

## 分散知識ベース管理システムPHI

羽生田 博美\*, 山下 祥司\*, 宮崎 収兄\*, 森田 幸伯+, 伊藤英則+

\*: 沖電気工業株式会社, +: 財団法人 新世代コンピュータ技術開発機構

第5世代コンピュータ研究開発プロジェクトでは、知識ベースに関する研究の一環として分散知識ベース管理システムPHIの研究が行われている。PHIは、ICOT-LANで結合した複数台の逐次型推論マシンPSI上で稼動する実験システムであり、PHIを構成する各分散サイトは、PSIと知識ベース演算エンジンを結合したハードウェア上に実現される。知識ベース演算エンジンは、重ね合わせ符号索引を用いて、関係代数演算やルールに対する処理を高速に実行する専用ハードウェアである。

PHIは、分散演繹データベース管理を核として実現される。PHIにおける演繹推論を効率的に行うため、ホーン節形式の問い合わせは、関係する内包データベース中のルールと統合され関係データベース演算にコンパイルされる。

本稿では、PHIの問い合わせ処理方式および知識ベース演算エンジンの処理方式について述べる。

## PHI-A Distributed Knowledge Base Management System

Hiromi HANIUDA\*, Shouji YAMASHITA\*, Nobuyoshi MIYAZAKI\*

Yukihiro MORITA+, Hidenori ITOH+

\*: Oki Electric Industry Co., Ltd.,

4-11-22 Shibaura, Minato-ku, Tokyo 108, Japan.

+: Institute for New Generation Computer Technology,

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

A distributed knowledge base management system PHI is being developed as part of the knowledge base research in the Fifth Generation Computer Systems project. PHI is an experimental system which consists of a number of nodes linked by the ICOT-LAN. Each node of PHI is a personal sequential inference machine PSI with an attached knowledge base engine (KBE). KBE is being designed based on a superimposed code scheme to process relational operations efficiently.

The kernel of PHI is a distributed deductive database system. Queries expressed in the form of Horn clause are combined with related rules and compiled to relational operations to realize the efficient processing of deductive inferences in PHI.

This paper outlines the model of the query processing strategy of PHI and the superimposed code scheme of KBE for compiled relational operations.

## 1. まえがき

大規模共有知識ベース管理技術の確立は、知識情報処理システムを実現する上での重要な技術課題の一つである。知識ベース管理技術の研究では、大規模知識ベースの管理技術や処理の高速化を図る並列型アーキテクチャの確立が重要な課題であると共に、最近の分散型システムの発展に対応するために分散環境下における知識ベース管理技術の確立も重要な課題の一つである。

筆者らは第5世代コンピュータ・システム(FGCS)プロジェクトの一環として分散知識ベースの実験システムPHI(Predicate logic based hierarchical knowledge base management system)を開発中である。FGCSプロジェクトでは、2つの知識ベース・モデル、すなわち結合モデルと統合モデルを研究している。PHIでは、知識ベース・モデルとして結合モデルを採用しており、基本的には分散演繹データベース管理システムを核としている。アプリケーション・プログラムからPHIへはホーン節形式の問い合わせが与えられる。ホーン節問い合わせは、関連する内包データベース(IDB: Intensional Database)中のルールと統合され、外延データベース(EDB: Extensional Database)に対する関係代数演算列にコンパイルされる。これにより、知識ベース管理に際してデータベース技術の有効利用を可能にしている。

また、PHIでは大規模知識ベースに対する処理を高速化するための専用ハードウェアとして知識ベース演算エンジン(KBE)を付加している。KBEは、重ね合わせ符号を用いることにより関係代数演算やルールに対する処理を高速に処理する。

本稿では、PHIにおける演繹データベースの処理方式について述べる。第2章でPHIの概要について、第3章でその問い合わせ処理方式について述べる。第4章では、KBEの処理方式について述べる。

## 2. システム概要

PHIは、分散環境下における知識ベース管理技術を確立するため、

- ・ 複数の知識ベース・マシンによる分散知識ベース制御方式、
- ・ 推論マシンの付加装置としてのKBE

の実験を行うシステムである。物理的には、PHIシステム全体は図1に示すように、逐次型推論マシン(PSI)と一斉同報・グループ通信機能やコミットメント通信機能を持つ狭域ネットワーク(ICOT-LAN)によって構成される(以下、分散知識ベース管理システム(KBMS)全体をPHI、PHIを構成する各サイトをphiと表記する)。各phiは図2に示すように、PSIと知識ベース演算エンジンを結合したハードウェア上に実現される。

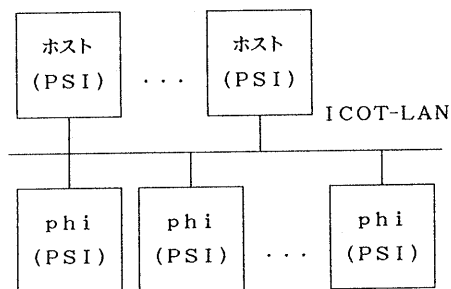


図1 PHIシステムの物理構成

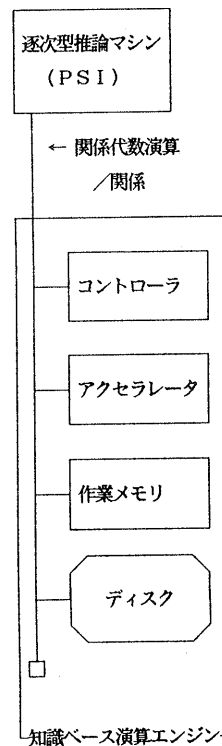


図2 知識ベース演算エンジン

PHIは、論理的には図3に示すようにグローバル知識ベース・マネージャ (GKBM) とローカル知識ベース・マネージャ (LKBM) から構成される。GKBMは、KBMSにアクセスするアプリケーション・プログラムに対してコーディネータとして動的に割り当てられる。各LKBMは自分に関連する知識ベースの管理を行い、他のLKBMと協調してアプリケーションからの問い合わせを処理する。この構成は、[Miyazaki82]で提案されたモデルを分散化、拡張したモデルである。先に提案したモデルは次のような特徴を持っていた。

- (1) 関係データベース管理システム (RDBMS) によりEDBを管理する。大規模知識ベースに対して集合指向の関係演算を使うことにより、全体的な性能を向上させることが期待できる。
- (2) 再帰問い合わせに対する拡張を施したRDBMSと論理プログラム間のインタフェースはホーン節である。

第一の特徴は、演繹推論と関係データベースとの関係に対するFGCSプロジェクトでの基本的考え方を反映したものであり、第二の特徴が先の提案の中心であった。PHIのモデルは、先のモデルに対して次のような拡張を行ったモデルである。

- (3) IDBの管理を強化するために、RDBMSに対する拡張部をRDBMSと分離し知識管理層とする。
- (4) 分散環境に適応させるために分散制御部を付加する。

したがって、PHIの核は分散演繹データベースであり、PHIは利用者に対して知識管理層とデータベース管理層から構成されるKBMSとして機能する。データベース管理層は、機能的には関係データベース管理システムと同等であり、Prologでのファクト集合に対応するEDBの管理を行う。知識管理層は、Prologのルール集合に対応するIDBの管理とそれをを用いた演繹推論機能を持つ。分散制御部は、分散データベース機能を核とし、グループ通信機能等を持つICOT-LANの特徴を生かして、問い合わせ処理等の動的最適化を行う方式を採用している [吉田86]。

### 3. 問い合わせ処理方式

PHIでは、IDBの大きさはEDBの大きさに比較して小さいと仮定している。このような環境では、演繹推論を関係データベース演算により実現できるという事実は魅力的である。すなわち、PHIの分散知識ベース処理を分散演繹データベース処理を基本として実現することにより、演繹データベースが持っている演繹推論機能により論理型で表現された知識の処理が可能になり、データベース技術の利用により大容量の共有知識の処理・管理が可能になる。

構造を含まないホーン節形式の問い合わせは、再帰を含まなければ容易に関係問い合わせにコンパイルできる。したがって、問い合わせ処理の中心的問題は再帰処理である。多くの再帰問い合わせ処理戦略が提案されているが、これらの戦略はその特徴、すなわちインタプリタとコンパイル、トップ・ダウンとボトム・アップにより分類することができる [Bancilhon86]。コンパイ

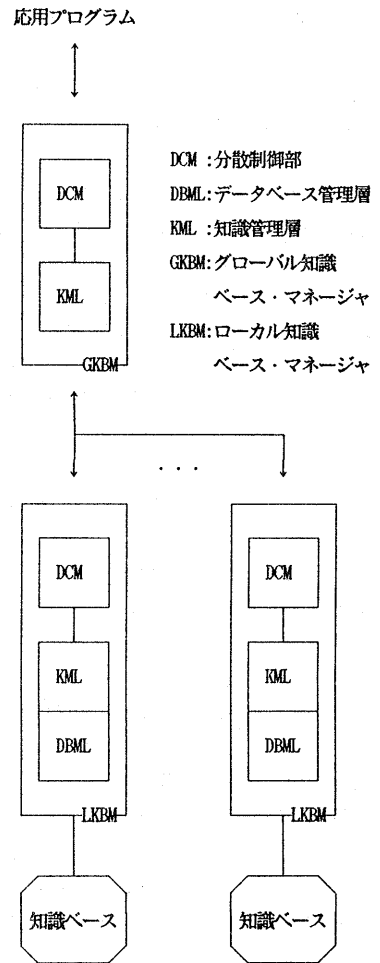


図3 PHIの論理システム構成

ル法はデータベース管理における性能向上技法を有効に利用することができる。したがって、PHIの問い合わせ処理戦略はコンパイル法とボトム・アップに基づいている[Miyazaki86]。

以下本章では、PHIの問い合わせ処理の概要と問い合わせ処理の最適化について述べる。

### 3. 1 問い合わせ処理の概要

ホストから PHI に対しては、ホーン節形式の問い合わせが出される[Miyazaki86, 宮崎86]。PHIは、この問い合わせをコンパイル法[Gallaire84]に基づいて処理する。コンパイル法は、演繹等のIDBに対する処理をEDB に対する検索戦略の決定・実行に先立って行うことを特徴としている。また、再帰的問い合わせは、最小不動点(LFP)演算(関係代数演算の繰り返し処理によりLFPを求める演算)により処理する[Miyazaki86]。したがって、PHIの問い合わせ処理は、(a) IDBに関する処理、(b) LFP戦略の決定とEDBに関する処理、に分けることができる。(a)はシステム全体に関係した処理であり、(b)はEDB(各サイト)に密着した処理である。PHIでは、分散資源を有効に利用するために、これらの処理をGKBMとLKBMとに分けて処理する。

以下、GKBMとLKBMにおけるPHIの問い合わせ処理方式について述べる。

#### 3. 1. 1 グローバル知識ベース・マネージャ

GKBMは、利用者からの問い合わせの受け付け、結果のとりまとめ、IDBの範囲での問い合わせの最適化、等のシステム全体に関係した処理を行う。以下、GKBMが行う処理について述べる。

##### (1) ホーン節変換

利用者からの問い合わせとIDBとを統合したホーン節(このホーン節集合はホストからの問い合わせとみなすことができるので、HCQ: Horn Clause Queryと呼ぶ)に対して、ホーン節変換(HCT)[Miyazaki86]を行う。HCTは、部分評価法を応用し、式の代入と類似の処理により問い合わせの簡単化を行う。演繹データベース処理の効率性は、問い合わせの複雑さに依存するので、HCTにより処理を効率化することができる。

##### (2) 問い合わせの部分問い合わせへの分解

HCQ 中のルールではヘッド述語の評価はボディ部の述語の評価結果に依存していると考えることができ、HCQはこの依存関係に基づいて有向グラフとして表現できる。PHIでは、このグラフの強連結成分分解に基づいて、HCQを部分問い合わせへ分解する。分解後、各成分はEDBに対する処理またはLFP演算に関して1つのまとまった処理単位になっており、各成分間の依存関係も半順序的である。したがって、基本的には個々の成分をゴールから最も遠い側から順に評価すれば解を得ることができる。

この分解により、以後のHCQに対する処理を各成分毎に独立に行うことを可能としている。また、HCT後のHCQに対して強連結成分分解を行うと、各成分は再帰述語数、再帰の線形性、他成分への依存性により幾つかの限られたパターンに分類できる。

##### (3) 内部問い合わせの生成

HCQ全体に対して制約条件の伝播に基づいた最適化を、また各成分に対して上述した各強連結成分のパターンに基づいた最適化を行う。この最適化により、HCQ を限定された空間で探索を行う形式に変換する。次に、EDBに格納されている関係に対応する述語に対して、その関係の格納サイト情報を付加し、関連LKBMに対する問い合わせ(内部問い合わせ)とする。

#### 3. 1. 2 ローカル知識ベース・マネージャ

LKBMは、各サイトでそのサイトに密着した処理を、必要に応じて他のLKBMと協調しながら、自律的に行う。以下、LKBM

が行う処理について述べる。

### (1) ローカル・プランの生成

各LKB M毎にデータ・アクセス・プラン(ローカル・プランと呼ぶ)を生成する。まず、内部問い合わせの内、自分が関連する成分に対してLFP計算の戦略を決定し、次に各成分に対して関係代数演算列を生成する。ホーン節形式では、ルールの配列順序や各ルールのボディ中における述語の順序に意味はないが、これらの順序は関係代数演算列では二項演算の実行順序に対応する。関係データベースでは、二項演算の実行順序が処理効率に大きな影響を与えるので、これら実行順序の最適化も合わせて行う。

### (2) ローカル・プランの実行

(1)で生成されるプランは、依存グラフの個々の強連結成分に対応する部分問い合わせの集合であり、上述したように半順序関係がある。各LKB Mはこの順序にしたがって部分問い合わせを処理する。繰り返しを必要とする部分問い合わせに対しては、その問い合わせに関連するLKB M 同士が協調して処理実行サイトを動的に決定する[吉田86]。

## 3. 2 問い合わせ処理の最適化

演繹問い合わせ処理方式には、大きく分けて、EDBに存在する関係側から順次部分解を組み立てて行き最終解を得るボトム・アップ方式と、逆に問い合わせ側からサブゴールを展開しながら最終解を求めるトップ・ダウン方式がある[Bancilhon86]。トップ・ダウン型の処理は、問い合わせやIDB中で与えられた制約条件を早い段階で利用するための制約伝播に適している。また、ボトム・アップ型の処理は、従来のデータベース技術で確立されている効率化処理方式の利用に適している。

一方、ボトム・アップ方式では、単純な問い合わせを効率的に処理することは容易であるが、複雑な問い合わせではなんらかの制約条件(束縛条件)の伝播機構を用いる必要がある。また、PHIのような分散型システムでは、トップ・ダウン型の方式は、不必要なサイト間インタラクションの増加により処理コストが増大したり、問い合わせの効率化が複雑で困難になる可能性がある。

このように、両方式には異なった特徴がある。したがって、PHIではボトム・アップ型を基本とし、制約条件の伝播にはトップ・ダウン型の方式を併用することにより、両方式の特長を生かして問い合わせの内容に応じた最適な処理を行うことを可能とする。すなわち、

- (1) 問い合わせ全体に対しては、ボトム・アップ型とトップ・ダウン型の方式を併用し、各部分問い合わせを通じて制約条件の伝播を行うことにより問い合わせ全体を最適化し、
- (2) 制約条件を伝播された各部分問い合わせに対しては、差分展開方式[Aho79, Ceri86, Miyazaki86]、制約付最小不動点[宮崎87](RLFP)方式等を各部分問い合わせの構造に基づいて選択することにより、各部分問い合わせを最適化する。

以下、この最適化方式を図4のIDBルール中のproperty に対する問い合わせを例に説明する(ここで、edbProperty とedbIsAに対応する関係がEDBに存在すると仮定する。

図4のpropertyに対する問い合わせは、第3章で述べた部分問い合わせへの分解により、isA とproperty自身に対する2つの部分問い合わせに分解され

る。Objectに関する制約条件(例えば、Object=penguin)は単純に isAまで伝播させ、Propertyに対する制約条件(例えば、Property=canFly)は Classの取りうる値の集合 {Class|property(Class, canFly)} を制約条件として is Aまで伝播させる。

このように制約条件を伝播させることで、問い合わせ ?-property(penguin, Property) は、?-isA(penguin, Class) を解き、次に、?-property(penguin, Property) を解くという完全にボトム・アップ的動きになる。また、問い合わせ?-property(Object, can

```
property(Object, Property) :-
    edbProperty(Object, Property).
property(Object, Property) :-
    isA(Object, Class),
    property(Class, Property).
isA(Object, Class) :-
    edbIsA(Object, Class).
isA(Object, Class) :-
    edbIsA(Object, Subclass),
    isA(Subclass, Class).
```

図4 内包データベースのルールの例

Fly)は、property(Object, canFly) の部分解を求め、次に ?-isA(Object, Class), Class ∈ {Class|property(Class, canFly)} を求めるプロセスを繰り返すトップ・ダウン的動きとなる。また、propertyが複数の部分問い合わせに依存する場合にも、propertyが依存する部分問い合わせを、順次トップ・ダウン的に解くことで制約条件を伝播させる。ここで、制約条件を伝播された部分問い合わせ ?-isA(penguin, Class) 自身は、部分問い合わせに対する最適化として、差分展開方式を適用して効率的に処理する。

このような最適化を行うことにより、PHIでの最適化の問題は、すべての問い合わせパターンに対して処理戦略決定の問題として対処することができる。

#### 4. 知識ベース演算エンジン

PHIの問い合わせ処理は2つの段階、すなわちIDBからの問い合わせに関係するルールの抽出処理と問い合わせのコンパイル、解集合を求めるためのEDBに対する関係演算の実行、から構成される。重ね合わせ符号を使用することにより、IDBとEDBから構成される演繹データベースに対する処理を高速化できると考えられる。したがって、KBEでは重ね合わせ符号を用いた方法を演繹データベース処理を実現する一つの方法として研究している。

コンパイルされた問い合わせ中で頻繁に出現する関係演算は選択、結合、集合演算と集合比較演算である。集合演算と集合比較演算の頻出は、演繹データベースと関係データベースとの主な相違であり、これらの演算に重ね合わせ符号を用いることにより、全体的な処理性能を向上させることができる。PHIでは、IDBに比較してEDBが巨大であり、大容量のファクトを関係として格納することを想定しているためこれらの関係演算はIDBに対する処理に比較して性能的により決定的な処理である。

phiでは、これらの処理への対応として重ね合わせ符号語(scw)を用いたKBEを採用している。scwを用いた索引(scw索引)は大量のテキスト・ファイルの検索のために提案された方式であるが、

- (1) 部分照合検索に適している：すなわち複数の属性に対する条件を指定した検索に適している。
- (2) 索引サイズが重ね合わせる属性数に依存しない：したがって複数属性に対する索引が必要な時、全体としての索引のサイズを小さくできる。

等の特長があるため、知識ベースの検索にも適していると考えられる[Wada87]。

KBEでは、各関係に対して scw索引を与え、この索引を用いて処理対象を絞り込むことで高速な演算の実行を行う。索引は索引レコードの集合であり、各索引レコードを関係の組と1対1に対応させるため、対応する組のキー属性値から構成される索引値と呼ぶ scwと対応する組の識別子から索引レコードを構成する。KBEは図2に示したように、scw に対する処理を行う専用ハードウェアのアクセラレータ、コントローラおよび作業メモリから構成される。

以下本章では、KBEにおけるscw索引の生成方式および scw索引を用いた関係代数演算等の処理方式(scw方式) について述べる。

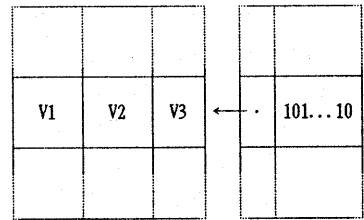
##### 4. 1 scw索引の生成

大量テキスト・データの検索手法として、文献[Roberts79, 有川82, 有川83] 等で scwを用いた検索方式が提案されている。これらの方式では、検索対象のテキスト・レコードが複数個のキーワードを含むようなレコード・ファイルに対してscw を用いた部分照合により高速な検索を実現している。

KBEでは、scwを関係に対する検索に応用するため、次のように索引を生成する。すなわち、N個の組を持つ関係 $r=(t_1, \dots, t_N)$  に対して索引を生成する場合を考える。この時、 $r$  のスキーマには $Kr$ 個のキー属性と、phiが提供する属性の型に応じて属性値を二進符号語(bcw)に写像するハッシュ関数  $h$  が与えられていると仮定する。ある組  $t_i$  に対する索引値は、次の手順で生成する。

- (1) tiの各キー属性値を hによりbcwに写像する。
- (2) tiのすべてのbcwを重ね合わせて(ビット毎にor演算を施して), scwを生成する。

このscwが索引値となる。rのすべての組に対して索引値を生成し、対応する組への識別子と組み合わせることで索引を作成する。図5にscw索引の生成例を示す。



関係

scw索引

V1 -ハッシュ→ 010...10  
 V2 -ハッシュ→ 010...00  
 V3 -ハッシュ→ 100...00  
 ————— or演算  
 101...10

図5 scw索引の生成

#### 4.2 scw方式による関係の検索

問い合わせqに対する検索では、 $Kq (\leq Kr)$ 個のキー属性値に対して検索条件を与え、この条件を満足するすべての組を抽出する。KBEでは、この処理を、索引を用いて処理の対象となるタプルを絞り込む段階と、絞り込まれた組に対する処理の二段階に分けて行う。

まず第一段階では、検索条件として指定された各キー属性値をハッシュしてbcwを生成し、これらを重ね合わせてscwを生成する(問い合わせマスクQと呼ぶ)。この時Qと索引値Siとの間で式(\*)を満足しなければ、その索引レコードに対応する組は問い合わせを満たさない。

$$Q \text{ and } Si = Q \dots\dots\dots (*)$$

したがって、問い合わせマスクQと索引レコードの値とを比較して、索引から式(\*)を満たすSiの集合Σを抽出する。この集合Σは、検索条件を満足していない組に対応する索引レコードも含むので、Σからは問い合わせを満足する候補の組集合を得ることができる。

第二段階では、Σに対応する候補組について、検索条件を満足するかどうか実際に個々の組を検査する。候補組の内、検索条件を本当に満足する組集合が、問い合わせの解である。

先に述べたように、関係全体は大容量であるため、このΣに対応する候補組は、二次記憶から読み上げる必要がある。したがって、実際には問い合わせを満足しないか問い合わせマスクは満足するような組数の期待値をFdとすると、Fdの値は直接二次記憶へのアクセス・コストに影響を与える。二次記憶アクセスは、全体の処理コストに対して大きな比重を占めるので、scw法ではFdの値をできるだけ小さく抑えることが重要である。

scwを用いた検索では、scw中でセットされているビットとリセットされているビットの数が等しくなるように索引レコードを構成すると、Fdの値を最小とすることができる[Roberts79]。キーワードを用いてテキスト・レコードを検索するシステムでは、各テキスト・レコード中のキーワード数はレコード毎に異なるため、重ね合わすbcwの個数が変化して、Fdの値を制御することが難しい。

一方、KBEの方式では、関係に対してscw索引を生成する場合、テキスト・レコードでのキーワードは関係でのキー属性に相当し、キー属性数は関係のスキーマによって決定される。関係スキーマは、静的であるため、重ね合わすbcwの個数は各組で常に一定である。したがって、ハッシュ関数を適当に設計することにより、索引値の分布を制御することが可能であり、先のFdの値を小さく抑えることが容易にできる。このような点からも、scw方式は関係の検索に適している。また、scw法は、検索条件で指定するキー属性数が多くなるほどFdの値を小さく抑えることができる。一方、大容量の関係に対する検索を高速化する一般的な方法としてB+木やハッシングがあるが、これらの方法は通常、単一の属性に対する検索条件しか扱えないために、Fd値を小さく抑えることが難しい。したがって、KBEは特に複数属性に検索条件が適用されるような検索処理の高速化を実現することが可能である[Wada87]。

#### 4.3 集合演算と包含関係の検査

先に述べたように、演繹データベース管理では、再帰問い合わせ処理が重要である。再帰問い合わせ処理の基本となるLFP演算は、和集合演算を含む関係代数演算を繰り返して実行し、繰り返し終了の判定を包含関係の検査や関係の同一性の検査により行う演算である[Bancilhon86, Miyazaki86]。したがって、PHIでは、通常の関係データベース管理システムに比べて、和集合演算や包含関係の検査等の演算が多くなる。一般にこれらの演算は処理負荷が大きいので、PHIではこれらの演算を、KBEを用いて高速に処理する。和集合、共通集合、差集合等の集合演算や関係間での包含関係の検査では、2関係間での同一組の検出が主な処理である。KBEでは、これらの処理を検索の場合と同様に二

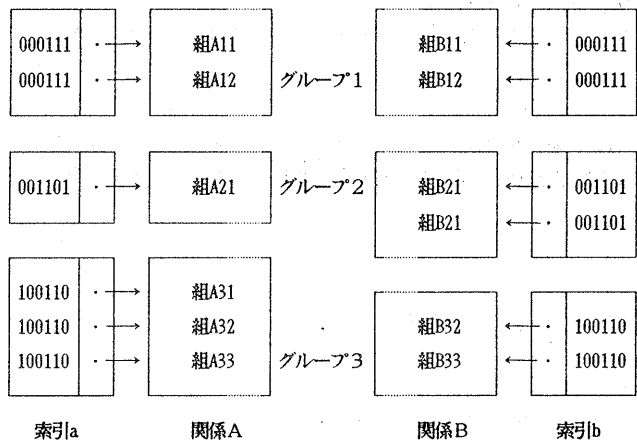


図6 索引値によるグループ対の生成

段階で行う。このうちの第一段階は集合演算と包含関係の検査の両方に共通である。ここでは索引を用いて、索引値が同一の組どうしは同一のグループに属するように、演算対象の各関係内の組をグループ分けする。各関係に対して同一のハッシュ関数を用いて索引を生成することにより、各関係内で同一値を持つ組は、それぞれ対応する索引値が一致することが保証され、2関係間で同一の組は必ず同一のグループ対に含まれる。したがって、第二段階では、二つの関係のグループ間で同一の索引値を有するグループの対をつくり、このグループ対の間で組を比較し、組の同一性を判断することにより目的の演算結果を得る。関係Aと関係Bをそれぞれグループ分けし、2関係間でグループ対を生成した例を図6に示す。

### 5. あとがき

分散知識ベース管理システムの実験システム PHIの概要、処理方式および知識ベース演算エンジンについて報告した。今後は、本稿で述べた方式の試作・実験を行うとともに、方式の改良検討を行っていく予定である。

### <<参考文献>>

[Aho79]Aho, A. V., Ullman, J. D., "Universality of Data Retrieval Languages," 6th POPL, pp.110-120, 1979.  
 [Bancilhon86]Bancilhon, F., Ramakrishnan, R. "An Amateur's Introduction to Recursive Query Processing Strategies," SIGMOD '86, pp.16-52, 1986.  
 [Ceri86]Ceri, S., et al. "Translation And Optimization of Logic Queries:The Algebraic Approach," VLDB86, pp.395-402, 1986.  
 [Gallaire84]Gallaire, H., et al., "Logic and Data Bases:A Deductive Approach," ACM Comp. Surveys, Vol.16, No.2, 1984.  
 [Miyazaki82]Miyazaki, N., "A Data Sublanguage Approach to Interfacing Predicate Logic Languages and Relational Databases," ICOT Technical Memorandum, 1982.  
 [Miyazaki86]Miyazaki, N., et al., "Compiling Horn Clause Queries in Deductive Data Bases:A Horn Clause Transformation Approach," ICOT Technical Report, 1986.  
 [Roberts79]Roberts, C., "Partial-Match Retrieval via the Method of Superimposed Code," Proc. IEEE, 67-12, pp.1624-1642, 1979.  
 [Wada87]Wada, M., et al., "A Superimposed Code Scheme for Deductive Data Bases," 5th IWDM, 1987, to Appear.  
 [有川82]有川 他, "重ね合わせ符号と逐次サーチを利用する文献情報検索システムについて," 情処第25回全国大会, 3P-10, 1982.  
 [有川83]有川 他, "重ね合わせ符号と逐次サーチを利用する文献情報検索システムについて(2)," 情処第27回全国大会, 7K-8, 1983.  
 [宮崎86]宮崎他, "KBMS PHI (2) 知識とデータに関する一考察," 情処第34回全国大会, 2M-6, 1986.  
 [宮崎87]宮崎, 伊藤, "演繹データベースにおける制約付最小不動点," ICOT Technical Report, 1987.  
 [吉田86]吉田他, "KBMS PHI (3) 分散知識ベース制御方式," 情処第34回全国大会, 2M-7, 1986.