ICOT Technical Memorandum: TM-0445他

TM-0445他

学会発表論文(Data Base)

第3研究室

March, 1988

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03) 456-3191~5
Telex ICOT J32964

Institute for New Generation Computer Technology

TM 0445　ロジックデータベースのConsistencyチェックに関する一方法
庄司　功他

TM 0446　例外のある性質継承に関する並列アルゴリズム
毛受　哲他

TM 0447　演繹データベースにおける制約伝播の実現方式
阿比留幸展他

TM 0448　分散演繹データベースにおける問い合せ処理方式
高杉哲朗他

TM 0457　知識ベースマシンMu-X(1)－項集合に対する検索言語－
物井秀俊他

TM 0458　知識ベースマシンMu-X(2)－並列処理のための基本機能
酒井　浩他

TM 0459　知識ベースマシンMu-X(3)－ハードウェア実験機によるシミュレーション
伊藤文英他

TM 0460　知識ベースマシンMu-X(4)－項に対するインデックス方式の評価－
仲瀬明彦他

ロジックデータベースのConsistencyチェックに関する一方法

庄司 功　伊藤 英則
(財) 新世代コンピュータ技術開発機構

1. はじめに

ロジックデータベースにおいて，データを論理式で表わす方法については既に（例えば，文献［1］，［2］などで）いろいろと試みられている。そこでは通常データの最小単位はファクトであり，これは論理の言葉でいえば一つの原始命題に相当する。

しかし，データベースの対象を構成する個々の事象が必ずしも原始命題に置き換えられるというわけではない。例えば，事象A，Bのいずれが成立するかは不明であるが，AまたはBが成立することは確実であるという事象をデータベース上に表現するような場合を考えればわかるだろう。

この例からもわかるように，データの最小単位を，原始命題からより一般的な形の命題に拡張することによって，データベースの扱う対象範囲を広げることが期待できる。

ところが，一般的な形の命題を取り扱うとすれば，データベースのConsistencyを保障することや，与えられた命題がこのような命題達から論理的に帰結されるかどうかを判定すること，が難しくなると予想される。

本稿では一般的な形の命題を構成単位とするデータベースのConsistencyをチェックする方法，及び，与えられた命題が，このような命題達から構成されるデータベースから論理的に導き出されるか否かを判定する方法について述べる。更に，これらの方法に基づいて，データの冗長性やデータベースの更新について述べる。

2. 判定方法

本稿では命題論理で表現されるデータベースについて検討する。

命題$A_1,\dots,A_k$をデータとするデータベースは，$A_1,\dots,A_k$を公理とするTheory Tとみなされる。ここに，各A_iは原始命題$P_1,\dots,P_n$及び論理結合子から構成されているものとする。

例えば，事象E_1, E_2, E_3が成り立つという主張をそれぞれP_1, P_2, P_3という原始命題で表わすとき，E_1またはE_2，E_2またはE_3，E_3またはE_1，がそれぞれ成立することが確実であるという事情の下にデータベースを構築する場合，これを$P_1 | P_2, P_2 | P_3, P_3 | P_1$を公理とするTheoryとみなすということである。

ところで，形式論理の代数化としてブール代数が知られているが，本稿では有限個（原始命題と同数個）の元$a_1,\dots,a_n$から生成されるブール代数を導入し，これと先のTheory Tとの対応を考える。

原始命題$P_1,\dots,P_n$と論理結合子から構成される論理式の集合から元$a_1,\dots,a_n$から生成されるブール代数への写像fを次のように定める。

(1)　$f(P_i) = a_i$

(2)　$f(P_i | P_j) = a_i + a_j$

(3)　$f(P_i \& P_j) = a_i * a_j$

(4)　$f(\neg P_i) = a_i'$　(a_i'はa_iの補元)

但し，$1 \leq i,j \leq n$とする。

このとき，Theory Tの公理$A_1,\dots,A_k$に対して$f(A_i) = x_i$としx_1からx_kの積$M = x_1 * \dots * x_k$とすると，以下の命題1及び命題2が成り立つ。

［命題1］

Theory Tがinconsistentであるための必要十分条件は$M = 0$となることである。

［命題2］

命題BがTheory Tの定理であるための必要十分条件は$M * f(B) = M$となることである。

命題1によって，Theory Tの，言い換えればデータベースの，Consistencyをチェックすることができる。また，命題2によって，与えられた命題がデータベースから論理的に帰結されるか否かを判定することができる。

［例1］

事象E_1, E_2, E_3が成立するという主張をそれぞれ命題P_1, P_2, P_3と表わす。このとき以下の各命題が真である，即ち，各命題が主張する事象が成り立つ場合，これらの事実の下に構築されるデータベースがConsistentであるかどうかを判定する。

A_1 :　$(P_1 \& P_2) | (P_1 \& P_3)$

A_2 :　$(P_1 \& P_2) | (P_2 \& P_3)$

A_3 :　$(\neg P_1 \& P_2) | (P_1 \& \neg P_2)$

ここで$f(P_1) = a$，$f(P_2) = b$，$f(P_3) = c$とすると，

$f(A_1) = a*b + a*c$

$f(A_2) = a*b + b*c$

$f(A_3) = a'*b + a*b'$

Consistency Check of Logic Datatbase
Isao SHOJI, Hidenori ITOH
ICOT

となる。そこでMを計算すると，

$$M = (a*b+a*c)*(a*b+b*c)*(a'*b+a*b') = (a*b+a*b*c)(a'*b+a*b') = 0$$

となるから，このデータベースはInconsistentであることがわかる。

［例２］

P1, P2, P3の f による像を例１の通りとする。このとき，以下の各命題を公理とするTheory Tにおいて P1｜(P2&P3)がTの定理かどうかを判定する。

A1 ： P1｜P2

A2 ： P2｜P3

A3 ： P3｜P1

$f(A1) = a+b$

$f(A2) = b+c$

$f(A3) = c+a$

$f(P1 | (P2 \& P3)) = a+b*c$

ここでMを計算すると，

$$M = (a+b)*(b+c)*(c+a) = (b+a*c)*(c+a) = a*b+b*c+c*a$$

となる。次に，M＊f (P1|(P2&P3))を計算すると，

$$M*f(P1|(P2\&P3)) = (a*b+b*c+c*a)*(a+b*c) = a*b+b*c+c*a$$

従って，M＊f (P1｜(P2&P3)) ＝ Mとなるから，P1｜(P2&P3)はTの定理であることがわかる。

3．冗長性の判定

A1，...，Akを公理とするTheory Tにおいて，A1が他のA2，...，Akから論理的に帰結されるならば，A1は冗長な公理であるとみなすことは自然であろう。

そこで，A1が冗長であるということを上のように定義した場合，先に述べた方法を用いてA1の冗長性を言い換えると次のようになる。

A1が冗長であるとは $f(Ai) = xi\ (0 \leq i \leq k)$ として $x2*...*xk = x1*x2*...*xk$ となることである。

［例３］

P1&P2, P2&P3, P3&P1を公理とするTheoryにおいてP1&P2は冗長である。

実際，P1, P2, P3の f による像を例１の通りとすれば，

$f(P1\&P2) = a*b$

$f(P2\&P3) = b*c$

$f(P3\&P1) = c*a$

$M = (a*b)*(b*c)*(c*a) = a*b*c$

$f(P2\&P3)*f(P3\&P1) = a*b*c$

となるからである。

［注意１］

一つの公理の集合から得られる冗長でない公理の集合は必ずしも一意的に決まらない。更に，それら両者の要素の個数も必ずしも等しいとは限らない。

例えば，公理の集合として｛P1&P2, P3&P4, P5&P6, P1&P2&P3, P4&P5&P6｝が与えられたとする。このとき，二つの集合｛P1&P2, P3&P4, P5&P6｝と｛P1&P2&P3, P4&P5&P6｝は共に冗長でない公理の集合であるが明らかに要素の個数が異なる。

4．データベースの更新

本稿ではデータの削除と追加についてのみ述べる。

4．1．データの削除

データの削除においては，冗長なデータ，言い換えれば冗長な公理を含んでいる場合とそうでない場合とに分ける。A1，...，Akを公理とし，A1が冗長な公理であるとする。このときA1を削除しても高々冗長な公理を除去したという効果にとどまり，A1は依然としてA2，...，Akから導かれる。他方，A1が冗長でない場合，A1は決してA2，...，Akから導かれないから，A1の主張する仮定は完全に捨てられる。

4．2．データの追加

A1，...，Akを公理とし，これにA0を追加する場合を考える。ここで $M = f(A1)*...*f(Ak)$ と置く。

（１） $M*f(A0) = M$ のとき

A0はA1，...，Akを公理とするTheoryの定理である。A0を追加すると冗長な公理の集合が生成される。

（２） $M*f(A0) = 0$ のとき

A0を追加するとInconsistentなTheoryが得られる。A0を追加しかつConsistencyを維持するには，$f(A0)*f(Ai) = 0$ ならばAiを捨て，そうでなければ公理として採用する。このようにして作られた新たな公理の集合はConsistencyを維持する。

（３）（１），（２）以外

A0は冗長な公理でなくかつ新たに生成された公理の集合はConsistencyを維持する。

［注意２］

A1，...，Akが冗長でない公理の集合であってもA0を加えた公理の集合A0, A1，...，Akは必ずしも冗長でない公理の集合とは限らない。

例えば，P1&P2, P2&P3は冗長でない公理の集合であるが，これにP3&P4を加えると冗長な公理の集合が得られる。

［参考文献］

[1] Gallaire, H., et. al, "Logic and Database:A Deductive Approach", ACM Computing Surveys, 1984.

[2] Reiter, R., "On Closed World Data Bases", Logic and Data Bases, 1978