# 関係論理から最適な関係代数表現への変換法†

中野良平†† 斉藤和巳††

主な関係データベース言語 SQL, QUEL 等は関係論理を基にしているが，データベースマシンのサポート言語は多くの場合関係代数である．したがって，ユーザインタフェースに優る関係論理で表現した検索を，データベースマシンでの実行を想定して，最適な関係代数表現に変現する研究が重要となる．関係論理から関係代数への変換は，Codd, Ullman, Klug らによって研究されたが，いずれも両言語の記述力が同等であることの証明に主眼があるため，簡単な例の場合にも，一般に実行効率の極めて良くない関係代数表現を生成する．筆者らは，関係論理表現を最適な関係代数表現に変換する新変換体系を提案する．新変換法は関係論理表現を再帰構造に沿って逐次的に変換する中で，基本変換則に加え発見的変換則を用意し，それを優先的に適用することにより関係代数レベルの最適化を行う．本論文では関係論理と関係代数を規定した後，新変換法の特徴と処理手順を述べ，次いで，新変換法の詳細を基本変換と発見的変換に分けて，定理の形で述べる．さらに，変換プログラムを prolog で作成し，約 100 件の検索例に適用し所期の結果を得たことを示す．

## 1. まえがき

関係モデルの検索系は多様であるが，組関係論理，定義域関係論理，および関係代数を3極とする[1]．主な関係データベース言語 SQL[2]，QUEL[3]等は組関係論理を基にするが，近年商用機が登場し始めたデータベースマシンは多くの場合関係代数を実現する[4]．したがって，ユーザインタフェースに優る組関係論理(以下，単に関係論理と呼ぶ)で表現した検索を，データベースマシンでの実行を想定して，最適な関係代数表現に変換する研究が重要となる．なお，ソフトウェアのみで構成された従来の関係データベースシステムは，基本的にはネスト繰返し法[5]を用いて検索を実行しており，純粋の関係代数表現への変換は行っていない．

関係論理表現から関係代数表現への変換は，Codd[6]，Ullman[7]，Klug[8]らによって研究されたが，いずれも両言語の記述力が同等であることの証明に主眼があるため，簡単な例の場合にも，一般に実行効率の極めて良くない関係代数表現を生成する[9]ので，実行を前提とした変換法として使えない．

また，SQL を最適正準形に変換する Kim の研究[10]は，対象言語は違うが，本論文と類似した狙いを持つ．しかし，その変換体系は，比較演算子を等号に限定，準結合の考慮なし，除算のパターンが1種類のみ等，本論文で提案する変換体系の部分体系に当たる．

本論文では関係論理表現を最適な関係代数表現に変換する新変換体系を提案する．ただし，Codd, Ullman と同様，集約関数，四則は除外する．以下の章では，まず，関係論理と関係代数を規定した後，新変換法の特徴と処理手順を述べ，次いで，新変換法の詳細を定理の形で述べ証明を付す．さらに，変換プログラムを prolog で作成し約 100 件の検索例に適用し所期の結果を得たことを示す．

## 2. 関係論理と関係代数

### 2.1 関係論理

Klug[8] は Codd[6] の関係論理を基に，範囲式，集約関数等の記法拡張を行った．本論文の関係論理は Klug のに準じ，その構文規定を**図1**とする．

アルファは関係論理の基本要素であり，派生関係を規定する．アルファの形式は以下である．

目標リスト：範囲規定：条件式

範囲規定は組変数の動く範囲を規定する．条件式を満足する目標リストが取り出され派生関係を構成する．より具体的なアルファの形式は，$t_i$ を項，$R_j$ を範囲，$u_j$ を組変数，$R_j(u_j)$ を範囲式，$f$ を条件式とするとき，

$$(t_1, \cdots, t_n) : R_1(u_1), \cdots, R_m(u_m) : f$$

であり，SQL の select-from-where 構文に対応する．

アルファの意味規定はスキーマの任意のインスタンス $I$ の下で以下とする．これは以降の証明で使用する．

$$\{(t_1, \cdots, t_n) : u_1 \in R_1(I) \wedge \cdots \wedge u_m \in R_m(I) \wedge f(u_1, \cdots, u_m) = 1\}$$

関係論理の記述力を増すために，アルファを範囲と

† Reduction from Relational Calculus to Optimal Algebraic Expression by RYOHEI NAKANO and KAZUMI SAITOH (Knowledge Systems Laboratory, NTT Communications and Information Processing Laboratories).

†† NTT 情報通信処理研究所知識処理研究部

〈アルファ〉 ::= 〈関係スキーム〉
| 〈目標リスト〉:〈範囲規定〉:〈条件式〉
〈関係スキーム〉 ::= 〈文字列〉
〈目標リスト〉 ::= (〈目標要素〉)
〈目標要素〉 ::= 〈項〉 | 〈組変数〉 | 〈目標要素〉, 〈目標要素〉
〈項〉 ::= 〈定数〉 | 〈組変数〉 [〈属性番号〉]
〈組変数〉 ::= 〈英小文字〉
〈範囲規定〉 ::= 〈範囲式〉 | 〈範囲式〉, 〈範囲規定〉
〈範囲式〉 ::= 〈範囲〉 (〈組変数〉)
〈範囲〉 ::= 〈範囲項〉 | 〈範囲〉∨〈範囲項〉
〈範囲項〉 ::= 〈範囲因〉 | 〈範囲項〉∧〈範囲因〉
〈範囲因〉 ::= 〈範囲元〉 | 〈範囲因〉∧～〈範囲元〉
〈範囲元〉 ::= 〈アルファ〉 | (〈範囲〉)
〈条件式〉 ::= 〈ブール項〉 | 〈条件式〉∨〈ブール項〉
〈ブール項〉 ::= 〈ブール因〉 | 〈ブール項〉∧〈ブール因〉
〈ブール因〉 ::= 〈ブール元〉 | ～〈ブール元〉
〈ブール元〉 ::= 〈原子式〉 | (〈条件式〉)
| 〈限量子〉〈範囲式〉〈条件式〉
〈原子式〉 ::= 〈項〉〈比較演算子〉〈項〉
〈比較演算子〉 ::= = | <> | < | <= | > | >=
〈限量子〉 ::= ∃ | ∀

図 1 関係論理の構文
Fig. 1 Syntax of relational calculus.

して指定可能とする．これによりアルファのネスト表現が可能になり，SQL のような構造化表現が自然に記述できる（図 2）．なお，限量子はすべて，Codd の関係論理と同様，範囲式を伴うもの（範囲付き）とする．

以上規定した関係論理表現は安全である．すなわち常に評価可能である．証明を付録に示す．これは関係代数への変換が常に可能であることを裏付ける．

### 2.2 関係代数

本論文の関係代数は Codd の提案[6]を基本に，選択

（検索文） a 型の全商品を 2 階の店に出荷している会社を求めよ。

関係スキーム：supply(会社, 店, 商品, 数量),
class(商品, 型), loc(店, 階)

(u[1]):supply(u):∀((t):class(t):t[2]=a)(v)
∃((s):supply(s):∃loc(r)(r[1]=s[2]∧r[2]=2))(w)
(w[1]=u[1]∧w[3]=v[1])

図 2 関係論理で表現した検索の例
Fig. 2 Query example expressed by relational calculus.

〈関係代数表現〉 ::= 〈関係スキーム〉
| 〈関係代数表現〉〈単項演算〉
| 〈関係代数表現〉〈二項演算〉〈関係代数表現〉
| (〈関係代数表現〉)
〈単項演算〉 ::= 〈射影〉 | 〈選択〉
〈二項演算〉 ::= 〈結合〉 | 〈準結合〉 | 〈除算〉 | 〈集合演算〉
〈射影〉 ::= [〈属性リスト〉]
〈属性リスト〉 ::= 〈属性〉 | 〈属性〉, 〈属性リスト〉
〈属性〉 ::= #〈属性番号〉
〈選択〉 ::= [〈選択条件式〉]
〈選択条件式〉 ::= 〈選択項〉 | 〈選択条件式〉∨〈選択項〉
〈選択項〉 ::= 〈選択元〉 | 〈選択項〉∧〈選択元〉
〈選択元〉 ::= 〈選択原子式〉 | (〈選択条件式〉)
〈選択原子式〉 ::= 〈属性〉〈比較演算子〉〈定数〉
| 〈属性〉〈比較演算子〉〈属性〉
〈結合〉 ::= [〈結合条件式〉] | []
〈結合条件式〉 ::= 〈結合述語〉 | 〈結合条件式〉, 〈結合述語〉
〈結合述語〉 ::= 〈属性〉〈比較演算子〉〈属性〉
〈準結合〉 ::= [∃ ; 〈結合条件式〉] | [～∃ ; 〈結合条件式〉]
〈除算〉 ::= [(〈属性リスト〉) ／ (〈属性リスト〉)]
〈集合演算〉 ::= [+] | [*] | [−]

図 3 関係代数の構文
Fig. 3 Syntax of relational algebra.

(検索文) a 型の全商品を 2 階の店に出荷している会社を求めよ.
((supply[∃ ; #2=#1] (loc[#2=2]))[#1, #3])
[(#2)/(#1)] (class[#2=a])

図 4 関係代数で表現した検索の例
Fig. 4 Query example expressed by relational algebra.

条件の論理式化，準結合の導入，および複数属性対による結合／準結合／除算の拡張を行ったものとする．演算の種類は，射影，選択，結合，準結合，除算，集合演算（和集合，積集合，差集合）である．その構文規定を図 3 に記す．関係代数表現の記述例を図 4 に示す．

属性は関係ごとに 1 から振られた属性番号で識別する．定数と区別するため，属性番号にはシャープ記号 (#) を冠する．なお，準結合は結合条件を満足する組を選ぶ通常の準結合のほかに，満足しないものを選ぶ否定型の準結合も用意する．後者は，Kim[10]の提案した否定結合に対応して，否定準結合とでも呼べる．

関係代数の各演算の意味規定を以下に記す．$e_1, e_2$ を関係代数表現，$A_1, A_2$ を属性リスト，$Fs$ を選択条件式，$F_j$ を結合条件式，$\phi$ を空集合，$I$ をスキーマの任意のインスタンスとするとき，

- 射影 $(e_1[A_1])(I)=\{t[A_1]\mid t\in e_1(I)\}$
- 選択 $(e_1[Fs])(I)=\{t\mid t\in e_1(I)\wedge t[Fs]=1\}$
- 結合 $(e_1[F_j]e_2)(I)$
  $=\{t\circ s\mid t\in e_1(I)\wedge s\in e_2(I)\wedge t[F_j]s=1\}$
  ただし，$t\circ s$ はタプル $t$ とタプル $s$ の合成を表す．
- 肯定準結合 $(e_1[\exists\,;F_j]e_2)(I)$
  $=\{t\mid t\in e_1(I)\wedge\{s\mid s\in e_2(I)\wedge t[F_j]s=1\}\langle\rangle\phi\}$
- 否定準結合 $(e_1[\sim\exists\,;F_j]e_2)(I)$
  $=\{t\mid t\in e_1(I)\wedge\{s\mid s\in e_2(I)\wedge t[F_j]s=1\}=\phi\}$
- 除算 $(e_1[A_1/A_2]e_2)(I)=\{t[\sim A_1]\mid t\in e_1(I)$
  $\wedge\{s[A_2]\mid s\in e_1(I)\wedge s[A_1]=t[A_1]\}$
  $\supseteq\{r[A_2]\mid r\in e_2(I)\}\}$
- 和集合 $(e_1[+]e_2)(I)=\{t\mid t\in e_1(I)\vee t\in e_2(I)\}$
- 積集合 $(e_1[*]e_2)(I)=\{t\mid t\in e_1(I)\wedge t\in e_2(I)\}$
- 差集合 $(e_1[-]e_2)(I)=\{t\mid t\in e_1(I)\wedge t\notin e_2(I)\}$

## 3. 新変換法

### 3.1 新変換法の特徴と処理手順

本論文で提案する変換法は次の特徴を有す．

（1） 最適な関係代数表現を生成

データベース検索の最適化は以下に大別される[7]．

- 第1種の最適化：関係代数演算の並び替え，置き替えに関するもので，関係の格納状態とは無関係に成立するという意味で論理レベルの最適化
- 第2種の最適化：関係の格納状態（キーやインデクス等）を利用した最適化で，格納レベルの最適化

本論文では，適用システムを制限しない観点から，論理レベル（第1種）の最適化のみを問題とする．“最適な”とは，以下の指標が満足されることを指す．

a）同じ結果を与える関係代数表現の中で，負荷の重い演算*1の回数が最少であること

b）負荷の重い演算の回数が同じ場合は，負荷の軽い演算*2の回数が最少であること

c）負荷の軽い演算を優先実行することにより，負荷の重い演算の実負荷を軽減すること

*1 負荷の重い演算：結合，除算，集合演算，射影

*2 負荷の軽い演算：選択，準結合

本変換法は上述の意味での最適な関係代数表現を生成することを目標とし，第一の特徴とする．

（2） 関係論理構文の再帰構造に沿った逐次変換方式

本変換法の入力となる関係論理表現は検索内容を簡潔かつ巧妙に反映していると考える．これは本変換法の前提条件である．この前提は，人間は検索表現を記述するとき，より簡潔なものを書こうと努力し指向するであろうという比較的自然な期待を背景にする．約100件の検索例を実際に記述してみて，複雑な検索の簡潔かつ巧妙な表現は多くの場合，関係論理の再帰構造を利用することがわかった．

本変換法は関係論理構文の再帰構造を重視し，基本的には関係論理表現の構文解析木に忠実に沿いながら逐次的に変換を進める．すなわち，関係論理表現の簡潔性，巧妙さを直截的に変換過程に反映する．この方式は，例えば，条件式を冠頭標準形[11]に変形してから変換する方法[6]とは対照的である．冠頭標準形に変形すると，かなり異なる条件式でも同一の標準形に帰着され，元の条件式が有していた微妙な情報を損なう．

（3） 最適な関係代数表現を生む発見的変換

関係論理表現を最適な関係代数表現に変換するには，一律の変換アルゴリズム（基本変換則）を用意するだけでは十分でない．我々人間は，検索文が与えられて，最適な関係代数表現を作ることができる．人間が関係代数表現を考える場合，何種類かの発見的ルール（○○のようなパターンの検索結果は△△の関係代数表現で求まるの類い）を試行錯誤的に適用し，最終的に無駄のない表現に到達すると思われる．したがって，本変換法では，人間が無意識のうちに利用していると思われる発見的変換則を多くの検索例の記述実験の中から探り出し，基本変換則に加えて用意する．なお，発見的変換則は基本変換則をどのように組み合わせても同じ変換効果が出ないもののみとし，他の変換則の組合せと等価なものは採り上げない．

（4） 条件式を構成する論理式の分類

従来変換法[6)~8)]では，条件式中の論理積や論理和を各々一律に積集合，和集合の演算に変換していたが，それでは最適な変換結果は得られない．選択，結合，または除算として変換すべきケースがあるためである．それを可能にするために，本変換法では，条件式を構成する論理式を**表1**のように分類し，その分類を参照して変換を進める[9]．

次に，本変換法の処理手順を述べる（**図5**）．まず，否定消去と論理式分類を行った後，関係論理表現の構文解析木に沿って，発見的変換を優先的に適用し，合致するものがないときに基本変換を適用するという順序で，完全に関係代数表現に変換されるまで，両者を繰り返しながら変換を進める．

否定消去は本格的変換に入る前の準備処理で，限量

表 1 論理式の分類
Table 1 Classification of formulas.

| 略称 | 名　称 | 意　味 |
|---|---|---|
| saf | 選択型原子式 | 現れる組変数が1種類の原子式 |
| eaf | 等結合型原子式 | 比較演算子が等号で，両側の組変数が異なる原子式 |
| jaf | 結合型原子式 | 比較演算子が等号以外で，両側の組変数が異なる原子式 |
| si | 選択型論理式 | 構成する原子式がすべて saf で，組変数がすべて同一の論理式 |
| ef | 等結合型論理式 | eaf が論理積で結ばれ，組変数対がすべて同一の論理式 |
| jf | 結合型論理式 | eaf または jaf が論理積で結ばれ，組変数対がすべて同一の論理式 |
| mf | 混合型論理式 | 上記のいずれでもない論理式 |

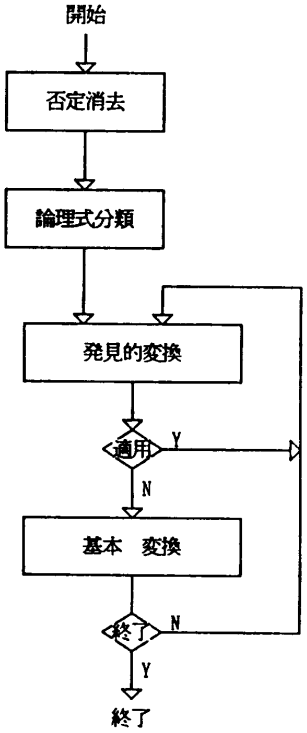


図 5 新変換法の処理手順
Fig. 5 General flow of the present reduction.

子直前の否定を除き，論理学でよく知られた変形[11]により条件式中の否定記号を消去する．

基本変換には，

- 範囲中に現れる論理演算子
- 選択型論理式
- 肯定存在限量子，否定存在限量子
- 結合型論理式
- 全称限量子
- 条件式中に現れる論理演算子
- 目標リスト

に関する変換則がある．また，発見的変換には，

- 除算(1)，除算(2)，除算(3)
- 結合省略
- 射影省略
- 肯定準結合，否定準結合

に関する変換則がある．

ここで，変換則の適用順序の問題を考える．同じ変換対象に対し複数の変換則が適用可能な場合，適用順序が異なれば一般に変換結果も異なる．すなわち，変換則の適用順序が変換結果の最適性に関与する．しかし，実際には，発見的変換が基本変換に優先するという大きな枠組みがあり，各々で閉じて考えられるので，変換則適用の競合問題は深刻ではない．発見的変換では除算が問題となる（3.3 節参照）．基本変換では，競合の可能性は条件式を扱う変換則に限られ，実は上の列挙順に適用するのがよいことがわかった．

### 3.2 基本変換

以降で使用する記号の意味を表 2 に示す．

前章で規定した関係論理と関係代数について，次のような変換の存在性が証明できる．同種の定理は従来研究でも与えられているが，証明は異なる．

［変換の存在定理］ 関係論理で表現した任意のアルファに対し，関係代数表現が存在する．

（証明） アルファを範囲とすることにより，アルファのネストができる．ネストしたアルファは最も内側から順に関係代数表現に変換する．最も内側のアルファの範囲規定に現れる範囲はすべて関係スキームであ

表 2 記号の説明
Table 2 Notation of symbols.

| 分　類 | 記　号 | 意　味 |
|---|---|---|
| 共　通 | $I$ | スキーマの任意のインスタンス |
| | 添字 $s$ | 選択型（論理式／原子式） |
| | 添字 $e$ | 等結合型（論理式／原子式） |
| | 添字 $j$ | 結合型（論理式／原子式） |
| | len($L$) | リスト $L$ の要素数 |
| | { } | 集合 |
| 関係論理 | $tl$ | 目標リスト |
| | $\Omega(\omega)$ | 範囲規定，ただし，$\Omega$ は範囲ベクトル，$\omega$ は対応する組変数ベクトル |
| | $x, y, \cdots$ | 組変数 |
| | $f, g, \cdots$ | 論理式 |
| 関係代数 | $X, Y, \cdots$ | 関係代数表現 |
| | $TL$ | 属性リスト（$tl$ に対応） |
| | $F, G, \cdots$ | 演算の条件式（$f, g, \cdots$ に対応） |

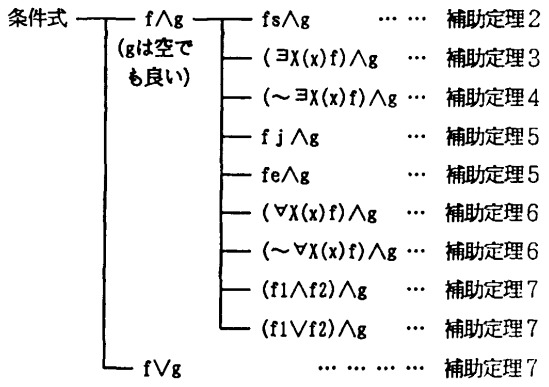


**図 6** 条件式の分解
Fig. 6 Breakdown of qualifier formula.

り，関係スキームは変換の必要がないので，最も内側のアルファはネストのないアルファであり，関係代数表現に変換できるとする（証明の後半参照）．そこで，1段外側のアルファを考える．その範囲規定に現れるアルファはすべて既に関係代数表現に変換済みであり，補助定理1を適用して範囲をすべて関係代数表現に変換できる．こうして，1段外側のアルファもネストのないアルファとなる．同様にすれば，多段にネストしたアルファもネストのないアルファの変換に帰着できる．

そこで，ネストのないアルファの変換を考える．範囲規定中の範囲はすべて関係代数表現に変換済みとする．今，条件式を網羅的に分解して考える（**図6**）．条件式に合わせて補助定理2から7を適用すれば，条件式はその再帰的な構造に沿って解決され，遂には空式に至る．条件式が空になれば，補助定理8を適用して，純然たる関係代数表現が得られる．　（証明終）

以下，補助定理1から8までを示す．

［補助定理1］　範囲中に現れる論理演算子

1. $(tl: (X_1 \wedge X_2)(x), \Omega(\omega): f)(I)$
$= (tl: (X_1[*]X_2)(x), \Omega(\omega): f)(I)$
2. $(tl: (X_1 \vee X_2)(x), \Omega(\omega): f)(I)$
$= (tl: (X_1[+]X_2)(x), \Omega(\omega): f)(I)$
3. $(tl: (X_1 \wedge \sim X_2)(x), \Omega(\omega): f)(I)$
$= (tl: (X_1[-]X_2)(x), \Omega(\omega): f)(I)$

（証明）　簡明なので省略．

［補助定理2］　選択型論理式

$(tl: X(x), \Omega(\omega): f_i(x) \wedge g)(I)$
$= (tl: X[F_i](x), \Omega(\omega): g)(I)$

（証明）

左辺$= \{(x, \omega)[TL]: x \in X(I) \wedge \omega \in \Omega(I) \wedge f_i(x) = 1 \wedge g = 1\}$
$= \{(x, \omega)[TL]: x \in [F_i](I) \wedge \omega \in \Omega(I) \wedge g = 1\}$
＝右辺

［補助定理3］　肯定存在限量子

$(tl: \Omega(\omega): (^{\exists}X(x)f) \wedge g)(I)$
$= (tl: \Omega(\omega), X(x): f \wedge g)(I)$

（証明）

左辺$= \{\omega[TL]: \omega \in \Omega(I) \wedge {}^{\exists}X(I)(x)(f(x, \omega) = 1) \wedge g = 1\}$
$= \{\omega[TL]: \omega \in \Omega(I) \wedge {}^{\exists}x(x \in X(I) \wedge f(x, \omega) = 1) \wedge g = 1\}$
$= \{\omega[TL]: \omega \in \Omega(I) \wedge x \in X(I) \wedge f(x, \omega) = 1 \wedge g = 1\}$
＝右辺

［補助定理4］　否定存在限量子

$(tl: \Omega(\omega): (\sim {}^{\exists}X(x)f) \wedge g)(I)$
$= ((z[1], \cdots, z[n]): (((tl, \Delta): \Omega(\omega): g)[-]((tl, \Delta): \Omega(\omega): {}^{\exists}X(x)f))(z): (\ ))(I)$

ただし，$n = \mathrm{len}(tl)$，$\Delta$ は $f$ 中の属性 $\omega[i]$ の集合．

（証明）

左辺$= \{\omega[TL]: \omega \in \Omega(I) \wedge g = 1 \wedge \sim {}^{\exists}x(x \in X(I) \wedge f(x, \omega) = 1)\}$
$= \{\omega[TL]: \omega \in \Omega(I) \wedge g = 1 \wedge \omega \notin \{\omega': \omega' \in \Omega(I) \wedge x \in X(I) \wedge f(x, \omega') = 1\}\}$
$= \{\omega[TL]: \omega \in (\{\omega'': \omega'' \in \Omega(I) \wedge g(\omega'') = 1\}[-]\{\omega': \omega' \in \Omega(I) \wedge x \in X(I) \wedge f(x, \omega') = 1\})\}$

差集合の前後で相違を生じるのは $\Delta$ であり，それを $TL$ に追加して目標リストとして絞り込んでも同じで，

$= \{(v[1], \cdots, v[n]): v \in (\{\omega''[TL, \Delta]: \omega'' \in \Omega(I) \wedge g(\omega'') = 1\}[-]\{\omega'[TL, \Delta]: \omega' \in \Omega(I) \wedge x \in X(I) < f(x, \omega') = 1\})\}$
$= \{(v[1], \cdots, v[n]): v \in (\{\omega''[TL, \Delta]: \omega'' \in \Omega(I) \wedge g(\omega'') = 1\}[-]\{\omega'[TL, \Delta]: \omega' \in \Omega(I) \wedge {}^{\exists}X(I)(x)f(x, \omega') = 1\})\}$ ＝右辺

［補助定理5］　結合型論理式

$(tl: X(x), Y(y), \Omega(\omega): f_j(x, y) \wedge g)(I)$
$= (tl': (X[F_j]Y)(w), \Omega(\omega): g')(I)$

ただし，$tl'$，$g'$ は各々 $tl$，$g$ 中の $x$ または $y$ に関する項を $w$ の項に置換したもの．なお，$f_j$ は $f_i$ でもよい．

（証明）

左辺$= \{(x \circ y \circ \omega)[TL]: x \in X(I) \wedge y \in Y(I)$

$\wedge\omega\in\Omega(\omega)\wedge f_j(x, y)=1\wedge g=1\}$

$=\{(w\circ\omega)[TL']: w\in(X[F_j]Y)(I)\wedge\omega\in\Omega(\omega)$

$\wedge g'=1\}$

$=$右辺

[補助定理6] 全称限量子

1. $(tl: \Omega(\omega): (^{\forall}X(x)f)\wedge g)(I)$

$=(tl: \Omega(\omega): (\sim{}^{\exists}X(x)(\sim f))\wedge g)(I)$

2. $(tl: \Omega(\omega): (\sim{}^{\forall}X(x)f)\wedge g)(I)$

$=(tl: \Omega(\omega): ({}^{\exists}X(x)(\sim f))\wedge g)(I)$

(証明)

1. 左辺$=(tl: \Omega(\omega): ({}^{\forall}x(\sim X(x)\vee f))\wedge g)(I)$

$=(tl: \Omega(\omega): (\sim{}^{\exists}x(X(x)\wedge\sim f))\wedge g)(I)$

$=$右辺

2. 1. とほぼ同様にして証明できる.

[補助定理7] 条件式中に現れる論理演算子

1. $(tl: \Omega(\omega): (f\wedge g)\wedge h)(I)$

$=(tl: \Omega(\omega): f\wedge(g\wedge h))(I)$

2. $(tl: \Omega(\omega): (f\vee g)\wedge h)(I)$

$=(tl: \Omega(\omega): (f\wedge h)\vee(g\wedge h))(I)$

3. $(tl: \Omega(\omega): f\vee g)(I)$

$=((tl: \Omega(\omega): f)[+](tl: \Omega(\omega): g))(I)$

(証明) 1, 2 は論理式の等値性より自明.

3. 左辺$=\{\omega[TL]: \omega\in\Omega(I)\wedge(f\vee g)=1\}$

$=\{\omega[TL]: \omega\in(\{\omega': \omega'\in\Omega(I)\wedge f=1\}$

$[+]\{\omega'': \omega''\in\Omega(I)\wedge g=1\})\}$

$=\{\omega[TL]: \omega\in\Omega(I)\wedge f=1\}$

$[+]\{\omega[TL]: \omega\in\Omega(I)\wedge g=1\}=$右辺

[補助定理8] 目標リスト

1. $(tl: X(x): (\ ))(I)=X[TL](I)$

2. $(tl: X(x), Y(y), \Omega(\omega): (\ ))(I)$

$=(tl': (X[\ ]Y)(z), \Omega(\omega): (\ ))(I)$

ただし, $tl'$ は $tl$ 中の $x$ または $y$ に関する項を $z$ の項に置換したもの.

(証明)

1. 左辺$=\{x[TL]: x\in X(I)\}$

$=\{y: y\in X[TL](I)\}=$右辺

2. 1. とほぼ同様にして証明できる.

### 3.3 発見的変換

変換の存在定理は基本変換のみを用いて証明した. すなわち, 基本変換だけでも変換は完結するが, より簡潔な関係代数表現を生成するには, 発見的変換を使用する必要がある. 発見的変換は一般に条件式の特定のパターンに反応し, 変換をバイパスする働きを持つ.

除算に関し, 3種の変換則を発見した. 補助定理9が最も汎用で, 以下 10, 11 の順になる. したがって, 適用順序はその逆でなければならない. 補助定理9の演算手順は結合 ($f$ が $f_j$ のとき) →射影→除算, 補助定理10は射影→除算→準結合, 補助定理11は射影→除算である. 補助定理10の存在意義は補助定理9に比して除算, 結合の対象データ量が一般に少なく, かつ結合が準結合で済む点にある.

準結合は存在限量子を伴う結合型論理式にちょうど符合する. 準結合は普通の結合に比べ, 存在の有無の確認のみでよく, 組を結合する必要がないため, 処理負荷が大幅に軽い. ここでは, 普通の肯定型準結合(補助定理14)だけでなく, 否定型(補助定理15)も用意する. 否定型は, 全称限量子の変形により生成されることがあり, これがないと, 補助定理4により複雑な変形を強いられるだけに, その存在意義は大きい.

その他, 結合省略, 射影省略も内容は簡単であるが, 変換結果の簡略化に果たす役割は大きい.

以上列挙した発見的変換則は, いずれも他の変換則からは決して合成されることはない. それは, 適用ケースが異なることや証明の過程を見れば明らかである.

以下, 個々の発見的変換則を述べる.

[補助定理9] 除算(1)

$(tl: X(x): {}^{\forall}Y(y){}^{\exists}Z(z)(f(x, z)\wedge g_\sigma(y, z)))(I)$

$=(tl': (((x, z_\sigma): X(x), Z(z): f)[z_\sigma/y_\sigma]Y)$

$(w): (\ ))(I)$

ただし, $tl'$ は $tl$ 中の属性 $x[i]$ を対応する属性 $w[i]$ に置換したもの, また, $z_\sigma$ は $g$ に現れる属性 $z[j]$ の集合を表す. $y_\sigma$ も同様. さらに,

• $Y(I)$ は空でないこと, すなわち, $Y(I)\diamond\phi$

(証明) 準備として全称限量子の集合表現を確認する.

$(tl: X(x): {}^{\forall}Y(y)h(x, y))(I)$

$=\{x[TL]: x\in X(I)\wedge{}^{\forall}Y(I)(y)h(x, y)=1\}$

$=\{x[TL]: x\in X(I)\wedge$

$\{y': y'\in Y(I)\wedge h(x, y')=1\}$

$=\{y: y\in Y(I)\}\}$

$=\{x[TL]: x\in X(I)\wedge$

$\{y'_\wedge: y'\in Y(I)\wedge h(x, y')=1\}$

$=\{y_\wedge: y\in y(I)\}\}$

ここで証明に入る. 全称限量子の集合表現を用いて,

左辺$=\{x[TL]: x\in X(I)\wedge\{y'_\sigma: y'\in Y(I)\wedge$

${}^{\exists}Z(I)(z)(f(x, z)\wedge g_\sigma(y', z))=1\}$

$=\{y_\theta: y\in Y(I)\}\}$

存在限量子を展開して,

$=\{x[TL]: x\in X(I)\wedge\{y'_\theta: y'\in Y(I)\wedge z\in Z(I)\wedge f(x,z)=1\wedge g_\theta(y',z)=1\}$
$=\{y_\theta: y\in Y(I)\}\}$

$g_\theta(y',z)=1$ であるので $y'_\theta$ が $z_\theta$ に置換でき，さらに $y'\in Y(I)$ を除けば制約が弛み,

$=\{x[TL]: x\in X(I)\wedge\{z_\theta: z\in Z(I)\wedge f(x,z)=1\}\supseteq\{y_\theta: y\in Y(I)\}\}$
$=\{x[TL]: x\in X(I)\wedge\{z_\theta: z\in Z(I)\wedge x'\in X(I)\wedge x'=x\wedge f(x',z)=1\}\supseteq\{y_\theta: y\in Y(I)\}\}$

今，$R(I)=\{(x',z_\theta): x'\in X(I)\wedge z\in Z(I)\wedge f(x',z)=1\}$ とおくと,

$=\{x[TL]: x\in X(I)\wedge\{r[z_\theta]: r\in R(I)\wedge r[x']=x\}\supseteq\{y_\theta: y\in Y(I)\}\}$

$Y(I)$ が空ならば上式は $\{x[TL]: x\in X(I)\}$ となり条件式が全くないに等しい．今は空でないと仮定したので，$R(I)$ に属さない $x$ は捨てられる．すなわち，$X(I)$ を $R(I)$ に置換しても結果は同じで,

$=\{r'[TL]: r'\in R(I)\wedge\{r[z_\theta]: r\in R(I)\wedge r[x']=r'[x']\}\supseteq\{y_\theta: y\in Y(I)\}\}$
$=\{w[TL']: w\in\{r'[x']: r'\in R(I)\wedge\{r[z_\theta]: r\in R(I)\wedge r[x']=r'[x']\}\supseteq\{y_\theta: y\in Y(I)\}\}\}$

これは除算の集合表現に符号し，その後 $R$ を展開して,

$=\{w[TL']: w\in(R[z_\theta/y_\theta]Y)(I)\}=$右辺

[補助定理 10] 除算(2)

$(tl: X(x): {}^\forall Y(y)^\exists Z(z)(f_\epsilon(x,z)\wedge g_\epsilon(y,z)))(I)$
$=(tl': (X[\exists; F_\epsilon](Z[z_\phi, z_\theta][z_\theta/y_\theta]Y))(w): ( ))(I)$

ただし，$tl'$, $z_\phi$ 等は前補助定理と同様．さらに,

- $Y(I)$ は空でないこと，すなわち，$Y(I)\langle\rangle\phi$

(証明) 前補助定理と同様にして,

左辺$=\{x[TL]: x\in X(I)\wedge\{z_\theta: z\in Z(I)\wedge f_\epsilon(x,z)=1\}\supseteq\{y_\theta: y\in Y(I)\}\}$

今，$Y(I)$ は空でないので，所詮満足すべき $z'\in Z(I)\wedge f_\epsilon(x,z')=1$ を条件に加えても同じことになり,

$=\{x[TL]: x\in X(I)\wedge z'\in Z(I)\wedge f_\epsilon(x,z')=1\wedge\{z_\theta: z\in Z(I)\wedge z_\phi=z'_\phi\}\supseteq\{y_\theta: y\in Y(I)\}\}$

$Z(I)$ にて問題なのは，$z_\phi$, $z_\theta$ のみであるから，$S(I)=\{(z_\phi, z_\theta): z\in Z(I)\}$ とおいて $Z(I)$ を置換して,

$=\{x[TL]: x\in X(I)\wedge u\in\{z'_\phi: z'\in S(I)\wedge\{z_\theta: z\in S(I)\wedge z_\phi=z'_\phi\}\supseteq\{y_\theta: y\in Y(I)\}\}\wedge f_\epsilon(x,u)=1\}$

除算の集合表現に符合し，$S=Z[z_\phi, z_\theta]$ を代入して,

$=\{x[TL]: x\in X(I)\wedge u\in(Z[z_\phi, z_\theta][z_\theta/y_\theta]Y)(I)\wedge f_\epsilon(x,u)=1\}$
$=\{x[TL]: x\in X(I)\wedge\{u: u\in(Z[z_\phi, z_\theta][z_\theta/y_\theta]Y)(I)\wedge f_\epsilon(x,u)=1\}\langle\rangle\phi\}$

準結合の集合表現に符合し,

$=$右辺

[補助定理 11] 除算(3)

$(tl: X(x): {}^\forall Y(y)^\exists X[H_\epsilon](z)(f_\epsilon(x,z)\wedge g_\epsilon(y,z)))(I)$
$=(tl': (X[H_\epsilon][z_\phi, z_\theta][z_\theta/y_\theta]Y)(w): ( ))(I)$

ただし，$tl'$, $z_\phi$ 等は前補助定理と同様．さらに,

- $Y(I)$ は空でないこと，すなわち，$Y(I)\langle\rangle\phi$
- $x_\phi\supseteq tl$

(証明) 補助定理 10 を適用して,

左辺$=(tl': (X[\exists; F_\epsilon](X[H_\epsilon][z_\phi, z_\theta][z_\theta/y_\theta]Y))(w): ( ))(I)$
$=(tl: X(x), (X[H_\epsilon][z_\phi, z_\theta][z_\theta/y_\theta]Y)(y): f_\epsilon(x,y))(I)$

除算も組のふるい落とし（選択）にほかならない点に留意すると補助定理 12 が適用できて，結合が省略でき,

$=$右辺

[補助定理 12] 結合省略

$(tl: X(x), (X[H_\epsilon])(y), \Omega(\omega): f_\epsilon(x,y)\wedge g)(I)$
$=(tl': (X[H_\epsilon])(y), \Omega(\omega): g')(I)$

ただし，$tl'$, $g'$ は各々 $tl$, $g$ 中の属性 $x[i]$ を論理式 $f_\epsilon$ 中で等しいとした属性 $y[j]$ に置換したもの.

- $X[x_\phi](I)\supseteq Y[y_\phi](I)$
- $x_\phi\supseteq x_{\phi\phi}$
- $x_\phi\supseteq x_\theta$

(証明) 条件 $x_\phi\supseteq x_{\phi\phi}$, $x_\phi\supseteq x_\theta$ から，目標リストおよび条件式中の属性 $x[i]$ はすべて論理式 $f_\epsilon$ 中で属性 $y[j]$ に等しいとされている．すなわち,

$f_\epsilon=\cdots\wedge x[i]=y[j]\wedge\cdots$

また，このとき，$X[x_\phi](I)\supseteq Y[y_\phi](I)$ であるから，$y[j]$ のとる値は常に $x[i]$ のとる値に含まれる．したがって，範囲 $X$ の存在意義はなく，$x[i]$ を $y[j]$

で置換して論理式 $f_s$ を除去できる.

[補助定理 13] 射影省略

$((X[1],\cdots,X[m])\colon X(x)\colon (\ ))(I)=X(I)$

ただし, $m=X$ の次数

(証明) 自明なので省略.

[補助定理 14] 肯定準結合

$(tl\colon X(x),\ \Omega(\omega)\colon (^{\exists}Y(y)(f_j(x,\ y)\wedge g_s))\wedge h)(I)$
$=(tl\colon (X[\exists\ ;\ F_j](y_f\colon Y(y)\colon g_s))(x),$
$\Omega(\omega)\colon h)(I)$

ただし, $f_j$ は $f_s$ でもよい.

(証明)

左辺$=\{(x,\ \omega)[TL]\colon x\in X(I)\wedge\omega\in\Omega(I)\wedge$
$\{y\colon y\in Y(I)\wedge f_j(x,\ y)=1\wedge g_s(y)=1\}\langle\rangle\phi\wedge$
$h=1\}$
$=\{(x,\ \omega)[TL]\colon x\in X(I)\wedge\omega\in\Omega(I)\wedge\{u\colon u\in$
$\{y\colon y\in Y(I)\wedge g_s(y)=1\}\wedge f_j(x,\ u)$
$=1\}\langle\rangle\phi\wedge h=1\}$

準結合演算の定義を適用して,

$=\{(x,\ \omega)[TL]\colon x\in(X(I)[^{\exists}\ ;\ F_j]$
$\{y\colon y\in Y(I)\wedge g_s(y)=1\})\wedge\omega\in\Omega(I)\wedge h=1\}$
$=\{(x,\ \omega)[TL]\colon x\in(X[^{\exists}\ ;\ F_j](y_f\colon Y(y)\colon$
$g_s))(I)\wedge\omega\in\Omega(I)\wedge h=1\}$ =右辺

[補助定理 15] 否定準結合

$(tl\colon X(x),\ \Omega(\omega)\colon (\sim^{\exists}Y(y)(f_j(x,\ y)\wedge g_s))\wedge h)(I)$
$=(tl\colon (X[\sim\exists\ ;\ F_j](y_f\colon Y(y)\colon g_s))(w),\ \Omega(\omega)\colon$
$h)(I)$

ただし, $f_j$ は $f_s$ でもよい.

(証明) 前補助定理で肯定を否定に置き換えればよい.

## 4. 最適性の評価

### 4.1 変換プログラムによる変換例

前記変換法が正しく動作するか, さらに狙いどおり最適な関係代数表現を生成するかを検証するため, 変換プログラムを作成し, 約 100 件の検索例を試験した. 変換プログラムは, 構文解析, 記号変換に向く prolog で作成した. 実用的な検索例は各種文献を利用し, 試験のための検索例は筆者が作為的に作成した.

以下に, 文献[12]の例題の一部の変換結果を示す.

スキーマ: emp (氏名, 給料, 管理者, 店),
sales (店, 商品, 数量), supply (会社, 店, 商品, 数量), class (商品, 型), loc (店, 階)

(例 1) おもちゃ屋で販売している銃の数を求めよ.

★ (関係論理) $(u[3])\colon \mathrm{sales}(u)\colon u[1]=\mathrm{toy}\wedge$
$u[2]=\mathrm{gun}$

☆ (関係代数) sales[(#1=toy)∧(#2=gun)][#3]

(例 2) 2階の店で販売している商品は何か.

★ $(u[2])\colon \mathrm{sales}(u)\colon {}^{\exists}\mathrm{loc}(v)(v[1]=u[1]\wedge v[2]=2)$

☆ (sales[∃ ; #1=#1](loc[#2=2]))[#2]

(例 3) 上司より給料が多いのは誰か.

★ $(u[1]\colon \mathrm{emp}(u)\colon {}^{\exists}\mathrm{emp}(v)(v[1]=u[3]$
$\wedge v[2]<u[2])$

☆ (emp[∃ ; #3=#1, #2>#2] emp)[#1]

(例 4) 靴屋の誰よりも給料が多いのは誰か.

★ $(u[1]\colon \mathrm{emp}(u)\colon {}^{\forall}\mathrm{emp}(v)\ (v[4]\langle\rangle\mathrm{shoe}\vee$
$v[2]<u[2])$

☆ (emp[～∃ ; #2<=#2](emp[#4=shoe]))[#1]

(例 5) 全商品を出荷している会社を求めよ.

★ $(u[1])\colon \mathrm{supply}(u)\colon {}^{\forall}\mathrm{class}(v){}^{\exists}\mathrm{supply}(w)$
$(w[1]=u[1]\wedge w[3]=v[1])$

☆ (supply[#1, #3])[(#2)/(#1)] class

(例 6) a 型の全商品を 2 階の店に出荷する会社を求めよ.

★ $(u[1])\colon \mathrm{supply}(u)\colon {}^{\forall}((t)\colon \mathrm{class}(t)\colon t[2]=\mathrm{a})(v)$
${}^{\exists}((s)\colon \mathrm{supply}(s)\colon {}^{\exists}\mathrm{loc}(r)(r[1]=s[2]\wedge r[2]=2))$
$(w)(w[1]=u[1]\wedge w[3]=v[1])$

☆ ((supply[∃ ; #2=#1](loc[#2=2]))[#1, #3])
[(#2)/(#1)](class[#2=a])

(例 7) 全商品を売る店がある階を求めよ.

★ $(u[2])\colon \mathrm{loc}(u)\colon {}^{\forall}\mathrm{class}(v){}^{\exists}\mathrm{sales}(w)$
$(w[1]=u[1]\wedge w[2]=v[1])$

☆ (loc[∃ ; #1=#1]((sales[#1, #2])
[(#2)/(#1)] class))[#2]

(例 8) すべての店が a 型の商品を販売する階を求めよ.

★ $(u[2])\colon \mathrm{loc}(u)\colon {}^{\forall}\mathrm{loc}(v)(v[2]\langle\rangle u[2]\vee{}^{\exists}\mathrm{sales}(w)$
$(w[1]=v[1]\wedge{}^{\exists}\mathrm{class}(x)(x[1]=w[2]$
$\wedge x[2]=\mathrm{a})))$

☆ (loc[～∃ ; #2=#2](loc[～∃ ; #1=#1]
(sales[∃ ; #2=#1](class[#2=a]))))[#2]

### 4.2 最適性の達成度評価

まず, 従来変換法がいかに冗長な変換結果を生成するか, 実例を通して見てみる. まず, Codd の方法[6]を例 4 に適用すると, 以下となる.

((((emp[ ] emp)[#2>#6])[+]((emp[ ] emp)
[#4〈〉shoe]))[(#5, #6, #7, #8)/(#1, #2,
#3, #4)] emp)[#1]

負荷の重い演算が 5 回, 軽いのが 2 回である. 本変換

法では，各々，1回と2回である．

Klug の方法[8] は次のような極めて簡単な例に対しても長大な結果を生成する．

(例) John の管理者は誰か．

関係論理表現 $(u[3])$ : emp $(u)$ : $u[1]$=John

Klug の結果 ((emp[#3][ ](emp[#1][#1=#1]
{John}))[ ]emp)[#1=#6, #2=#4]
[#1]

本変換法の結果 emp[#1=John][#3]

負荷の重い演算，軽い演算，各1回ずつで済むものが，各々6回，1回も必要になる．

Ullman の方法は例を挙げないが，組関係論理表現を定義域関係論理表現に変形してから関係代数表現に変換するため，限量子が多数生成され極めて効率の悪いものとなる．また，Kim の方法は SQL を正準形に変換するため，アイデアの比較はできても，例による比較はできない．

そこで，本変換法について考察する．前節の例題は高度に複雑な内容のものを含んでいるにもかかわらず，驚くべきことに，本変換法の変換結果と人間（筆者）の作成するものとは全く同じであった．また，その他の例題も構文的には類似しており，ほとんどの場合に最適な結果が得られた．一連の実験から，本論文で提案した変換体系は変換結果の最適性に関して満足できるものであることが確認できた．

なお，現実には，種々の理由から，冗長で洗練されない関係論理表現が入力される可能性はある．そのとき，本変換法は最適とはいえない関係代数表現を生成する可能性がある．どのように冗長で洗練されない関係論理表現を入力しても常に最適な関係代数表現を生成するには，本論文で示した変換則より更に多くの変換則を用意することが必要である．

## 5. むすび

関係モデル検索系の中でユーザインタフェースに優れた関係論理に基づく表現を最適な関係代数表現に変換する体系を変換則を中心に述べた．新変換法は関係論理表現を再帰構造に沿って逐次的に変換する中で，基本変換則に加え発見的変換則を優先的に適用することにより関係代数レベルの最適化を行う．変換プログラムを作成し，約100件の検索例を試験して，所期の結果を得た．今後は，集約関数，四則，非正規形等が扱えるように本変換法を拡張する研究が必要である．

**謝辞** 本研究に際し，ご指導頂いた NTT 電気通信研究所の橋本昭洋部長（基礎研究所情報科学研究部），村上国男部長（情報通信処理研究所知識処理研究部），吉田清グループリーダ，ならびに有益なコメントを頂いた査読委員の方々に感謝いたします．

## 参考文献

1) Brodie, M. and Schmidt, J.: Final Report of the ANSI/X3/SPARC DBS-SG Relational Database Task Group, SIGMOD Record, Vol. 12, No. 4, pp. 1-62 (1982)
2) ISO/TC 97/SC 21: Database Language SQL, ISO/TC 97/SC 21/WG 5-15 (1985).
3) Held, G. D. et al.: INGRES—A Relational Database System, *Proc. of NCC*, pp. 409-416 (1975).
4) Luo, D. et al.: Data Language Requirements of Database Machines, *Proc. of NCC*, pp. 617-626 (1982).
5) Selinger, P. G. et al.: Access Path Selection in a Relational Database Management System, *Proc. of Int. Conf. Manage. of Data*, pp. 23-34 (1979).
6) Codd, E. F.: Relational Completeness of Database Sublanguages, in *Database Systems, Courant Comput. Sympo. 6*, pp. 65-98, Prentice-Hall, Englewood Cliffs, N. J. (1972).
7) Ullman, J. D.: *Principles of Database Systems*, Second Ed. Computer Science Press, Rockville, Maryland (1982).
8) Klug, A.: Equivalence of Relational Algebra and Relational Calculus Query Languages Having Aggregate Functions, *J. ACM*, Vol. 29, No. 3, pp. 699-717 (1982).
9) 中野良平，斉藤和巳：ルールに基づいた関係論理／関係代数変換法，情報処理学会データベースシステム研究会資料，86-DB-54 (1986).
10) Kim, W.: On Optimizing an SQL-like Nested Query, *ACM Trans. Database Syst.*, Vol. 7, No. 3, pp. 443-469 (1982).
11) Chang, C. L. and Lee, R. C. T.: *Symbolic Logic and Mechanical Theorem Proving*, Academic Press, London (1973).
12) Chang, C. L.: DEDUCE2: Further Investigation of Deduction in Relational Data Bases, in Gallaire, H. and Minker, J. (eds.), *Logic and Data Bases*, pp. 201-236, Plenum Press, New York (1978).

## 付録

[安全性の定理] 関係論理で表現した任意のアルファは安全である．

(証明) 関係論理で表現した任意のアルファは，$t_i$

を項，$R_j$ を範囲，$u_j$ を組変数，$f$ を条件式とするとき，

$$(t_1,\cdots,t_n):\ R_1(u_1),\cdots,R_m(u_m):\ f$$

で表せた．これは，次のような Ullman の組関係論理に帰着できる．

$$\{s\mid {}^{\exists}u_1,\cdots,{}^{\exists}u_m(s[1]=t_1\wedge\cdots\wedge s[n]=t_n \wedge R_1(u_1)\wedge\cdots\wedge R_m(u_m)\wedge f\} \qquad (1)$$

$R_j$ の中にアルファがある場合にも，この変形を繰り返し行うことにより，最終的には上記の形に帰着できる．

一方，Ullman の組関係論理 $\{s\mid\psi(s)\}$ が安全であるための十分条件は，

(イ) $\psi(s)=1$ ならば，$s$ の各成分は $\mathrm{Dom}(\psi)$ の要素であること，

(ロ) $\psi$ 中の任意の $({}^{\exists}x)(\xi(x))$ に対し，$\xi$ 中の自由変数の任意の値に対し $\xi(x)=1$ ならば，$x$ の各成分は $\mathrm{Dom}(\psi)$ の要素であること，

(ハ) $\psi$ 中の任意の $({}^{\forall}x)(\xi(x))$ に対し，$x$ のどの成分も $\mathrm{Dom}(\psi)$ に属さないならば，$\xi$ 中の自由変数の任意の値に対し $\xi(x)=1$ であること．

ただし，$\mathrm{Dom}(\psi)$ は $\psi$ 中に現れる値と関係を構成する値の和集合である．

(1)式には $s$ と $u_1,\cdots,u_m$ の関係が陽に規定されていないが，$t_1,\cdots,t_n$ は Klug の関係論理における目標リストで，$u_1,\cdots,u_m$ は範囲規定で規定された組変数という関係があるので，範囲である関係の積和の中から解を求めることになり，(イ)を満足する．また，(1)式の $f$ 中の限量子はすべて範囲付きであるので，

$${}^{\exists}R(x)\eta={}^{\exists}x(R(x)\wedge\eta),$$

$${}^{\forall}R(x)\eta={}^{\forall}x(\sim R(x)\vee\eta)$$

のいずれかである．これらは，各々(ロ)，(ハ)を満足する．したがって，(1)式は安全であり，元のアルファも安全となる．　(証明終)

(昭和62年1月 5 日受付)
(昭和63年2月10日採録)

**中野　良平**（正会員）

昭和22年生．昭和46年東京大学工学部計数工学科卒業．同年日本電信電話公社（現 NTT）入社．以来，統計解析，分散処理，データベース，知識処理の研究に従事．現在 NTT 情報通信処理研究所グループリーダ．人工知能学会，ACM，IEEE 各会員．

**斉藤　和巳**（正会員）

昭和38年生．昭和60年慶応義塾大学理工学部数理科学科卒業．同年日本電信電話(株)入社．現在，同情報通信処理研究所にて知識処理の研究に従事．人工知能学会会員．