

TABLEAUX ANALÍTICOS DE PRIMEIRA ORDEM: UTILIZAÇÃO COMO PROCEDIMENTO DE DECISÃO PARA DEPENDÊNCIAS DE DADOS

Claudio M.O. Moura
IBM Instituto Latino Americano de Pesquisa em Sistemas
IBM do Brasil Ltda.
Rio de Janeiro

Marco Antonio Casanova
IBM Centro Científico
IBM do Brasil Ltda.
Brasília

Palavras-chaves ou frases: Banco de Dados, projeto de bancos de dados, modelo relacional, procedimentos de decisão, dependências de dados, tableaux, tableaux analíticos de primeira ordem, inferência lógica.

Resumo:

Uma linguagem para definição de restrições de integridade que apresenta uma notação uniforme para as dependências de dados mais comumente encontradas na literatura é apresentada.

O problema da decisão para dependências de dados, ou seja, dado um conjunto de dependência de dados Σ determinar se Σ implica logicamente uma nova dependência de dados, σ ou seja, determinar se $\Sigma \models \sigma$, é analisado através da utilização do método dos tableaux analíticos, adaptado da lógica de primeira ordem. A utilização deste método como procedimento de decisão para certas classes de dependências é proposto e uma comparação com o método dos tableaux proposto por Aho, Sagiv e Ullman, utilizados também como procedimento de decisão é apresentada.

O problema da indecidibilidade para algumas classes de dependências de dados é analisado e a indecidibilidade das classes de dependências de inclusão em presença de dependências funcionais contornada através de um esquema relacional restrito. Esta particularização apresenta interesse prático devido à importância daquelas classes de dependências no projeto de bancos de dados relacionais.

1. Introdução

Um Banco de Dados Relacional tem por objetivo modelar a realidade de uma parte da operação de uma empresa. Podemos citar como exemplo um banco de dados sobre produtos de uma companhia, que visa mostrar como suas vendas estão se comportando, seus estoques sendo administrados, suas linhas de fabricação reagindo, sua linha de produtos satisfazendo as necessidades de seu mercado e de seus clientes etc.

Um segundo exemplo seria um banco de dados estatístico, com dados demográficos, que permitisse a análise de fatos da realidade para previsão ou planejamento de medidas econômicas, sociais, etc que visem melhorar as condições de vida de uma comunidade, de um grupo qualquer ou mesmo de uma nação como um todo.

A necessidade de representar ou modelar o mundo real tão fielmente quanto possível impõe ao projeto de um banco de dados uma série de problemas. Sem dúvida, o entendimento ou captação do que é aquela realidade que se vai modelar é a fase mais difícil e complexa com que se defronta o projetista. Através da percepção do projetista do banco de dados em relação ao objeto que está sendo modelado, um conjunto de classes de dados e restrições sobre valores ou comportamento destes dados são então definidos. Estes resultados formam um conjunto de descrições ou esquemas que definem os valores ou estados do banco de dados, bem como as mudanças ou transições válidas para um estado.

A proposta do Grupo de Estudos sobre Sistemas de Gerência de Bancos de Dados ANSI/SPARC [ANSI] chama àquela descrição do banco de dados de esquema conceitual. O esquema conceitual descreve o banco de dados como um todo ou seja como a reunião das visões dos vários usuários.

A complexidade decorrente da grande quantidade de detalhes, associada ao

grande número de interrelações existentes entre os dados, faz com que a tarefa de definir um esquema conceitual seja extremamente difícil. Acresce-se ainda o grau de subjetividade inerente ao processo de comunicação entre o projetista e os usuários gerada da necessidade do projetista entender, tão próximo quanto possível, a realidade como vista pelos usuários.

Assim, a qualidade de um banco de dados, como ferramenta útil para ajudar a tomada de decisão em uma empresa, está diretamente ligada à proximidade do modelo à semântica pretendida da operação da empresa. A utilização de sistemas formais para especificação de esquemas visa tornar a descrição de um banco de dados mais precisa e mais clara. Sua utilização vem se difundindo com o auxílio principalmente de alguns ramos da matemática.

Este trabalho mostrará como podemos utilizar alguns métodos práticos e teóricos para Procedimentos de Decisão para Dependências de Dados, como explicado a seguir.

Começemos pelo conceito de Dependência de Dados. Na teoria de Bancos de Dados, determinadas classes de sentenças acerca de relações têm interesse prático e teórico para o projeto de bancos de dados relacionais. A primeira destas classes de sentenças a ser estudada, introduzida por Codd [Co], foi chamada dependência funcional FD. Por este motivo aquelas classes de sentenças receberam, historicamente, o nome de dependências. Outras classe de dependência importante é a chamada dependência de inclusão ou IND [Fa³]. Estas e outras classes de dependências que serão mencionadas neste trabalho são de interesse particular para a normalização de bancos de dados, bem como para determinar que dados devem ser duplicados nas relações que constituem o banco de dados.

Passemos agora a explicar o que é um procedimento de decisão. O problema da decisão para Σ , sendo Σ um conjunto de sentenças, consiste em determinar um algoritmo ou procedimento efetivo através do qual, para cada sentença ϕ , podemos determinar se $\Sigma \models \phi$ ou não. Se tal procedimento efetivo existe o problema da decisão para Σ é solúvel e Σ é dito decidível.

Por procedimento efetivo entendemos [En]:

- Existe um número finito de instruções precisas explicando como executar o procedimento. Estas instruções não devem requerer nenhuma engenhosidade, conhecimento ou inteligência da pessoa ou máquina que as estão executando.
- O procedimento não é probabilístico.
- No caso de um problema de decisão, o procedimento deve, dado uma sentença ϕ , "responder" sim ou não após um número finito de passos se $\Sigma \models \phi$ ou $\Sigma \not\models \phi$.

Um algoritmo ou procedimento efetivo para solução de um problema de decisão é chamado de procedimento de decisão.

A necessidade de procedimentos de decisão para dependências de dados aparece em várias etapas da especificação do esquema conceitual. Como exemplos podemos citar a determinação se um esquema de relação está normalizado, o teste de equivalência entre esquemas relacionais, a integração de visões, etc.

Procedimentos de decisão para dependências de dados vem sendo estudados por vários autores. Fagin [Fa²] introduziu o conceito de "faithfulness" que permite provar resultados poderosos sobre a existência de "relações de Armstrong". Uma relação de Armstrong é uma relação que obedece rigorosamente as dependências de dados que são consequências lógicas de um dado conjunto de dependências.

Casanova, Fagin e Papadimitrou [ICFP] estudaram as dependências de inclusão e suas interações com as dependências funcionais e mostraram que o problema da decisão para dependências funcionais "PSPACE-complete".

Honeyman em [Ho] apresenta algumas definições de "satisfação" para determinar se um conjunto de relações satisfazem a um conjunto de dependências funcionais e apresenta um algoritmo, em tempo polinomial, para aquele teste de satisfação. Maier, Mendelzon e Sagiv [MMS] apresentam um método para teste de implicação de dependências funcionais a partir de um conjunto de dependências de dados.

Klug e Price [KP] apresentam uma técnica para determinar se restrições sobre visões são consequências lógicas das restrições da relação base.

Casanova [Ca] descreve uma teoria formal para análise de dependências implicacionais definidas sobre expressões relacionais e discute como o método dos tableaux analíticos pode, para algumas classes, se tornar um procedimento de decisão. Isto conclui a nossa discussão geral sobre dependências e procedimentos de decisão.

Este trabalho está estruturado em quatro capítulos.

O capítulo 1 apresenta os objetivos, motivação e trabalhos correlatos.

O capítulo 2 introduz algumas definições preliminares sobre Bancos de Dados. O capítulo 3 apresenta o método dos tableaux analíticos de primeira ordem, da lógica matemática, e sua utilização como procedimento de decisão para dependências de dados. Uma comparação com o método dos tableaux proposto por [ASU, MMS] é também apresentada. Por último, no capítulo 4, conclusões e direções para trabalhos futuros.

2. Dependências de Dados

Antes de tratarmos das dependências de dados apresentamos a definição de linguagens de primeira ordem e mostraremos, após algumas definições preliminares sobre Bancos de Dados, como aquelas dependências de dados comumente encontradas na literatura são expressas nestas linguagens.

2.1 - Linguagens de Primeira Ordem

Uma linguagem de primeira ordem L é uma linguagem formal na qual proposições sobre uma determinada estrutura podem ser expressas. Podemos descrever uma linguagem arbitrária de primeira ordem L através de seus símbolos e de leis de associação dos mesmos.

Os símbolos de L são:

- Uma sequência infinita de variáveis individuais v_1, v_2, \dots
- Para cada número natural n , um conjunto de símbolos funcionais n -ários;
- Para cada número natural positivo n , um conjunto de símbolos predicativos n -ários (Para pelo menos um n este conjunto tem que ser não vazio);
- Os conectores \neg (negação) e \rightarrow (implantação);
- O quantificador universal \forall ;

Os símbolos funcionais 0 -ários, se existirem, são chamados de constantes.

Dizemos que L é uma linguagem com igualdade se ela tem o símbolo predicativo binário especial $=$.

As variáveis, conectivos, quantificadores e o predicativo especial $=$ são chamados símbolos lógicos. Os símbolos funcionais e os símbolos predicativos, com exceção de $=$, são chamados símbolos extralógicos.

Uma L -cadeia é uma sequência finita de L -símbolos, possivelmente vazia.

Um L -termo é uma L -cadeia formada de acordo com as seguintes regras:

- Qualquer L -cadeia formada por uma ocorrência simples de uma variável é um L -termo;
- Se f é um símbolo funcional n -ário de L e t_1, \dots, t_n são L -termos, ft_1, \dots, t_n é um L -termo e t_1, \dots, t_n são os argumentos de f .

Uma L -fórmula é uma L -cadeia formada de acordo com as seguintes regras:

- Se P é símbolo predicativo n -ário de L e t_1, \dots, t_n são L -termos

Pt_1, \dots, t_n é uma L -fórmula

- Se α é uma L -fórmula

$\neg \alpha$ é uma L -fórmula

- Se α e β são L -fórmulas

$\alpha \rightarrow \beta$ é uma L -fórmula

- Se α é uma L -fórmula e x uma variável

$\forall x \alpha$ é uma fórmula L -fórmula

Uma fórmula do tipo Pt_1, \dots, t_n ou $x=y$ é chamada uma fórmula atômica. Uma fórmula atômica da forma $x=y$ é chamada de uma igualdade.

Uma L -estrutura é uma tripla ordenada $A = \langle U, R, C \rangle$ onde

- U é um conjunto não vazio, chamado universo de discurso, universo ou domínio de A . Os elementos de U são chamados de indivíduos de A .
- R é uma função que atribui a cada símbolo funcional f de L uma operação f^A sobre U tal que se f é um símbolo funcional n -ário, f^A é uma operação n -ária sobre U . Se α é uma constante então α^A é um indivíduo de A .
- C é uma função que atribui a cada símbolo predicativo estralógico P de L uma relação P^A sobre U tal que se P é um símbolo predicativo n -ário P^A é uma relação n -ária em U .

Uma L -valorização σ é uma L -estrutura A associada a uma atribuição de um valor $x^\sigma \in U$ para cada variável x de L . Por definição cada valorização σ está relacionada a uma estrutura particular A ou seja A é a estrutura subjacente de σ .

Se A é a estrutura subjacente de σ definimos f^σ e P^σ como a operação f^A e a relação P^A , respectivamente, onde f é qualquer símbolo funcional e P é qualquer símbolo predicativo extralógico de L . O universo U de A é o universo de σ .

sob a valorização σ , denotado por $t^\sigma \in U$, da seguinte forma:

- Se x é uma variável x^σ está definida.
- Se f é um símbolo funcional n -ário de L e t_1, \dots, t_n são L -termos,

$$(ft_1, \dots, t_n)^\sigma = f(t_1^\sigma, \dots, t_n^\sigma)$$

Dada uma L -valorização σ em U , definimos para cada L -fórmula α o valor de α sob a valorização σ , denotada por α^σ , tal que α^σ ou é verdadeira T ou falsa \perp , da seguinte forma:

- Se P é um símbolo predicativo extralógico n -ário de L e t_1, \dots, t_n são L -termos

$$(Pt_1, \dots, t_n)^\sigma = \begin{cases} T & \text{se } \langle t_1^\sigma, \dots, t_n^\sigma \rangle \in P^\sigma \\ \perp & \text{caso contrário} \end{cases}$$

- Se \neg e \rightarrow são L -termos e L é uma linguagem com igualdade

$$(s \rightarrow t)^\sigma = \begin{cases} T & \text{se } s^\sigma = \perp \text{ ou } t^\sigma = T \\ \perp & \text{caso contrário} \end{cases}$$

- Para cada L -fórmula β

$$(\neg \beta)^\sigma = \begin{cases} T & \text{se } \beta^\sigma = \perp \\ \perp & \text{caso contrário} \end{cases}$$

- Para cada L -fórmulas β e γ ,

$$(\beta \rightarrow \gamma)^\sigma = \begin{cases} T & \text{se } \beta^\sigma = \perp \text{ ou } \gamma^\sigma = T \\ \perp & \text{caso contrário} \end{cases}$$

- Para cada L -fórmula β e variável x ,

$$(\forall x \beta)^\sigma = \begin{cases} T & \text{se } \beta^\sigma(x/x) = T \text{ para todo } u \in U \\ \perp & \text{caso contrário} \end{cases}$$

onde U é o universo de σ

Uma ocorrência de x em uma fórmula α é dita ligada se esta ocorrência se dá dentro de uma subfórmula de α da forma $\forall x \beta$. Todas as demais ocorrências de x são chamadas livres.

Dizemos que x é livre em α se x tem pelo menos uma ocorrência livre em α . Uma sentença é uma fórmula sem variáveis livres.

Dizemos que a estrutura A é modelo para uma sentença α , $A \models \alpha$, se qualquer valorização de α em A for verdadeira, $\alpha^A = T$. Se num conjunto de sentenças \mathcal{A} , cada sentença ϕ é satisfeita em A dizemos que A é modelo para \mathcal{A} ou abreviadamente $A \models \mathcal{A}$. Uma teoria em L é um par $\mathcal{T} = \langle L, \Sigma \rangle$, onde L é uma linguagem de primeira ordem e Σ um conjunto de sentença de L , chamadas de axiomas de \mathcal{T} .

2.2 - Definições Preliminares sobre Bancos de Dados

Para definirmos alguns conceitos relacionados a Bancos de Dados introduziremos, neste ponto, uma linguagem de primeira ordem na qual as variáveis são de diferentes sortes ou tipos, percorrendo universos diferentes. Uma linguagem com tal característica chamamos Linguagem Polisortida (En).

Uma linguagem relacional L é uma linguagem polisortida de primeira ordem, sem símbolos funcionais e com sortes

$$A = \{A_1, \dots, A_p\}$$

Um esquema de relação é uma expressão $R(A_1, \dots, A_n)$ onde R é um símbolo predicativo n -ário e $\{A_1, \dots, A_n\}$ é o sorte de R . A_1, \dots, A_n são os atributos de R . Em certas partes deste trabalho identificaremos L com o conjunto de todos os seus esquemas de relação. Em algumas ocasiões trataremos L como uma linguagem sem sortes, isto é, ignoraremos as restrições sobre a formação de

(L,C) define uma teoria.

Um estado do banco de dados é uma estrutura para L. Um estado válido do banco de dados é um modelo para C.

2.3 - Classes de Dependências de Dados

Determinadas classes de sentenças relacionais apresentam interesse, tanto prático como teórico, para o estudo de Bancos de Dados. (Col, Fal, Fa2, ABU) Por razões históricas estas classes de sentenças são chamadas de dependências. A primeira classe destas sentenças, propostas e estudadas por Codd e outros foram as dependências funcionais ou FDs.

É facilmente demonstrado (NI) que dependências funcionais podem ser expressas por sentenças usando-se uma linguagem da lógica de primeira ordem. Esta observação foi estendida por Fagin (Fa3) para todas as classes de dependências de dados geralmente encontradas na literatura, através das dependências implicacionais, IDs (implicational dependencies), dependências implicacionais embutidas, EIDs (imbedded implicational dependencies) e da generalização destas, as dependências implicacionais embutidas e estendidas, XEIDs, (extended imbedded implicational dependencies).

Em seguida mostraremos como isto pode ser feito para cada uma destas classes de dependência.

Seja L uma linguagem relacional no que se segue.

Uma dependência implicacional (ID) é uma sentença tipificada da forma:

$$\forall x_1 \dots \forall x_n ((A_1 \wedge \dots \wedge A_n) \rightarrow B)$$

onde A_i são fórmulas relacionais e B é uma fórmula atômica e onde cada variável x_j que aparece em pelo menos um A_i ou em B é A sentença

$$\forall x \forall y (R_1 x y \rightarrow \exists z \exists y' R_2 x y')$$

é chamada de uma dependência de inclusão e abreviada como $R_2[A] \subseteq R_1[A]$. □

A classe de dependências acima é sem dúvida uma das mais expressivas classes de dependências não tipificadas. Através das dependências de inclusão podemos expressar o fato de que a projeção de uma relação por um atributo é um subconjunto próprio ou não da projeção de outra ou da mesma relação por outro atributo.

Por exemplo, se se quer especificar que todos os engenheiros da fábrica de uma empresa são também funcionários daquela empresa, podemos assumir que as relações

ENGENHEIRO (NOME, ESPECIFICAÇÃO) e
FUNCIONÁRIO (NOME, DEPARTAMENTO, SALÁRIO)

representam os engenheiros da fábrica e todos os funcionários da empresa e escrevemos:

$$\forall n \forall e (\text{ENGENHEIRO } n e \rightarrow \exists d \exists s \text{ FUNCIONÁRIO } n d s)$$

universalmente quantificada.

Podemos mostrar que FDs e MVDs são casos particulares de IDs conforme exemplificado abaixo.

Exemplo 2.1:

Suponha que R(ABCD) seja um esquema de relação de uma linguagem relacional L.

(a) A sentença

$$\forall a \forall b \forall c_1 \forall c_2 \forall d_1 \forall d_2 ((R a b c_1 d_1 \wedge R a b c_2 d_2) \Rightarrow (c_1 = c_2))$$

é chamada de uma dependência funcional e abreviada como $R:AB \rightarrow C$

(b) A sentença

$$\forall a \forall b_1 \forall b_2 \forall c_1 \forall c_2 ((R a b_1 c_1 \wedge R a b_2 c_2) \Rightarrow R a b_1 c_2)$$

é chamada de uma dependência multivalorada (Fal) e abreviada como $LR:A \twoheadrightarrow B$.

□

DEPENDÊNCIAS IMPLICACIONAIS EMBUTIDAS

Uma dependência implicacional embutida é uma sentença tipificada da forma

$$\forall x_1 \dots \forall x_m ((A_1 \wedge A_2 \wedge \dots \wedge A_n) \Rightarrow (\exists y_1 \dots \exists y_r) (B_1 \wedge \dots \wedge B_s))$$

onde cada A_i é uma fórmula relacional e cada B_j é atômica. x_j aparece em pelo menos um dos A_i , $n \geq 1$ e $s \geq 1$.

Exemplo 2.2:

Suponha que R(ABCD) seja um esquema de relação de uma linguagem relacional L.

A sentença:

$$\forall a \forall b_1 \forall b_2 \forall c_1 \forall c_2 \forall d_1 \forall d_2 ((R a b_1 c_1 d_1 \wedge R a b_2 c_2 d_2) \Rightarrow \exists d_3 R a b_1 c_2 d_3)$$

é chamada dependência multivalorada embutida - EMVD - e abreviada como

$$R:A \twoheadrightarrow B/C$$

A sentença acima exprime o fato que a MVD $A \twoheadrightarrow B$ é preservada na projeção de R em ABC.

DEPENDÊNCIAS IMPLICACIONAIS EMBUTIDAS E EXTENDIDAS

Uma dependência implicacional embutida e estendida é uma sentença da forma $\forall x_1 \dots \forall x_m ((A_1 \wedge \dots \wedge A_n) \Rightarrow (\exists y_1 \dots \exists y_r) (B_1 \wedge \dots \wedge B_s))$

onde cada A_i é uma fórmula relacional e cada B_j é atômica. Deve existir ao menos um A_i ou seja $n \geq 1$. Diferentemente das EIDs, que devem ser tipificadas, as XEIDs têm uma restrição mais fraca onde somente o lado esquerdo da implicação deve ser tipificado.

Exemplo 2.3:

Suponha que $R_1(AXY)$ e $R_2(AZ)$ são esquemas de relação de uma linguagem relacional L.

3. Método dos Tableaux Analíticos

Apresentamos neste capítulo os tableaux analíticos de primeira ordem e sua aplicação à teoria de Bancos de Dados como procedimento de decisão para dependências de dados. Na seção 3.1 apresentamos a teoria dos tableaux analíticos de primeira ordem; na seção 3.2 mostraremos o tableau analítico para a classe das dependências implicacionais embutidas e estendidas, estudadas no capítulo anterior e na seção 3.3. Analisaremos as vantagens e desvantagens dos tableaux apresentados por (MMS).

3.1 - Tableaux Analíticos de Primeira Ordem

Introduziremos os tableaux analíticos de primeira ordem através de algumas definições preliminares (BM, Sm).

Um tableau analítico é um conjunto de objetos chamados nós onde cada nó está associado a um conjunto finito de fórmulas. O conjunto de nós de um tableau é parcialmente ordenado pela relação de ancestralidade onde cada nó é ancestral de seus próprios sucessores, dos sucessores destes e assim por diante. Cada nó está associado a um único nível, identificado por um número natural. Existe um único nó de nível 0 chamado nó inicial do tableau analítico. Cada nó de nível $n+1$ é um sucessor de um único nó, de nível n . Um nó é terminal se ele não tem sucessores.

Dizemos que h é a altura do tableau analítico se o mesmo tem, pelo menos, um nó de nível h e nenhum nó de nível maior.

Um ramo de um tableau analítico é uma seqüência finita de nós ϕ_0, \dots, ϕ_k tal que ϕ_0 é o nó inicial do tableau analítico e, para $i=1, \dots, k$, ϕ_i é um sucessor de ϕ_{i-1} e ϕ_k é um nó terminal. Este ramo é chamado de ramo terminado em ϕ_k . Fórmulas pertencentes a um nó de um ramo são ditas fórmulas do ramo. As fórmulas do nó inicial são chamadas fórmulas iniciais do tableau analítico. Um tableau com nó inicial ϕ_0 é chamado de um tableau para ϕ_0 .

Definição 3.1.1 - Um tableau analítico de primeira ordem para um conjunto de fórmulas Φ é uma árvore cujo nó inicial é $\phi_0 = \Phi$ e os nós seguintes são sucessores de ϕ_0 obtidos pela aplicação das regras abaixo:

- 1 - Regra $\neg \neg$: Se entre as fórmulas de um ramo terminado em um nó ϕ_k existir uma fórmula da forma $\neg \neg \alpha$ acrescente um novo nó $\{\alpha\}$ como sucessor de ϕ_k .
- 2 - Regra \rightarrow : Se entre as fórmulas de um ramo terminando em ϕ_k existir uma fórmula da forma $\alpha \rightarrow \beta$ acrescente dois novos nós $\{\neg \alpha\}$ e $\{\beta\}$ como sucessores de ϕ_k .
- 3 - Regra $\neg \rightarrow$: Se entre as fórmulas de um ramo terminando em ϕ_k existir uma fórmula da forma $\neg (\alpha \rightarrow \beta)$, acrescente um novo nó $\{\alpha, \neg \beta\}$ como sucessor de ϕ_k .
- 4 - Regra \forall : Se entre as fórmulas de um ramo terminando em ϕ_k existir uma fórmula da forma $\forall x \alpha$ acrescente um novo nó $\{\alpha(x/t)\}$, onde t pode ser qualquer L-termo, como sucessor de ϕ_k .
- 5 - Regra \forall : Se entre as fórmulas de um ramo terminando em ϕ_k existir uma fórmula da forma $\forall x \alpha$ acrescente um novo nó $\{\alpha(x/y)\}$, onde y pode ser qualquer variável que não esteja livre em qualquer fórmula do ramo sendo expandido, como sucessor de ϕ_k .

Para linguagens com igualdade devemos ainda acrescentar as regras abaixo cujos nomes são abreviaturas de auto-identidade (self-identity), substitutividade de iguais em funções (substitutivity of equals in functions) e substitutividade de iguais em predicados (substitutivity of equals in predicates).

- 6 - Regra SI: Estenda qualquer ramo terminando em ϕ_k acrescentando um novo nó $\{t=t\}$, onde t é qualquer L-termo, como sucessor de ϕ_k .
- 7 - Regra SF: Se entre as fórmulas de um ramo terminando em ϕ_k existirem, em qualquer ordem, fórmulas da forma $t_1 = t_{n+1}, t_2 = t_{n+2}, \dots, t_n = t_{2n}$, acrescente um novo nó $\{f(t_1-t_n = t_{n+1}-t_{2n})\}$, onde f é qualquer função n -ária de L e t_1-t_{2n} são L-termos, como sucessor de ϕ_k .
- 8 - Regra SP: Se entre as fórmulas de um ramo terminando em ϕ_k existirem, em qualquer ordem, fórmulas da forma $t_1 = t_{n+1}, t_2 = t_{n+2}, \dots, t_n = t_{2n}, P(t_1-t_n)$, acrescente um novo nó $\{P(t_{n+1}-t_{2n})\}$.

onde P é um símbolo predicativo n-ário de L e t_1, \dots, t_n são L-termos, como sucessor de ϕ_k . □

A Figura 3.1.1 representa esquematicamente as regras do tableau analítico de primeira ordem. Até o fim deste capítulo chamaremos o tableau analítico de primeira ordem simplesmente de tableau analítico.

Observe que toda regra R descrita acima parte da existência de um conjunto de fórmulas em um ramo do tableau e acrescenta conjuntos de fórmulas β_1, \dots, β_n como sucessores do nó terminal do ramo.

Chamamos então α de conjunto de antecedentes (ou, simplesmente, antecedente) de R e β_1, \dots, β_n de conjuntos de conseqüentes (ou, simplesmente, conseqüentes) de R. Desta forma, podemos representar R como a tupla $\langle \alpha, \beta_1, \dots, \beta_n \rangle$.

Um ramo de um tableau analítico é fechado se existe uma fórmula tal que α e $\neg \alpha$ pertencem àquele ramo. Caso contrário, dizemos que o ramo é aberto. Diz-se que α e $\neg \alpha$ foram usadas para fechar o ramo. Um tableau analítico para ϕ é chamado uma confutação de primeira ordem para ϕ se todos os seus ramos são fechados.

A Figura 3.1.2 mostra regras adicionais para os conectivos \wedge, \vee e \leftrightarrow e para o quantificador existencial, não sendo necessário portanto transformá-las nos símbolos de nossa metalinguagem sempre que elas apareçam em uma fórmula.

Podemos tornar o método do tableau analítico mais forte pela introdução de uma nova regra:

9 - Regra EM: Estenda qualquer ramo terminando em ϕ_k acrescentando dois novos nós $\{\delta\}$ e $\{\neg \delta\}$ como sucessores de ϕ_k .

A regra acima, regra do terceiro excluído (excluded middle), é justificada semanticamente pela observação de que qualquer valorização satisfaz δ ou $\neg \delta$. Embora redundante a regra EM ajuda a conseguirmos confutações mais rapidamente.

Teorema 3.1.1 (BM) - Teorema da Eliminação. Dada uma confutação de ϕ na qual a regra EM é usada podemos construir uma confutação de ϕ sem a utilização da regra EM. □

Podemos não considerar, devido ao teorema 3.1.1, a regra EM como uma regra do tableau analítico mas usá-la na prática.

Definição 3.1.3 - Um conjunto Ψ de L-fórmulas é um Conjunto de Hintikka em L se satisfaz as seguintes condições

- 1) Se ψ é atômica, então ψ e $\neg \psi$ não pertencem a Ψ ;
- 2) Se existe uma regra $R_i = \langle \alpha_i, \beta_{i1}, \dots, \beta_{in_i} \rangle$ tal que o antecedente $\alpha_i \in \Psi$ e $\alpha_i \in \Psi$, então o conseqüente $\beta_{ij} \in \Psi$, para algum $j \in \{1, \dots, n_i\}$;

Se L é uma linguagem com igualdade, então:

- 3) Para cada L-termo t, $\{t=t\} \in \Psi$ □

Definição 3.1.4 - Um ramo de um tableau analítico é completo se e somente se a união de todos os seus nós é um conjunto de Hintikka em relação a L. □

Um ramo completo indica como construir um contra-exemplo para ϕ .

Definição 3.1.5 - Um tableau analítico é completo se e somente se cada ramo é fechado ou algum ramo é completo. □

Definição 3.1.6 - Uma dedução de uma fórmula σ a partir de um conjunto de fórmulas Σ é um tableau analítico fechado para $\Sigma \cup \{\neg \sigma\}$. Neste caso σ é um teorema de Σ em L. □

Podemos mostrar que o método dos tableaux analíticos é coerente e completo, no seguinte sentido:

Teorema 3.1.1 (BM, Sm): Seja Σ um conjunto de L-fórmulas e σ uma L-fórmula. Então σ é se e somente se $\Sigma \vdash \sigma$. □

Exemplo 3.1.1 - Mostraremos pelo método do tableau analítico que, para

$\Sigma = \{\forall x(\alpha \rightarrow \beta)\}$ e $\sigma = \exists x \alpha \rightarrow \beta$, temos $\Sigma \vdash \sigma$. Para isso mostraremos que existe um tableau analítico fechado cujo nó inicial é $\Sigma \cup \{\neg \sigma\}$. Apresentaremos a estrutura espacial do tableau em vez de usar a notação de Dewey por considerarmos aquela mais clara para fins de exemplo. Um "X" sob um ramo do tableau analítico indica que aquele ramo está fechado. Os números inteiros, entre parênteses, à esquerda de uma fórmula, identificam-na unicamente; os números inteiros, entre parênteses, à direita, de uma fórmula indicam a origem da fórmula no nó e, que a fórmula de origem já foi analisada naquele ramo.

O tableau analítico da Figura 3.1.1 mostra que existe uma confutação para

se e somente se σ é um teorema de Σ em $\mathcal{S}(L \vdash \sigma)$ (Ver Definição 3.4.1).

O sistema formal \mathcal{S} consiste das regras SI, SF e SP anteriormente apresentadas e das seguintes regras específicas para XEIDs:

Regra XEID: Se entre as fórmulas de um ramo terminado em um nó ϕ_k existir uma fórmula da forma

$$\forall x_1 \dots \forall x_n (A_1 \wedge A_2 \wedge \dots \wedge A_n) \Rightarrow (\exists y_1 \dots \exists y_r) (B_1 \wedge \dots \wedge B_s)$$

e fórmulas A_1, \dots, A_n tais que existe uma tupla de termos $\bar{t} = \langle t_1, \dots, t_n \rangle$ tais que $A_i = A_i(\bar{t}/\bar{x})$, então acrescente um novo nó

$$B_1(\bar{y}/\bar{u}) (\bar{x}/\bar{t}), \dots, B_s(\bar{y}/\bar{u}) (\bar{x}/\bar{t})$$

como sucessor de ϕ_k , e onde \bar{u} deve ser uma tupla de variáveis que não tenham ocorrido no ramo sendo expandido.

Regra - XEID: Se entre as fórmulas de um ramo terminado em um nó ϕ_k existir uma fórmula da forma

$$\neg (\forall x_1 \dots \forall x_n (A_1 \wedge A_2 \wedge \dots \wedge A_n) \Rightarrow (\exists y_1 \dots \exists y_r) (B_1 \wedge \dots \wedge B_s))$$

acrescente novos nós $\{A_1(\bar{x}/\bar{u}), A_2(\bar{x}/\bar{u}), \dots, A_n(\bar{x}/\bar{u}), \neg B_1(\bar{y}/\bar{u}) (\bar{x}/\bar{t}), \dots, \{A_1(\bar{x}/\bar{u}), A_2(\bar{x}/\bar{u}), \dots, A_n(\bar{x}/\bar{u}), \neg B_s(\bar{y}/\bar{u}) (\bar{x}/\bar{t})\}$ como sucessores de ϕ_k , onde \bar{t} pode ser qualquer tupla de variáveis e \bar{u} deve ser uma nova tupla de variáveis que não tenham ocorrido no ramo sendo expandido.

As noções de tableau completo e de dedução são transportadas, sem modificações, da Seção 3.1, exceto que o conjunto de regras agora é (SI, SF, SP, XEID, -XEID).

O teorema 3.1.1 também pode ser adaptado para demonstrar que este novo sistema formal é consistente e completo para XEIDs:

Teorema 3.2.1: Seja um conjunto de XEIDs e uma XEID.

Então $\Sigma \vdash \sigma$ se e somente se $\Sigma \vdash \sigma$ □

Os exemplos 3.2.1 e 3.2.2 que seguem objetivam mostrar aplicações simples do método dos tableaux analíticos como procedimento de decisão para dependências de dados.

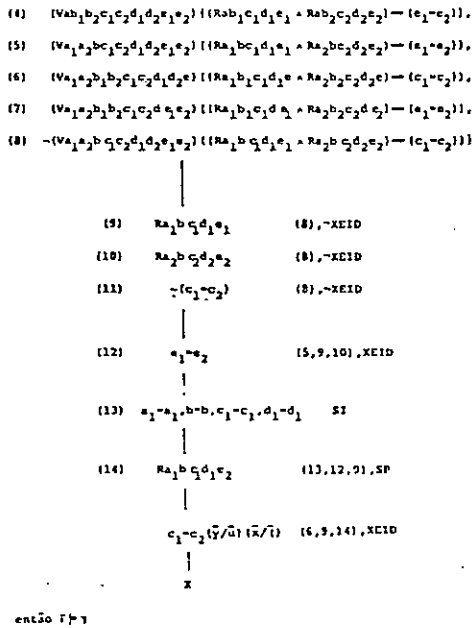
Exemplo 3.2.1 - Seja $R(ABCD)$ um esquema de relação e $\Gamma = \{R: A \rightarrow B,$

$R: B \rightarrow D, R: B \rightarrow C\}$ e $Y = \{R: A \rightarrow D\}$ uma dependência funcional que se deseja determinar se é uma implicação lógica de Γ .

Mostraremos que Y é uma dedução de Γ , $\Gamma \vdash Y$ e como o método dos tableaux é semanticamente coerente, então $\Gamma \models Y$.

Construiremos o tableau a partir do nó inicial $\Sigma \cup \{\neg \sigma\}$ como segue:

1		$\{(\forall a b_1 b_2 c_1 c_2 d_1 d_2) \{ (R a b_1 c_1 d_1 \wedge R a b_2 c_2 d_2) \rightarrow (b_1 = b_2) \}$
2		$\{(\forall a_1 a_2 b c_1 c_2 d_1 d_2) \{ (R a_1 b c_1 d_1 \wedge R a_2 b c_2 d_2) \rightarrow (d_1 = d_2) \}$
3		$\{(\forall a_1 a_2 b c_1 c_2 d_1 d_2) \{ (R a_1 b c_1 d_1 \wedge R a_2 b c_2 d_2) \rightarrow (c_1 = c_2) \}$
4		$\neg (\forall a b_1 b_2 c_1 c_2 d_1 d_2) \{ (R a b_1 c_1 d_1 \wedge R a b_2 c_2 d_2) \rightarrow (d_1 = d_2) \}$
5		$R a b_1 c_1 d_1$ (4), ~XEID
6		$R a b_2 c_2 d_2$ (4), ~XEID
7		$\neg (d_1 = d_2)$ (4), ~XEID
8		$b_1 = b_2$ (2, 5, 6), XEID
9		$a = a, c_2 = c_2, d_2 = d_2$ SI
10		$R a, b_1, c_2, d_2$ (9, 8, 6), SP
11		$d_1 = d_2$ (2, 5, 10), XEID
		X
		logo $\Gamma \models \sigma$ □



então $\Gamma \vdash \gamma$

3.3 - Tableaux e Tableaux Analíticos de Primeira Ordem - Uma Comparação dos Métodos

O método dos tableaux foi definido por Aho, Sagiv e Ullman (ASU) e uma versão simplificada foi apresentada por Maier, Mendelzon e Sagiv (MMS). Este método pode ser modificado, incluindo-se a remoção de certas restrições da definição original, para funcionar como procedimento de decisão para dependências de inclusão na presença de dependências funcionais para uma classe restrita de esquemas. (MMS,Mo).

Uma análise do método dos tableaux mostra que:

- Como procedimento de decisão para dependências funcionais, no qual o método opera sobre uma só tabela, as regras de inicialização e transformação são facilmente aplicadas e o método eficiente.
 - Como procedimento de decisão para dependências funcionais em presença de dependências de inclusão, é necessário criar uma tabela para cada esquema de relação. Isto complica o método e o torna mais trabalhoso. Mas, por outro lado, a generalização é simples pois os mecanismos de construção permanecem basicamente os mesmos.
 - Se tentarmos generalizar o método dos tableaux para XEIDs, teremos um passo de inicialização semelhante à regra XEID. Portanto, de início, é necessário introduzir análise de casos no método dos tableaux. A análise de casos complica, no entanto, o método pois é necessário indicar que tuplas pertencem a que tabelas e em que casos. Para outros tipos de dependências como dependências de exclusão (CV), análise de casos pode se tornar mais essencial ainda.
- É neste ponto que o método dos tableaux analíticos de primeira ordem, como discutido nas Seções 3.1 e 3.2, torna-se interessante. O fato de uma tupla estar em uma tabela é substituído aqui pela existência de uma fórmula da forma $R(t_1, \dots, t_n)$; e a idéia de análise de casos é inerente à estrutura em árvore dos tableaux analíticos.

Podemos então concluir que os dois métodos são bastante semelhantes na sua filosofia.

Para classes de dependências mais simples, o método dos tableaux, como discutido em Teoria de Banco de Dados (Ma), é mais conveniente por usar uma representação dos fatos mais simples: tabelas. Para classes de dependências mais complexas, o método dos tableaux analíticos, adaptado da lógica de primeira ordem, pode ser preferível por adotar uma representação natural para análise de casos.

Para finalizar, observamos que ambos os métodos podem ser transformados em procedimentos de decisão para determinadas classes de dependências de dados. Em particular, ambos são procedimentos de decisão para FDs, MVDs e outras classes cujo problema de decisão é solúvel.

5. Conclusões e Direções de Pesquisa

Após uma introdução sobre linguagens de primeira ordem e definições preliminares sobre Bancos de Dados, apresentamos uma formalização para especificar classes de dependências de dados através das Dependências Implícitas Embutidas e Estendidas - XEIDs.

Em seqüência apresentamos como o método dos tableaux analíticos de primeira ordem pode ser aplicado à teoria de Banco de Dados, particularmente como processo de decisão.

Uma comparação entre o método dos tableaux e o método dos tableaux analíticos de primeira ordem mostra que a necessidade de generalizar o primeiro método acarreta a introdução de análise de casos. É, neste ponto, que o método dos tableaux analíticos torna-se mais interessante, pois a idéia de análise de casos é inerente à estrutura em árvore dos mesmos.

Concluimos que os dois métodos são bastante semelhantes em sua filosofia. Para classes de dependências de dados mais simples, o método dos tableaux, como discutido na teoria de Banco de Dados é mais simples por usar uma representação mais simples: tabelas. Para classes de dependências mais complexas, o método dos tableaux analíticos, adaptado da lógica de primeira ordem, pode ser preferível por adotar uma representação natural para a análise de casos.

Finalizamos, observando que ambos os métodos podem ser transformados em procedimentos de decisão para determinadas classes de dependências. Em particular, ambos são procedimentos de decisão para FDs, MVDs e outras classes cujo problema de decisão é solúvel.

Futuros trabalhos devem ser orientados para estender novas regras de inferência que permitam a aplicação do método dos tableaux analíticos para classes mais gerais de dependências. Casos especiais de esquemas relacionais restritos que permitam reduzir o problema da indecidibilidade de algumas outras classes de dependências devem ser pesquisados para aumentar a viabilidade prática dos métodos.

Bibliografia

ABU - A.V. Aho; C. Beeri and J.D. Ullman, "The Theory of Joins in Relational Databases", ACM TODS vol. 4, nº 3, Sep. 1979.
 ANSI - "ANSI/X3/SPARC Study Group on Data Base Management Systems", Interim Report, ACM SIGMOD, vol. 7, nº 2, 1975.
 ASU - A.V. Aho; Y. Sagiv and J.D. Ullman, "Equivalence among Relational Expressions", SIAM J. Computing 8, 2, 1979.
 BM - J.L. Bell; M. Machover, "A Course in Mathematical Logic", North-Holland, 1977.
 Ca - M.A. Casanova, "A Theory of Data Dependencies over Relational Expressions", PUC - Technical Report 108104, 1981.
 CFP - M.A. Casanova; R. Fagin and C. Papadimitriou, "Inclusion Dependencies and their Interactions with Functional Dependencies, Conf. on Principles of Database Systems, Mar. 1982.
 Col - E.F. Codd, "A Relational Model of Data for Large Shared Data Banks", Comm. ACM vol. 13, nº 6, Jun, 1970.
 CV - M.A. Casanova; V.M.P. Vidal, "Towards a Sound View Integration Methodology", PODS 1983.
 En - H.B. Enderton, "A Mathematical Introduction to Logic", Academic Press, 1972.
 Fal - R. Fagin, "Multivalued Dependencies and a Normal Form for Relational Databases", ACM TODS, vol. 3, nº 2, Sep. 1977.
 Fa2 - R. Fagin, "A Normal Form for Relational Databases that is based on Domains and Keys", ACM TODS, vol. 6 nº 3 Sep. 1981.
 Fa3 - R. Fagin, "Horn Clauses and Database Dependencies", Proc. ACM-SIGAT Symp. Theory of Computing, 1980.
 Ho - P. Honeyman, "Testing Satisfaction of Functional Dependencies", Journal of ACM, vol. 29, nº 3, 1982.
 KP - A. Klug; R. Price, "Determining View Dependencies Using Tableaux", ACM-TODS, vol. 7, nº 3, 1982
 Ma - D. Maier, "Relational Database Theory", Computer Science Press, Maryland, 1983.
 Mo - C.M.O. Moura, "Procedimentos de Decisão para Dependências de Dados", a ser publicado.
 MMS - D. Maier; A. Mendelzon and Y. Sagiv, "Testing Implications of Data Dependencies", ACM-TODS, vol. 4, nº 4, Dec. 1979.
 Ni - J.M. Nicolas, "Mutual Dependencies and some Results on Undecomposable Relations", IV VLDB, 1978.
 Sm - R.M. Smullyan, "First Order Logic", Springer-Verlag, 1968.

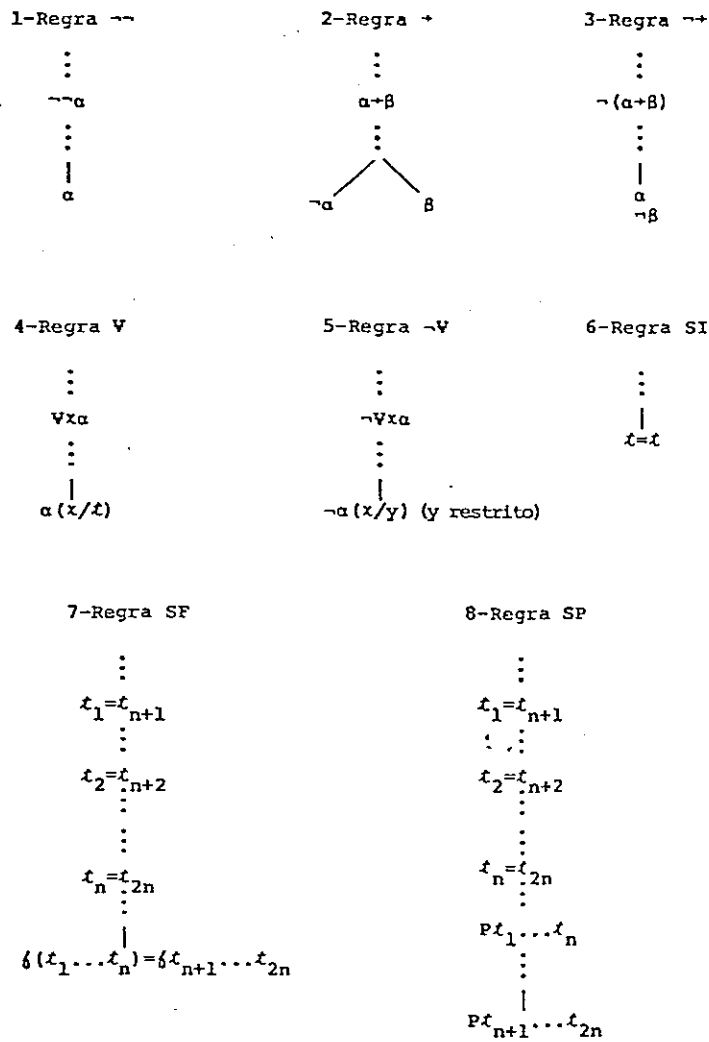


Figura 3.4.1 - Regras para o tableau analítico de primeira ordem

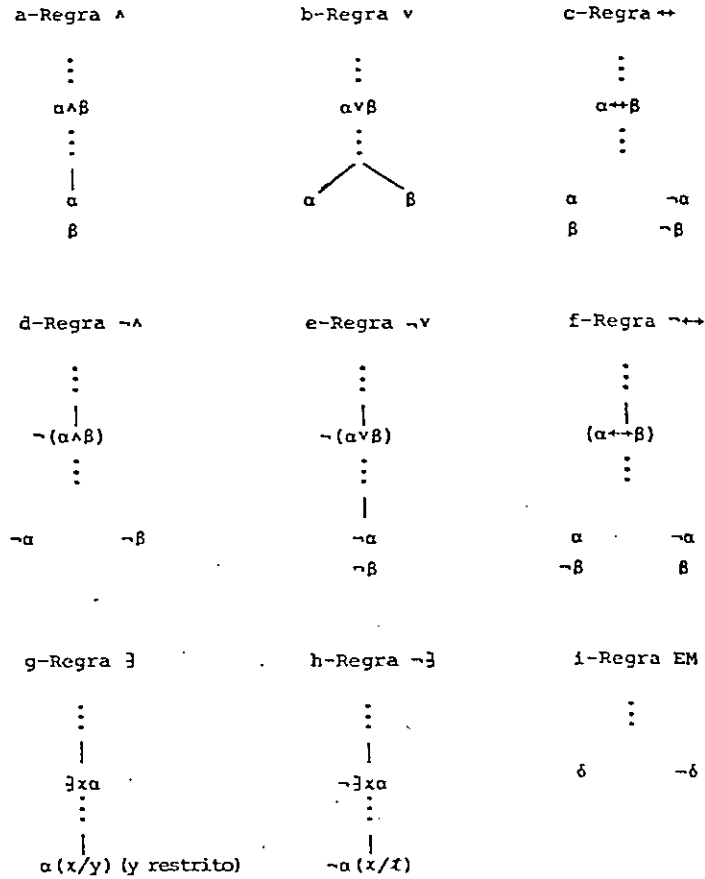
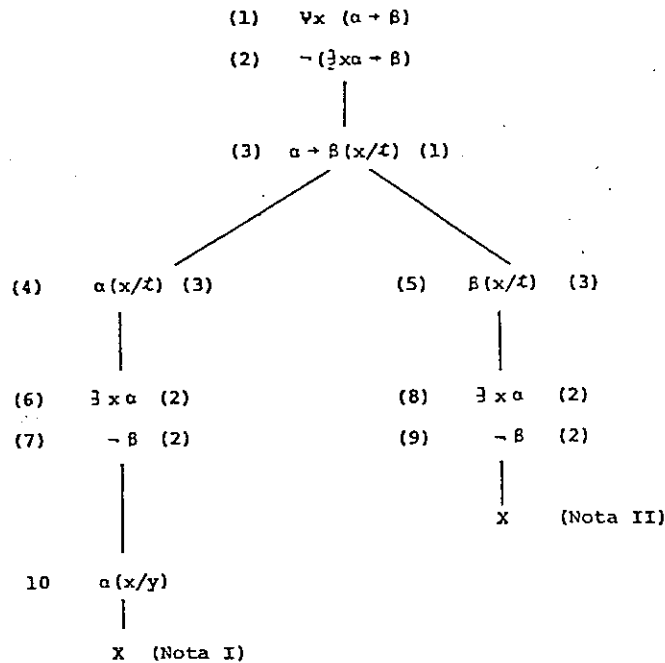


Figura 3.1.2 - Regras adicionais para o tableau analítico de primeira ordem.



Nota I - $\alpha(x/y)$ com y restrito confuta com $\neg \alpha(x/t)$ quando se faz $t = y$.

Nota II - $\neg \beta$ confuta com $\beta(x/t)$ para qualquer t

Figura 3.1.1 - Tableau analítico para