	$Q_D f_4$ (para $d=c$)
a	b	c
c	a	b
b	c	a
a	b	c
b	c	a
c	a	b
b	a	c

$Q_E f_2$ (para d qualquer)	$Q_F f_5$ (para d qualquer)
b	c
c	a
a	b
b	a
c	b
a	c
d	d

É fácil ver que todas as condições do Teorema 2 são satisfeitas, e que, portanto, todos estes são quantificadores completos. ∎

O teorema a seguir é uma espécie de recíproca do Teorema 2:

4. TEOREMA. *Sejam* $n=2,3$ *ou* 4, $Q: V_2^N \longrightarrow N$ *um quantificador e* w, w' *vetores fixos de* V_2^N; *se para todo conectivo* $f_j: N \longrightarrow N$ *os pares ordenados* $(Qf_i(w), Qf_j(w))$ *e* $(Qf_i(w'), Qf_j(w'))$ *são iguais, Q não é um quantificador completo.*

DEMONSTRAÇÃO. Basta notar que, se g é um conectivo binário, teremos $g(f_1Qf_2(w),\ f_3Qf_4(w)) = g(f_1Qf_2(w'),\ f_3Qf_4(w'))$, (onde f_1, f_2, f_3 e f_4 são conectivos unários arbitrários) e que as cláusulas (a), (b) e (c) da definição de quantificadores gerados podem ser obtidas desta expressão tomando-se f_i's adequados.

Portanto, qualquer quantificador gerado terá valores iguais em w e w' e obviamente Q não é um quantificador completo. ∎

5. COROLÁRIO. *Seja $n = 2, 3$ ou 4 e $V_1 \subset V_2^N$; se Q é um quantificador tal que $Q'V_1$ é a constante, Q não é completo.*

DEMONSTRAÇÃO. Pelo Lema 1, parte (c), temos que $Qf'V_1 \subset Q'V_1$ para todo conectivo unário f; como $Q'V_1$ é constante, $Qf'V_1$ é constante e há pelo menos dois vetores w e w' tais que as condições do teorema 4 são satisfeitas; portanto Q não é completo. ∎

6. COROLÁRIO. *Os únicos quantificadores clássicos completos são $\forall, \exists, \neg\forall$ e $\neg\exists$.*

DEMONSTRAÇÃO. Por um lado, sabemos que estes quantificadores são completos, pois qualquer deles com a negação gera todos os oito quantificadores clássicos (ver Seção IV.2). Por outro lado, desde que $V_1 = \{1\ 0,\ 0\ 1\}$ neste caso, é fácil ver que todos os outros quantificadores são tais que $Q'V_1$ é constante; pelo Corolário 5, portanto, nenhum outro quantificador é completo. ∎

7. TEOREMA. *Seja* L^1 *uma lógica trivalente e* Q *um quantificador ; se* Q *é de tipo* I-I, I-II, I-III, *ou* III-I, Q *não é completo.* ∎

DEMONSTRAÇÃO. (CASO 1). Q é de tipo I-I, I-II ou I-III; nesse caso, $Q'V_1$ é constante, e o resultado se segue diretamente do Corolário 5.

(CASO 2). Q é de tipo II-I ou III-I; nesse caso $Q'V_2$ é constante, e temos três possibilidades para $f : N \longrightarrow N$:

2.1) f é constante; então, $Qf'V_2$ é constante também.

2.2) f é uma permutação (i.e., f é bijetora); pelo lema 1, parte (b), $Qf'V_2$ tem os mesmos valores que $Q'V_2$, ou seja, $Qf'V_2$ é constante.

2.3) A imagem de f tem exatamente dois valores; logo f tem uma das seguintes formas:

	f_A	f_B	f_C
0	a	b	b
1	b	a	b
2	b	b	a

onde $a,b \in N = \{0,1,2\}$

Se f é da forma f_A, então pelo Lema 1, parte (a), Qf leva os vetores de V_2 **1 1 0** e **1 0 1** em vetores de V_2 (analogamente, se f é da forma f_B, leva **1 1 0** e **0 1 1** em vetores de V_2, e se f é da forma f_C leva **1 0 1** e **0 1 1** em vetores de V_2).

Portanto, de qualquer modo $Qf'V_2$ terá necessariamente dois valores de $Q'V_2$.

Concluimos, pois, que em qualquer dos casos 2.1, 2.2, ou 2.3 haverá dois vetores de V_2, w e w', tais que para quaisquer f_i, f_j, $(Qf_i(w), Qf_j(w))$ e $(Qf_i(w'), Qf_j(w'))$ são iguais; pelo Teorema 4, Q não é completo. ∎

Para os demais casos (i.e., quantificadores trivalentes de tipos II-II, II-III e III-II), não dispomos ainda de um resultado geral; contudo, há diversos casos particulares (que não trataremos aqui para não sobrecarregar o trabalho) que nos levam a conjecturar que todos os quantificadores destes tipos são completos em lógicas funcionalmente completas.

AI.2 - EXEMPLO DE CÁLCULO DE GERAÇÃO DE QUANTIFICADORES

Como ilustração do Lema 1, podemos tomar como exemplo o seguinte quantificador ∃ (descrito em [7]):

0	1	2	$\rightarrow$ ∃
1	0	0	0
0	1	0	1
0	0	1	2
1	1	0	1
1	0	1	2
0	1	1	2
1	1	1	2

Se ∇, $\neg$ são conectivos unários dados pelas tábuas: ($\nabla\neg$, $\neg\nabla$, e $\neg\nabla\neg$ são computados, obviamente, a partir de ∇ e $\neg$)

	∇	$\neg$	$\nabla\neg$	$\neg\nabla$	$\neg\nabla\neg$
0	0	2	2	2	0
1	2	1	2	0	0
2	2	0	0	0	2

então, pelo Lema 1, parte (a), as funções λ terão como imagem:

V_2^N		$\lambda_\nabla(V_2^N)$	$\lambda_\neg(V_2^N)$	$\lambda_{\nabla\neg}(V_2^N)$	$\lambda_{\neg\nabla}(V_2^N)$	$\lambda_{\neg\nabla\neg}(V_2^N)$
1 0 0	$\longrightarrow$	1 0 0	0 0 1	0 0 1	0 0 1	1 0 0
0 1 0		0 0 1	0 1 0	0 0 1	1 0 0	1 0 0
0 0 1		0 0 1	1 0 0	1 0 0	1 0 0	0 0 1
1 1 0		1 0 1	0 1 1	0 0 1	1 0 1	1 0 0
1 0 1		1 0 1	1 0 1	1 0 1	1 0 1	1 0 1
0 1 1		0 0 1	1 1 0	1 0 1	1 0 0	1 0 1
1 1 1		1 0 1	1 1 1	1 0 1	1 0 1	1 0 1

Portanto, os quantificadores $\exists\nabla$, $\exists\neg$, $\exists\nabla\neg$, $\exists\neg\nabla$ e $\exists\neg\nabla\neg$ são da seguinte forma

		$\exists\nabla$	$\exists\neg$	$\exists\nabla\neg$	$\exists\neg\nabla$	$\exists\neg\nabla\neg$
1 0 0	$\longrightarrow$	0	2	2	2	0
0 1 0		2	1	2	0	0
0 0 1		2	0	0	0	2
1 1 0		2	2	2	2	0
1 0 1		2	2	2	2	2
0 1 1		2	1	2	0	2
1 1 1		2	2	2	2	2

A partir destes, utilizando as funções ∇, $\neg$, $\nabla\neg$, $\neg\nabla$ e $\neg\nabla\neg$ podemos gerar os seguintes quantificadores:

Notações

$\neg^* = \neg$

$\Delta = \neg\nabla\neg$

$\Diamond = \nabla\neg$

TABELA 1

$\neg\exists\nabla$	$\neg\exists\neg$	$\neg\exists\Diamond$	$\neg\exists\neg^*$	$\neg\exists\Delta$	$\nabla\exists\nabla$	$\nabla\exists\neg$	$\nabla\exists\Diamond$	$\nabla\exists\neg^*$	$\nabla\exists\Delta$	$\neg^*\exists\nabla$
2	0	0	0	2	0	2	2	2	0	2
0	1	0	2	2	2	2	2	0	0	0
0	2	2	2	0	2	0	0	0	2	0
0	0	0	0	2	2	2	2	2	0	0
0	0	0	0	0	2	2	2	2	2	0
0	1	0	2	0	2	2	2	0	2	0
0	0	0	0	0	2	2	2	2	2	0

$\neg^*\exists\neg$	$\neg^*\exists\Diamond$	$\neg^*\exists\neg^*$	$\neg^*\exists\Delta$	$\Delta\exists\nabla$	$\Delta\exists\neg$	$\Delta\exists\Diamond$	$\Delta\exists\neg^*$	$\Delta\exists\Delta$	$\Diamond\exists\nabla$
0	0	0	2	0	2	2	2	0	2
0	0	2	2	2	0	2	0	0	0
2	2	2	0	2	0	0	0	2	0
0	0	0	2	2	2	2	2	0	0
0	0	0	0	2	2	2	2	2	0
0	0	2	0	2	0	2	0	2	0
0	0	0	0	2	2	2	2	2	0

$\Diamond\exists\neg$	$\Diamond\exists\Diamond$	$\Diamond\exists\neg^*$	$\Diamond\exists\Delta$	$\neg\exists$	$\nabla\exists$	$\neg^*\exists$	$\Delta\exists$	$\Diamond\exists$
0	0	0	2	2	0	2	0	2
0	0	2	2	1	2	0	0	2
2	2	2	0	0	2	0	2	0
0	0	0	2	1	2	0	0	2
0	0	0	0	0	2	0	2	0
0	0	2	0	0	2	0	2	0
. 0	0	0	0	0	2	0	2	0

Por simples inspeção na Tabela 1, fica claro que valem as igualdades seguintes:

a) $\exists\nabla = \nabla\exists = \nabla\exists\nabla = \Delta\exists\nabla$

b) $\exists\Diamond = \nabla\exists\neg = \nabla\exists\Diamond = \Delta\exists\Diamond$

c) $\exists\neg^* = \nabla\exists\neg^* = \Delta\exists\neg = \Delta\exists\neg^*$

d) $\exists\Delta = \nabla\exists\Delta = \Delta\exists\Delta = \Delta\exists$

e) $\neg\exists\nabla = \neg^*\exists\nabla = \Diamond\exists\nabla = \neg^*\exists$

f) $\neg\exists\Diamond = \neg^*\exists\neg = \neg^*\exists\Diamond = \Diamond\exists\neg = \Diamond\exists\Diamond$

g) $\neg\exists\neg^* = \neg^*\exists\neg^* = \Diamond\exists\neg^*$

h) $\neg\exists\Delta = \neg^*\exists\Delta = \Diamond\exists\Delta = \Diamond\exists$

($\neg\exists\neg$ coincide com a definição de $\forall$ dada em [7]).

Portanto, para qualquer conectivo $g(x,y)$ tal que $g(x,x)$ dê como resultado valores distinguidos, estas serão teoremas (notamos que em [7] os conectivos $\supset$, $\rightarrow$, $\equiv$, $\equiv_{\mathcal{L}}$, $\equiv^{*}$, $\equiv'_{\mathcal{L}}$ e $\equiv^{*}_{\mathcal{L}}$ têm esta propriedade).

Observamos que esses são todos os quantificadores possíveis de serem definidos a partir de $\exists$ utilizando $\neg$, ∇, Δ, $\neg^{*}$ e $\Diamond$ uma vez que este conjunto de conectivos unários é fechado por composição, e que em [7] o quantificador $\forall$ é considerado primitivo, assim como algumas das equivalências acima são consideradas como axiomas.

APÊNDICE II

UMA VERSÃO COMPUTACIONAL DO TABLEAU SINTÁTICO PARA O CÁLCULO TRIVALENTE P^1.

PASCAL COMPILATION LIST PRODUCED BY PASCAL VERSION FROM 30-DEC-76
ON 30-JUN-82 AT 11:19:44

```
PROGRAM GETEO;
(*ESSE PROGRAMA GERA PROPOSICOES E PROVA SE SAO TEOREMAS
 OU NAO*)
                    REGRAS DE FORMACAO:
  1-GERA PXI, I=0.
  2-ESCREVE A0X0,A1X0,A2X0.
  3-A-TENDO PY1,PY2, PRODUZA QY1=>QY2.
    B-TENDO PY, PRODUZA QNY.
  4-TENDO QY, PRODUZA PY E APAGUE QY.
  5-TENDO AI1Y1, AI2Y2, PRODUZA CF(I1,I2) (Y1=>Y2) E
   TENDO AIY, PRODUZA CG(I)NY.
    ONDE F  0 1 2         F     G  0 |2|
          0 |0|0|2|               1 |0|
          1 |0|0|2|               2 |0|
          2 |0|0|0|
  6-TENDO CIY, PRODUZA AIY E APAGUE CIY.
  7-PARA CADA Y, FACA:
    SE A2Y NAO EXISTE, ENTAO ESCREVA TY,
                         SENAO NAO E TEOREMA.
  8-PRODUZA PXI+1, E VOLTE AO PASSO 1.                    *)
TYPE
TAB=RECORD
(*A=TABELA DA VALIDADE DE A0,A1,A2.
  PROP=CODIGO DA PROPOSICAO.
  TEO=VERDADEIRO, SE A PROPOSICAO E TEOREMA*)
    A:ARRAY[0..2] OF BOOLEAN;
    PROP:ARRAY[1..3] OF INTEGER;
    TEO:BOOLEAN
    END;

(*ABRE=INDICA '(' DETERMINADO.
  FECHA=SE FECHA OU NAO ESSE PARENTESIS.*)
TP=RECORD
    ABRE:INTEGER;
    FECHA:BOOLEAN
    END;

VAR
(*TABPROP=TABELA DAS PROPOSICOES GERADAS.
  VETOR=DESCODIFICACAO DE 1 PROPOSICAO.*)
DIG:ARRAY[1..3] OF INTEGER;
VETOR:ARRAY[1..300] OF CHAR;
TABPROP:ARRAY[1..2000] OF TAB;
TPAR:ARRAY[1..100] OF TP;
IND,IND1,IND2:INTEGER;
JJ,II,I,J,L,K,VA,B,D:INTEGER;
TOPO,TOPO1ANT,TOPO2ANT:INTEGER;

PROCEDURE INTCAR(K:INTEGER);
(*TRANSFORMA UM INTEIRO EM UMA SEQUENCIA DE
  CARACTERES QUE E POSTO EM DIG *)
VAR
*AUX,DIVISAO:INTEGER;
BEGIN
(*SE K NAO PODE TER MAIS DE 3 DIGITOS*)
DIVISAO:=K DIV 10;
IF DIVISAO=0 THEN BEGIN
                         DIG[J]:=K;
```

```
                      J:=J+1
                      END
                 ELSE BEGIN
                      KAUX:=K MOD 10;
                      DIG[J]:=KAUX;
                      K:=(K-KAUX) DIV 10;
                      J:=J+1;
                      INTCAR(K)
                      END
END;  (*PROCEDURE INTCAR*)

PROCEDURE DESCOD(N:INTEGER);
VAR
M:INTEGER;
(*CADA PROPOSICAO PODE SER FORMADA PELO SEGUINTE
  CODIGO:
   N=NUMERO DA PROPOSICAO, NA TABELA
   [A1 A2 A3]
   SE E UMA PXI, ENTAO TERA O CODIGO [-1|I|0]
   SE E UMA IMPLICACAO: ELEMENTO 1=>ELEMENTO 2,TERA O CODIGO
                              [-1 |ELEM. 1| ELEM. 2]
   SE E UMA NEGACAO: N ELEMENTO, TERA O CODIGO
                           [0 | ELEM. | 0]
  ESSE PROCEDIMENTO DESCODIFICA ESSES CODIGOS,
  COLOCANDO PARENTESIS QUANDO NECESSARIO*)
BEGIN
CASE TABPROP[N].PROP[1] OF    (*IMPLICA*)
 1:BEGIN
   VETOR[I]:='(';
   IND:=IND+1;
   TPAR[IND].ABRE:=I;
   TPAR[IND].FECHA:=TRUE;
   I:=I+1;
   DESCOD(TABPROP[N].PROP[2]);
   IF TPAR[IND].FECHA THEN BEGIN
                           VETOR[I]:=')';
                           I:=I+1
                           END;
   IND:=IND-1;  (*OPERADOR DE IMPLICA*)
   VETOR[I]:='=';
   VETOR[I+1]:='>';
   I:=I+2;
   (*SEGUNDO MEMBRO*)
   VETOR[I]:='(';
   IND:=IND+1;
   TPAR[IND].ABRE:=I;
   TPAR[IND].FECHA:=TRUE;
   I:=I+1;
   DESCOD(TABPROP[N].PROP[3]);
   IF TPAR[IND].FECHA THEN BEGIN
                           VETOR[I]:=')';
                           I:=I+1
                           END;
   IND:=IND-1
   END;
 0:BEGIN  (*NEGACAO*)
   IF I<>1 THEN BEGIN
                VETOR[I-1]:='N';
                TPAR[IND].FECHA:=FALSE
                END
```

```
                ELSE BEGIN
                       VETOR[I]:='N';
                       I:=I+1
                       END;
        VETOR[I]:='(';
        IND:=IND+1;
        TPAR[IND].ABRE:=1;
        TPAR[IND].FECHA:=TRUE;
        I:=I+1;
        DESCOD(TABPROP[N].PROP[2]);
        IF TPAR[IND].FECHA THEN BEGIN
                                  VETOR[I]:=')';
                                  I:=I+1
                                 END;
        IND:=IND-1
    END;
    -1:BEGIN
        IF I<>1 THEN BEGIN
                       VETOR[I-1]:='X';
                       TPAR[IND].FECHA:=FALSE
                       END
                ELSE BEGIN
                       VETOR[I]:='X';
                       I:=I+1
                       END;
        J:=1;
        INTCAR(TABPROP[N].PROP[2]);
        FOR M:=J-1 DOWNTO 1 DO BEGIN
                                 CASE DIG[M] OF
                                 0:VETOR[I]:='0';
                                 1:VETOR[I]:='1';
                                 2:VETOR[I]:='2';
                                 3:VETOR[I]:='3';
                                 4:VETOR[I]:='4';
                                 5:VETOR[I]:='5';
                                 6:VETOR[I]:='6';
                                 7:VETOR[I]:='7';
                                 8:VETOR[I]:='8';
                                 9:VETOR[I]:='9'
                                 END;
                               I:=I+1
                                 END
        END
    END  (*CASE*)
    END;  (*PROCEDURE DESCOD*)
    BEGIN  (*PROGRAMA PRINCIPAL*)
    (*INICIALIZACAO DE VARIAVEIS*)
    TOPO:=0;
    (*CABECALHO*)
    WRITELN('TEOREMAS :');
    WRITELN;
    (*COMECO DA GERACAO/VERIFICACAO DE PROPOSICOES/
    TEOREMAS*)
    FOR K:=0 TO 2 DO BEGIN
    (*GERACAO DA PROPOSICAO INICIAL*)
    TOPO:=TOPO+1;
    TABPROP[TOPO].PROP[1]:=-1;
    TABPROP[TOPO].PROP[2]:=K;
    TABPROP[TOPO].PROP[3]:=0;
    FOR JJ:=0 TO 2 DO TABPROP[TOPO].A[JJ]:=TRUE;
```

```
TABPROP[TOPO].TEO:=FALSE;
(*DETERMINACAO DO VALOR DE TOPO1ANT,TOPO2ANT*)
IF K<>0 THEN BEGIN
               TOPO2ANT:=TOPO1ANT+1;
               TOPO1ANT:=TOPO
                  END
           ELSE BEGIN  (*K=0*)
               TOPO2ANT:=1;
                  TOPO1ANT:=1
               END;
(*GERACAO DAS PROPOSICOES 'IMPLICA' E DA 'NEGACAO'*)
(*PREENCHIMENTO DE A0,A1,A2 INICIAIS DE CADA NOVA
  PROPOSICAO='F' E DE TEO="1"*)
FOR VA:=TOPO1ANT DOWNTO TOPO2ANT DO
  BEGIN
  TOPO:=TOPO+1;
  TABPROP[TOPO].PROP[1]:=1;
  TABPROP[TOPO].PROP[2]:=VA;
  TABPROP[TOPO].PROP[3]:=VA;
  FOR JJ:=0 TO 2 DO TABPROP[TOPO].A[JJ]:=FALSE;
  TABPROP[TOPO].TEO:=TRUE;
  TOPO:=TOPO+1;
  TABPROP[TOPO].PROP[1]:=0;
  TABPROP[TOPO].PROP[2]:=VA;
  TABPROP[TOPO].PROP[3]:=0;
  FOR JJ:=0 TO 2 DO TABPROP[TOPO].A[JJ]:=FALSE;
  TABPROP[TOPO].TEO:=TRUE;
  IF K<>0 THEN FOR B:=VA-1 DOWNTO 1 DO
                  BEGIN
                  TOPO:=TOPO+1;
                  TABPROP[TOPO].PROP[1]:=1;
                  TABPROP[TOPO].PROP[2]:=VA;
                  TABPROP[TOPO].PROP[3]:=B;
                  FOR JJ:=0 TO 2 DO TABPROP[TOPO].A[JJ]:=FALSE;
                  TABPROP[TOPO].TEO:=TRUE;
                  TOPO:=TOPO+1;
                  TABPROP[TOPO].PROP[1]:=1;
                  TABPROP[TOPO].PROP[2]:=B;
                  TABPROP[TOPO].PROP[3]:=VA;
                  FOR JJ:=0 TO 2 DO TABPROP[TOPO].A[JJ]:=FALSE;
                  TABPROP[TOPO].TEO:=TRUE
                  END  (*FOR B*)
END;  (*FOR VA*)
(*VERIFICACAO DE TEOREMAS*)
FOR II:=1 TO TOPO DO
 BEGIN
 IF TABPROP[II].PROP[1]=1 THEN
BEGIN  (*PROPOSICAO "IMPLICA"*)
IND1:=TABPROP[II].PROP[2];  (*ELEMENTO 1*)
IND2:=TABPROP[II].PROP[3];  (*ELEMENTO 2*)
(*VERIFICACAO DO C => FUNCAO F*)
IF (TABPROP[IND1].A[0]) AND (TABPROP[IND2].A[0])
          THEN TABPROP[II].A[0]:=TRUE
          ELSE
IF (TABPROP[IND1].A[0]) AND (TABPROP[IND2].A[1])
          THEN TABPROP[II].A[0]:=TRUE
          ELSE
IF (TABPROP[IND1].A[1]) AND (TABPROP[IND2].A[0])
          THEN TABPROP[II].A[0]:=TRUE
          ELSE
```

```
IF (TABPROP[IND1].A[1]) AND (TABPROP[IND2].A[1])
        THEN TABPROP[II].A[0]:=TRUE
        ELSE
IF (TABPROP[IND1].A[2]) AND (TABPROP[IND2].A[0])
        THEN TABPROP[II].A[0]:=TRUE
        ELSE
IF (TABPROP[IND1].A[2]) AND (TABPROP[IND2].A[1])
        THEN TABPROP[II].A[0]:=TRUE
        ELSE
IF (TABPROP[IND1].A[2]) AND (TABPROP[IND2].A[2])
        THEN TABPROP[II].A[0]:=TRUE;
IF IND1<>IND2 THEN
IF ((TABPROP[IND1].A[0]) AND (TABPROP[IND2].A[2])) OR
   ((TABPROP[IND1].A[1]) AND (TABPROP[IND2].A[2]))
        THEN BEGIN
             TABPROP[II].A[2]:=TRUE;
             TABPROP[II].TEO:=FALSE
             END
END;  (*PROP. "IMP" *)
IF TABPROP[II].PROP[1]=0 THEN
        BEGIN  (*PROPOSICAO "NAO" *)
        IND1:=TABPROP[II].PROP[2];
        (*VERIFICACAO DO C=>FUNCAO G*)
        IF TABPROP[IND1].A[0] THEN
                BEGIN
                TABPROP[II].A[2]:=TRUE;
                TABPROP[II].TEO:=FALSE
                END;
        IF (TABPROP[IND1].A[1]) OR (TABPROP[IND1].A[2])
                THEN TABPROP[II].A[0]:=TRUE
        END  (*PROP. "N" *)
END;  (*FOR II*)

(*IMPRESSAO DOS TEOREMAS*)
FOR D:=1 TO TOPO DO
        BEGIN
        IF TABPROP[D].TEO THEN BEGIN
                I:=1;
                IND:=0;
                DESCOD(D);
                FOR L:=1 TO I-1 DO WRITE(VETOR[L]);
                WRITELN
                END;
        TABPROP[D].TEO:=FALSE  (*PARA NAO MAIS IMPRIMIR*)
        END  (*FOR D*)
END;  (*FOR K*)
WRITELN('FIM')
END.

  0 ERROR(S) DETECTED

HIGHSEG:    1K +  134 WORD(S)
LOWSEG :   14K +  278 WORD(S)

RUNTIME:    0: 3.608
```

TEOREMAS :

```
X0=>X0
X1=>X1
X1=>(X0=>X0)
NX0=>NX0
NX0=>(X0=>X0)
(X0=>X0)=>(X0=>X0)
X0=>(X0=>X0)
X2=>X2
X2=>(X0=>(X0=>X0))
N(X0=>X0)=>X2
X2=>((X0=>X0)=>(X0=>X0))
X2=>(NX0=>(X0=>X0))
X2=>(NX0=>NX0)
X2=>(X1=>(X0=>X0))
X2=>(X1=>X1)
X2=>(X0=>X0)
(X0=>(X0=>X0))=>(X0=>(X0=>X0))
((X0=>X0)=>X0)=>(X0=>(X0=>X0))
N(X0=>X0)=>(X0=>(X0=>X0))
(X0=>(X0=>X0))=>((X0=>X0)=>(X0=>X0))
((X0=>X0)=>(X0=>X0))=>(X0=>(X0=>X0))
(X0=>NX0)=>(X0=>(X0=>X0))
(NX0=>X0)=>(X0=>(X0=>X0))
((X0=>X0)=>NX0)=>(X0=>(X0=>X0))
(X0=>(X0=>X0))=>(NX0=>(X0=>X0))
(NX0=>(X0=>X0))=>(X0=>(X0=>X0))
NNX0=>(X0=>(X0=>X0))
(X0=>(X0=>X0))=>(NX0=>NX0)
(NX0=>NX0)=>(X0=>(X0=>X0))
(X0=>X1)=>(X0=>(X0=>X0))
(X1=>X0)=>(X0=>(X0=>X0))
((X0=>X0)=>X1)=>(X0=>(X0=>X0))
(X0=>(X0=>X0))=>(X1=>(X0=>X0))
(X1=>(X0=>X0))=>(X0=>(X0=>X0))
(NX0=>X1)=>(X0=>(X0=>X0))
(X1=>NX0)=>(X0=>(X0=>X0))
NX1=>(X0=>(X0=>X0))
(X0=>(X0=>X0))=>(X1=>X1)
(X1=>X1)=>(X0=>(X0=>X0))
X1=>(X0=>(X0=>X0))
NX0=>(X0=>(X0=>X0))
(X0=>(X0=>X0))=>(X0=>X0)
(X0=>X0)=>(X0=>(X0=>X0))
X0=>(X0=>(X0=>X0))
((X0=>X0)=>X0)=>((X0=>X0)=>X0)
N(X0=>X0)=>((X0=>X0)=>X0)
((X0=>X0)=>X0)=>((X0=>X0)=>(X0=>X0))
((X0=>X0)=>X0)=>(NX0=>(X0=>X0))
((X0=>X0)=>X0)=>(NX0=>NX0)
((X0=>X0)=>X0)=>(X1=>(X0=>X0))
((X0=>X0)=>X0)=>(X1=>X1)
((X0=>X0)=>X0)=>(X0=>X0)
N(X0=>X0)=>N(X0=>X0)
NN(X0=>X0)
N(X0=>X0)=>((X0=>X0)=>(X0=>X0))
N(X0=>X0)=>(X0=>NX0)
N(X0=>X0)=>(NX0=>X0)
N(X0=>X0)=>((X0=>X0)=>NX))
```

```
N(X0=>X0)=>(NX0=>(X0=>X0))
N(X0=>X0)=>NNX0
N(X0=>X0)=>(NX0=>NX0)
N(X0=>X0)=>(X0=>X1)
N(X0=>X0)=>(X1=>X0)
N(X0=>X0)=>((X0=>X0)=>X1)
N(X0=>X0)=>(X1=>(X0=>X0))
N(X0=>X0)=>(NX0=>X1)
N(X0=>X0)=>(X1=>NX0)
N(X0=>X0)=>NX1
N(X0=>X0)=>(X1=>X1)
N(X0=>X0)=>X1
N(X0=>X0)=>NX0
N(X0=>X0)=>(X0=>X0)
N(X0=>X0)=>X0
((X0=>X0)=>(X0=>X0))=>((X0=>X0)=>(X0=>X0))
(X0=>NX0)=>((X0=>X0)=>(X0=>X0))
(NX0=>X0)=>((X0=>X0)=>(X0=>X0))
((X0=>X0)=>NX0)=>((X0=>X0)=>(X0=>X0))
((X0=>X0)=>(X0=>X0))=>(NX0=>(X0=>X0))
(NX0=>(X0=>X0))=>((X0=>X0)=>(X0=>X0))
NNX0=>((X0=>X0)=>(X0=>X0))
((X0=>X0)=>(X0=>X0))=>(NX0=>NX0)
(NX0=>NX0)=>((X0=>X0)=>(X0=>X0))
(X0=>X1)=>((X0=>X0)=>(X0=>X0))
(X1=>X0)=>((X0=>X0)=>(X0=>X0))
((X0=>X0)=>X1)=>((X0=>X0)=>(X0=>X0))
((X0=>X0)=>(X0=>X0))=>(X1=>(X0=>X0))
(X1=>(X0=>X0))=>((X0=>X0)=>(X0=>X0))
(NX0=>X1)=>((X0=>X0)=>(X0=>X0))
(X1=>NX0)=>((X0=>X0)=>(X0=>X0))
NX1=>((X0=>X0)=>(X0=>X0))
((X0=>X0)=>(X0=>X0))=>(X1=>X1)
(X1=>X1)=>((X0=>X0)=>(X0=>X0))
X1=>((X0=>X0)=>(X0=>X0))
NX0=>((X0=>X0)=>(X0=>X0))
((X0=>X0)=>(X0=>X0))=>(X0=>X0)
(X0=>X0)=>((X0=>X0)=>(X0=>X0))
X0=>((X0=>X0)=>(X0=>X0))
(X0=>NX0)=>(X0=>NX0)
(X0=>NX0)=>(NX0=>(X0=>X0))
(X0=>NX0)=>(NX0=>NX0)
(X0=>NX0)=>(X1=>(X0=>X0))
(X0=>NX0)=>(X1=>X1)
(X0=>NX0)=>(X0=>X0)
(NX0=>X0)=>(NX0=>X0)
(NX0=>X0)=>(NX0=>(X0=>X0))
(NX0=>X0)=>(NX0=>NX0)
(NX0=>X0)=>(X1=>(X0=>X0))
(NX0=>X0)=>(X1=>X1)
(NX0=>X0)=>(X0=>X0)
((X0=>X0)=>NX0)=>((X0=>X0)=>NX0)
((X0=>X0)=>NX0)=>(NX0=>(X0=>X0))
((X0=>X0)=>NX0)=>(NX0=>NX0)
((X0=>X0)=>NX0)=>(X1=>(X0=>X0))
((X0=>X0)=>NX0)=>(X1=>X1)
((X0=>X0)=>NX0)=>(X0=>X0)
(NX0=>(X0=>X0))=>(NX0=>(X0=>X0))
NNX0=>(NX0=>(X0=>X0))
(NX0=>(X0=>X0))=>(NX0=>NX0)
```

```
(NX0=>NX0)=>(NX0=>(X0=>X0))
(X0=>X1)=>(NX0=>(X0=>X0))
(X1=>X0)=>(NX0=>(X0=>X0))
((X0=>X0)=>X1)=>(NX0=>(X0=>X0))
(NX0=>(X0=>X0))=>(X1=>(X0=>X0))
(X1=>(X0=>X0))=>(NX0=>(X0=>X0))
(NX0=>X1)=>(NX0=>(X0=>X0))
(X1=>NX0)=>(NX0=>(X0=>X0))
NX1=>(NX0=>(X0=>X0))
(NX0=>(X0=>X0))=>(X1=>X1)
(X1=>X1)=>(NX0=>(X0=>X0))
X1=>(NX0=>(X0=>X0))
NX0=>(NX0=>(X0=>X0))
(NX0=>(X0=>X0))=>(X0=>X0)
(X0=>X0)=>(NX0=>(X0=>X0))
X0=>(NX0=>(X0=>X0))
NNX0=>NNX0
NNX0=>(NX0=>NX0)
NNX0=>(X1=>(X0=>X0))
NNX0=>(X1=>X1)
NNX0=>(X0=>X0)
(NX0=>NX0)=>(NX0=>NX0)
(X0=>X1)=>(NX0=>NX0)
(X1=>X0)=>(NX0=>NX0)
((X0=>X0)=>X1)=>(NX0=>NX0)
(NX0=>NX0)=>(X1=>(X0=>X0))
(X1=>(X0=>X0))=>(NX0=>NX0)
(NX0=>X1)=>(NX0=>NX0)
(X1=>NX0)=>(NX0=>NX0)
NX1=>(NX0=>NX0)
(NX0=>NX0)=>(X1=>X1)
(X1=>X1)=>(NX0=>NX0)
X1=>(NX0=>NX0)
NX0=>(NX0=>NX0)
(NX0=>NX0)=>(X0=>X0)
(X0=>X0)=>(NX0=>NX0)
X0=>(NX0=>NX0)
(X0=>X1)=>(X0=>X1)
(X0=>X1)=>(X1=>(X0=>X0))
(X0=>X1)=>(X1=>X1)
(X0=>X1)=>(X0=>X0)
(X1=>X0)=>(X1=>X0)
(X1=>X0)=>(X1=>(X0=>X0))
(X1=>X0)=>(X1=>X1)
(X1=>X0)=>(X0=>X0)
((X0=>X0)=>X1)=>((X0=>X0)=>X1)
((X0=>X0)=>X1)=>(X1=>(X0=>X0))
((X0=>X0)=>X1)=>(X1=>X1)
((X0=>X0)=>X1)=>(X0=>X0)
(X1=>(X0=>X0))=>(X1=>(X0=>X0))
(NX0=>X1)=>(X1=>(X0=>X0))
(X1=>NX0)=>(X1=>(X0=>X0))
NX1=>(X1=>(X0=>X0))
(X1=>(X0=>X0))=>(X1=>X1)
(X1=>X1)=>(X1=>(X0=>X0))
X1=>(X1=>(X0=>X0))
NX0=>(X1=>(X0=>X0))
(X1=>(X0=>X0))=>(X0=>X0)
(X0=>X0)=>(X1=>(X0=>X0))
X0=>(X1=>(X0=>X0))
```

```
(NX0=>X1)=>(NX0=>X1)
(NX0=>X1)=>(X1=>X1)
(NX0=>X1)=>(X0=>X0)
(X1=>NX0)=>(X1=>NX0)
(X1=>NX0)=>(X1=>X1)
(X1=>NX0)=>(X0=>X0)
NX1=>NX1
NX1=>(X1=>X1)
NX1=>(X0=>X0)
(X1=>X1)=>(X1=>X1)
X1=>(X1=>X1)
NX0=>(X1=>X1)
(X1=>X1)=>(X0=>X0)
(X0=>X0)=>(X1=>X1)
X0=>(X1=>X1)
FIM
```

PROBLEMAS EM ABERTO.

Estudamos algumas propriedades bastante gerais das lógicas n-valentes de primeira ordem, embora, como vimos, dois aspectos tenham limitado essa generalidade: o primeiro, a respeito dos quantificadores de distribuição e o segundo, a respeito da existência de pelo menos um quantificador existencial e de um quantificador universal. Mesmo com estas hipóteses, não nos parece haver muita possibilidade de sucesso na tentativa de estudar a Teoria de Modelos para lógicas de primeira ordem em geral, desde que muitos resultados clássicos da Teoria de Modelos estão estreitamente ligados a conectivos particulares (como, por exemplo, o teorema da consistência conjunta).

Colocando condições nos conectivos e quantificadores, contudo, é possível obter algumas versões polivalentes de teoremas clássicos da Teoria de Modelos (ver, por exemplo, [7], [15], [26]).

Dentro desta perspectiva, poderíamos apresentar alguns problemas em aberto ligados aos resultados que obtivemos:

1) É possível obter uma versão polivalente geral ainda que restrita ao caso proposicional, do teorema da interpolação de Craig? Em termos formais, se X e Y são fórmulas de uma lógica proposicional polivalente L e $X \vdash_L Y$, em que condições existe uma fórmula Z, cujas componentes atômicas encontram-se entre as de X e de Y, tal que $X \vdash_L Z$ e $Z \vdash_L Y$?

2) Para quais quantificadores de distribuição se podem obter teoremas "prenex" (isto é, teoremas que garantem equivalência, em algum sentido, entre uma sentença e alguma forma "prenexada", obtida colocando-se todos os quantificadores juntos e no início da sentença) ?

3) Se permitíssemos uma versão infinitária de uma lógica n--valente de primeira ordem (isto é, se permitíssemos fórmulas de comprimento enumerável), seria possível obter uma versão correspondente do Teorema de Lindstrom? De acordo com as discussões que mantivemos com o Professor Xavier Caicedo, parece possível demonstrar, para algumas lógicas n-valentes funcionalmente completas, certas generalizações do Teorema de Lindstrom.

4) Sob que condições poderiam ser introduzidos os chamados "operadores que formam termos ligando variáveis de fórmulas" em lógica polivalente, de modo que os principais teoremas clássicos referentes a esses operadores permanecessem válidos? Em particular, seria interessante se dispuséssemos de uma teoria polivalente dos símbolos de Hilbert, ε e τ, e do operador de descrição (cf.[4]).

5) Nosso tratamento dos quantificadores em lógica polivalente pode ser visto como um caso particular dos quantificadores de Rescher-Mostowski (ver [15]) como esclarecemos na Introdução e no Capítulo IV; que parte, então, da exposição precedente pode se estender aos quantificadores de Rescher-Mostowski?

6) Como verificamos, em conjunto com o Professor Newton C. A. da Costa, boa porção de nossa exposição se estende à lógica polivalente de ordem superior, caso utilizemos estruturas de ordem superior não-standard, reformulando-se em termos polivalentes a noção de estrutura de Henkin (bivalente). A questão que se coloca é: que parte dos

resultados relativos aos tableaux quantificacionais se amplia para a lógica polivalente de ordem superior?

7) Outro problema digno de menção é o seguinte: englobar, num tratamento unitário, os sistemas polivalentes tratados neste trabalho e os sistemas chamados "quasi-thuth-functionals" (ver [15], pp. 166 e seguintes); que parte de nossa exposição poderia ser vista dessa maneira?

8) Por último, na Seção IV.3, exemplo 3, onde utilizamos como exemplo de tableaux quantificacionais os tableaux para o cálculo J_3, não é difícil perceber que as regras J-3 e J-4 poderiam, do ponto de vista intuitivo, ser substituidas por, respectivamente,

$$J\text{-}3'\colon \frac{a_2((\exists x)A)}{a_2(Ac)} \qquad \text{e} \qquad J\text{-}4'\colon \frac{a_o((\forall x)A)}{a_o(Ac)}$$

onde c não ocorreu anteriormente no tableau; o problema que se coloca é como definir tableaux quantificacionais para que uma tal simplificação nas regras seja possível, e que para o caso clássico, essa simplificação leve à formulações dos tableaux em [24].

BIBLIOGRAFIA

[1] ARRUDA, A.I., *Aspects of the historical development of paraconsistent logic*, a aparecer em "Paraconsistent Logic" (eds. R. Routley e G. Priest).

[2] BETH, F.W., THE FOUNDATIONS OF MATHEMATICS, North-Holland , Amsterdam, (1959).

[3] CHANG, C.C. e KEISLER, H.J.; MODEL THEORY, North-Holland , Amsterdam, (1973).

[4] da COSTA, N.C.A., A *model-theoretical approach to variable binding term operators*, in "Mathematical Logic in Latin America" (Eds.: (A.I.Arruda, R.Chuaqui, N.C.A.da Costa), North-Holland (1980), pp. 133-162.

[5] da COSTA, N.C.A. e ALCÂNTARA, L.P., *On paraconsistent set theories*, Relatório Interno nº 215, IMECC-UNICAMP, Campinas, (1982).

[6] da COSTA, N.C.A. e ALVES, E.H., *Relations between paraconsistent logic and many-valued logic*, Trabalhos do Departamento de Matemática, IME-USP, São Paulo (1981)

[7] D'OTTAVIANO, I.M.L, *Sobre uma teoria de modelos trivalente* , Tese de Doutoramento, IMECC, Universidade Estadual de Campinas, (1982).

[8] DRAKE, F.R., SET THEORY, North-Holland, Amsterdam, (1974)

[9] GOODSTEIN, R.L., *Truth Tables*, The Mathematical Gazette,vol. XLVI, nº 355, (1962), pp. 18-23.

[10] KALICKI, J., *A test for the existence of tautologies according to many-valued truth-tables*, The Journal of Symbolic Logic, vol. 15, nº 3, (1950), pp. 182-184.

[11] KLEENE, S.C., MATHEMATICAL LOGIC, John Wiley & Sons, Inc. (1967)

[12] MARTIN, N.M., *Some analogues of the Sheffer stroke functions in n-valued logics*, Indagationes Mathematicae, vol.12(1950), pp. 373-400.

[13] MARTIN, N.M., *The Sheffer functions of 3-valued logic*, The Journal of Symbolic Logic, vol. 19, nº 1 (1954), pp. 45-51.

[14] RASIOWA, H. e SIKORSKI, R., THE MATHEMATICS OF METAMATHEMATICS, Polish Scientific Publishers, Varsóvia, (1970).

[15] RESCHER, N., MANY-VALUED LOGIC, Mc Graw-Hill, (1969).

[16] RINE, D. (Editor), COMPUTER SCIENCE AND MULTIPLE-VALUED LOGIC, North-Holland, Amsterdam, (1977).

[17] ROSE, A., *Many-valued logical machines*, Proceedings of the Cambridge Phil. Soc. vol. 54, Part 3 (1958), pp. 307-321.

[18] ROSENBERG, I., *The number of maximal closed classes in the set of functions over a finite domain*, J. of Comb.Theory (A) 14, (1973), pp. 1-7.

[19] ROSENBLOOM, P.C., THE ELEMENTS OF MATHEMATICAL LOGIC, Dover Publications, Inc., New York (1951).

[20] ROSSER, J.B. e TURQUETTE, A.R., MANY-VALUED LOGICS, North-Holland, Amsterdam (1952).

[21] SALOMAA, A., *A theorem concerning the composition of functions of several variables ranging over a finite set*, The Journal of Symbolic Logic, vol. 25, nº 3 (1960), pp.203-208.

[22] SETTE, A.M., *On the propositional calculus* P^1, Mathematica Japonicae 16, (1973), pp. 173-180.

[23] SMULLYAN, R.M., *A unifying principle in quantification theory*, Proceedings of the National Academy of Sciences (1963).

[24] SMULLYAN, R.M., FIRST-ORDER LOGIC, Springer-Verlag, Berlin, (1968).

[25] SURMA, S.J., *An algorithm for axiomatizing every finite logic*, in "Computer science and multiple-valued logic" (ed. D. Rine), North-Holland Publishing Company (1977), pp. 137-143.

[26] WEAVER, G., *Compactness theorems for finitely - many - valued sentencial logics*, Studia Logica XXXVII (1978), pp.413-416.

[27] WOLF, R.G., *A survey of many-valued logic* (1966-1974), Notas mimeografadas, Southern Illinois University, Edwardsville.