

TM-0174

知識ベースシステム—KBMS PHI
—部分評価による知識とデータの扱い—

宮崎収兄(沖電気工業), 横田治夫(富士通)
伊藤英則, 森田幸伯, 坂間千秋, 大場雅博, 村上昌己

June, 1986

©1986, ICOT

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

伊藤英則 森田幸伯 坂間千秋 大場雅博 村上昌己

宮崎収兄* 横田治夫*

(財団法人 新世代コンピュータ技術開発機構)

(沖電気工業株式会社)* (富士通株式会社)*

1. はじめに

第5世代コンピュータ・プロジェクトの中期では知識情報処理システムとして推論システムと知識ベースシステムの物理的、論理的関連をさらに密にしてそのプロトタイプを開発する。知識ベースシステムの開発は、これまでに蓄積したデータの蓄積/管理/処理、専用エンジン⁽¹⁾、並列実行管理/並列資源管理等⁽²⁾の技術を知識情報処理の視点からさらに拡張する方向で進めている。

このための制御機構は以下のモデルとしている。

- (a) 基本モデル：推論型単一プロセッサ(PSI)と連想サーチ/知識演算専用エンジンからなる基本構成⁽³⁾
- (b) 分散モデル：ICOT-LANにより基本モデルを分散化⁽³⁾
- (c) 並列モデル：ネットワークとメモリ多重アクセス制御機構により基本モデルを並列化⁽⁴⁾⁽⁵⁾

また、知識ベース管理からみた知識とデータの扱いに関しては以下のモデルとしている。

- (a) 結合モデル：関係データベース管理層と知識管理層を構築し、2つの階層により知識ベース管理機能を実現⁽³⁾
- (b) 融合モデル：項(term)を単位とする関係型知識モデルによる知識とデータを融合管理する機能を実現⁽⁴⁾⁽⁵⁾

制御機構モデルと知識ベース管理モデルの組合せについては、適用分野から考察し選定する必要もあるが、これらの基本的特性の解析を行なうために、その一例として結合モデルを基本モデルと分散制御モデル上に研究開発を進めている。これらをそれぞれ知識ベースシステム PHI/1、および、分散知識ベースシステム PHI/2と呼ぶ。さらに、並列制御機構モデルと融合モデルの組合せによる実験システムの研究開発も進めている。

ここでは、結合モデルの知識ベースシステムに部分評価⁽⁶⁾(partial evaluation)の概念を導入し知識のコンパイルとデータの検索についての考察を中心に述べる。

2. 論理型言語処理系と関係データベースの結合の必要性

論理型言語処理系は推論機能および知識表現能力に優れている。関係型データ処理系はデータの集成的取り扱いの定式化が論理的に明確である。さらに関係型データは論理型言語の限定した範囲での表現とみることができる。論理型言語処理系と関係データベース処理系を結合してこれらの利点を相補的に活かしたシステムの意義付けは既に指摘されている^{EX. (7)}

論理型言語としては述語論理を基礎としているprologを使用する。prologプログラムと関係データベースシステム(RDBMS)と結合する手法にはいずれを主体とするかにより以下の2つがある。

(1) prologの演繹機能をRDBMSに付加しRDBMSの機能を高級化する。また逆に、

(2) prolog処理系にRDBMSの大量DB管理機能および処理機能を付加する。

(1)の観点は、演繹DB機能を実現することが特徴であり、(2)の観点は、例えばPL/1におけるSQLの埋め込みと同様にprologプログラムからデータベースへのアクセス機能を実現する立場である。現在までのところ、(1)の観点からの研究が多く行なわれている。その理由は、プログラム規模が比較的小さく(2)の必要性が明確に認識されていなかったことと、prologによる演繹DB機能がprologが持つ幾つかの機能は引き出し得ていたことによると考えられる。

しかしながら、prologのバックトラック等の知識制御機能等の高度なものについては(1)の観点からの結合では扱い難いこと、また例えば、知識の結合とも考えられるリスト結合処理Append等については(2)が自然であること、および、将来プログラムが大規模化することに対応するためには、(2)の手法は今後さらに重要となりデータ/知識の扱いに高度な推論機能を活用できること、メタ知識/オブジェクト知識を同一処理体系で扱えること、またメタプログラミング手法により知識管理/実行管理機能を充実できることなどの点で拡張性に富んでいる。

3. 三階層結合モデルによる知識ベース管理システム

論理型言語によるユーザプログラムが内部データベースIDB(インテンショナルデータ)、関係データベース管理システムが外部データベースEDB(エクステンショナルデータ)を管理、双方は、論理的に関係代数コマンドインタフェースにより結合した知識ベース管理システムの二階層モデル(図1)はKaiser/Deltaで研究開発してきた。

しかしながら、以下のような問題が残された。

- (1) ユーザプログラムと知識ベース管理システムの関係の明確化
- (2) ユーザプログラムと知識ベース管理システム間インタラクション量と物理的システム構成の関連付け
- (3) 知識の管理、特に知識の格納方法や知識のインテグリティ管理機能、等の拡張性

そこで、これらの問題を解決するために知識ベース管理システム内でIDBおよびEDBを管理し、ユーザプログラムは知識ベース管理システムに対してホーン節でIDBおよびEDBを問い合わせ/検索する論理インタフェースをもつ三階層結合モデル(図2)を提案した⁽⁸⁾。



図1 二階層結合モデル

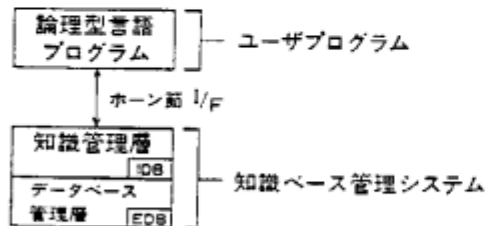


図2 三階層結合モデル

このモデルは知識ベース側に推論機能を活用することにより検索機能の論理型処理への拡張、EDB アクセスの最適化を知識ベース管理システムで実行し、システム全体として三階層モデルより処理の高機能化、効率化、ユーザプログラムの負荷軽減を狙え得るものである。

知識ベース管理システムをデータベース管理層と知識ベース管理層から構成し、データベース管理層ではEDBとしてリレーショナルデータに対応するホーン節のファクトを管理し、知識ベース管理層ではIDBとしてホーン節のルールに対応する知識を管理する。ここで、IDBはEDBに対するviewであり、EDBを検索するための知識であると解釈する。このことはview定義の拡張を論理型言語で行なうことを意味する。

4. 知識ベース管理システムの部分評価による

ホーン節問い合わせ処理

4.1 部分評価と知識コンパイル

ホーン節問い合わせをモデル論的にも証明論的にもとらえることができるが、いずれにしても問い合わせ処理の最適化を知識ベース管理側で実行することは重要である。ここでは、IDBをprologプログラムの集合と見なした、ホーン節問い合わせ処理の最適化のために部分評価の概念を導入する。すなわち、ホーン節問い合わせをトリガーとして知識ベース管理システムのIDBに対する部分評価手法を三階層結合モデル上で考察する。このモデルにおける知識の部分評価を知識コンパイル⁽⁹⁾の一例としてとらえる。

知識ベースのIDBに部分評価手法を適用して

- (1) ある一つの問い合わせに対して、EDB探索回数／探索空間を限定する。
- (2) 複数(m)の問い合わせに対して、共通のEDB探索が

ある場合にはその部分を $n(n \leq m)$ 回だけ検索する。

以上の(1)、(2)を実行するために、ユーザからの問い合わせを基にIDB(知識)をコンパイルする内部問い合わせを生成する。この内部問い合わせは見方を変えれば、EDB検索の処理の効率化のためのものと位置付けたメタキューリである。なお、ユーザプログラムから直接メタキューリを受けることも、また、知識ベースシステムがメタ知識を使って自発的にメタキューリを発行する等の場合が考えられる。

4.2 部分評価によるホーン節問い合わせ処理の例

知識ベース管理システム内で、4つのステップから構成するホーン節問い合わせ処理例を以下に示す。(図3)

(1) メタキューリ生成

ユーザプログラムからのホーン節問い合わせ(キューリ)からIDBの部分評価のためのメタキューリを生成する。

(2) 知識コンパイル

(1)で生成されたメタキューリから部分評価によりIDBのコンパイルを行なう。この知識コンパイル手法をHCT法⁽¹²⁾(Horn clause 変換法)と呼ぶ。HCT法はIDBの再帰定義の有無を検出し、さらに、EDB部での評価に必要な中間述語(中間的な仮想リレーション)形式に変換する。

(3) 関係演算コマンド生成

知識コンパイルされたものから関係演算コマンドを生成する。知識コンパイル後に再帰定義がなければ非繰返し型関係演算コマンド、再帰定義がある場合は繰返し型関係演算コマンドを生成する。

(4) 関係演算実行

(3)で生成された関係演算コマンドによりEDBを検索する。

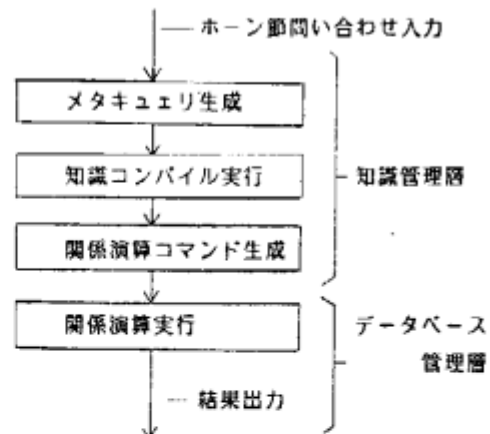


図3. ホーン節問い合わせの処理方式

(1)～(4)の基本的考え方をメタ述語 SET, METAQUERY, COMPILER, DEMOを用いて例示する。

```
SET(world:A)
METAQUERY(condition(A), query, metaquery)
COMPILE(metaquery, IDB, IDB')
DEMO(IDB'/EDB, retrieve, answer)
```

〔例1〕

今、知識ベースの一つのworld に以下のようなIDB/EDB が存在しているとする。

```

world  blood relationship
IDB    ancestor(X,Y) :- parent(X,Y)
       ancestor(X,Y) :- parent(X,Z),
                               ancestor(Z,Y)  (1)
       parent (X,Y) :- edb (father (X,Y))
       parent (X,Y) :- edb (mother (X,Y))
EDB    father relation
       mother relation

```

ここに、edb(relation(X,Y)) はEDB にrelation(X,Y) が存在していることを意味する。

このIDB をAND/OR木に展開すると図4のようになる。



図4. ancestorのAND/ORグラフ

4.2.1 メタクエリ生成

ユーザプログラムからのホーン節問い合わせを基にIDB へのメタクエリを生成する。

〔例2〕

```

query:- ancestor(taro,Y)
SET(world : blood relationship)
METAQUERY(condition(b.r.), query, metaquery)
metaquery:- ancestor(X,Y) / X-taro
なお、X-taroはEDB 検索のための関係演算コマンド
生成まで保存する。

```

4.2.2 HCT法による知識コンパイル実行

HCT 法は概念的にはprologのようなホーン節を基礎とした定理証明機構で問合せを評価する過程で、もし、先祖のノードと同じ述語が表われたらそこで評価を止め再帰記述と判定する。また、edb 述語でも再帰述語でもなければ評価を続け、EDB 問合せを必要とする述語と再帰述語の評価を後回しにする。HCT 法をAND/OR木上で説明する。すなわち、メタクエリをルートとするAND/OR木の展開過程で、先祖のノードと同じ述語が子孫にあらわれると再帰述語と判定し、そのノードの展開を行わない。木展開の過程で中間的な述語は消去され、EDB 問合せの必要な述語のみが残る。また木展開途中でORノード（1つの述語をヘッドとする2つ以上の節）があり、そのノードを表わす述語が消去されると、その親ノードをORノードにする。従って木

展開中に見つかったORノードをルートに近いノードに移すアルゴリズムと見なすことができる。AND ノードについても同様である。

以上のアルゴリズムによって図4は図5のAND/OR木に等価変換される。

これから分かるように、IDB(1)は以下の〔例3〕のホーン節(2)に等価変換される。

〔例3〕

```

COMPILE(metaquery:ancestor(X,Y), IDB(1), IDB(2))
ancestor (X,Y) :- edb(father (X,Y))
ancestor (X,Y) :- edb(mother (X,Y)),
ancestor (X,Y) :- edb(father (X,Z)),
                               ancestor (Z,Y)
ancestor (X,Y) :- edb(mother (X,Z)),
                               ancestor (Z,Y)

```

等価変換されたホーン節表現では、ヘッド部はゴール述語か、再帰述語のみであり、ボディ部はedb 述語が再帰述語、比較述語のみである。

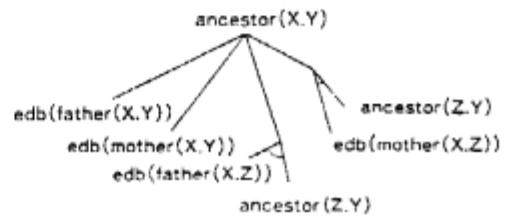


図5. ancestorの等価AND/ORグラフ

4.2.3 関係演算コマンド生成

4.2.2で述べたHCT 法により、等価変換されたホーン節から関係演算コマンドを生成する方法は以下である。

等価ホーン節に再帰述語があれば繰返し型関係演算コマンドを生成し、再帰述語がなければ非繰返し型関係演算コマンドを生成する。

〔例4〕

〔例3〕の等価ホーン節(2)と〔例2〕で保存していたX-taroから以下の関係演算コマンドを生成する。

```

DEMO(IDB(2)/EDB, X=taro, answer)
ancestor2 =
π2 ((σ1=taro (father))
μ( σ1=taro (mother))
μ( π1,4 ( σ1=taro (father ancestor))
2=1
μ( π1,4 ( σ1=taro (mother ancestor)))
2=1

```

以上、メタクエリからIDB の再帰記述を検出し完全コンパイルによるHCT 法を述べた。この方式の特徴としては以下に整理できる。

(1) 任意のメタクエリからIDB の等価ホーン節変換を行な

い、つぎに、繰返し型関係演算コマンドを生成することによりEDB 検索できる。

- (2) 非再帰型の等価ホーン節は非繰返し型の関係演算によりEDB 検索できる。
- (3) 等価ホーン節はedb 述語以外は現れない。

5. HCT 法の評価/考察

メタクエリを基に部分評価をIDB 部に対して実行し、EDB 部への問合せをコンパイル生成する方法は以下の2つに分類できる。

- (a) 部分コンパイル法⁽⁷⁾
- (b) 完全コンパイル法⁽⁷⁾

部分コンパイル法の例として遅延評価法⁽¹⁰⁾がある。部分コンパイル法は、IDB の処理で再帰性の検出を実行しないことによりEDB への問合せが停止しない場合がある。

完全コンパイル法では、IDB 全体を眺めた処理によりIDB の処理内で再帰表現されたものについては、その後の処理は有限集合であるEDB 検索を繰返し実行するので必ず停止する。完全コンパイル法の例には以下のものがある。

- (a) セッティング評価法⁽¹¹⁾
- (b) HCT 法⁽¹²⁾

セッティング評価法は、本来繰返しの不要な非再帰型問合せも繰返し型のEDB 問合せになる。また複雑な問合せで中間的にあらわれる述語も仮想リレーションとして評価するのでオーバーヘッドが生じる。

HCT 法では、仮想リレーションに対するEDB 評価を行わないこと、再帰性の検出をEDB 問合せ以前に行っているので再帰性のない問合せについては上記の処理オーバーヘッドを削減できる。

HCT 法はさらにつぎの利点がある。

- ・従来の評価方式では明らかでなかったが停止条件の明確化により、失敗としての否定(negation-as-failure)を含む再帰問合せの処理が可能である。
- ・知識ベースマシン側の知識検索にprologによる構文データ処理やカットなどの制御を可能にする拡張性が生まれた。

以上より、3. で述べた問題点のうち(1),(2)についての解決の見通しを得たことになる。また、(3),(4)についてはその解決に向けた制御機能を知識管理層に取り込むための検討の基礎を得たことになる。

6. おわりに

結合モデルは論理型言語処理系の知識表現能力・推論機能と、論理型言語と親和性がある関係データ処理系のデータ処理・管理機能を相補的にシステム化する狙いであった。KBMS PHIでは、論理型言語処理系に関係データ処理系を取り込む手法を探り、知識管理層とデータベース管理層からなる知識ベース管理システムを構成した。特に知識管理層に部分評価の概念を導入し遅延評価法、セッティング評価法、および、ホーン節変換(HCT)法について比較評価した。これらの方法は、知識コンパイルの一例としても見ることができ、不要なデータ検索を削除でき処理効率の向上が狙え得ることを示した。また、ホーン節変換(HCT)法そのものの拡張性はもとより、知識管理層へ論理型言語

処理系の能力を取り込む拡張性に富んでいることを示唆した。なお、EDB の持つ構造的性質からみた最適化とIDB 処理の最適化を考慮して総合的に捕らえた考察も今後の課題である。

さらに、ここでは述べなかったが分散知識ベースシステムに部分評価の概念を拡張することができる。例えば、分散知識ベースシステムにおける不完全知識ベース群への知識の提供等、を部分評価により実行させる協調問題解決の考察の基盤を設けることができる。この研究は PHI/2上で行なうことにしている。

最後に、KBMS PHI/1, PHI/2はICOTで開発した推論マシンPS I 上にESP にて開発中である(図6)。

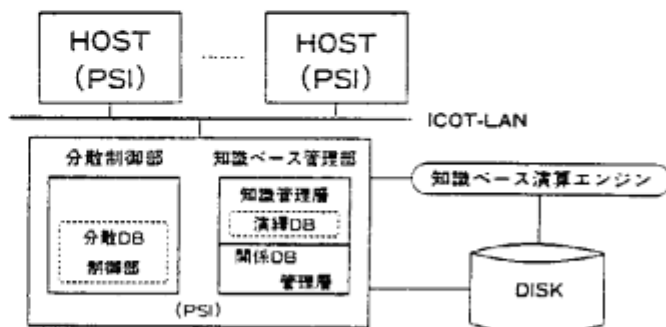


図6 PHI のシステム構成

謝 辞

熱心なご討論を戴いた研究空員の方々、OKB 打合せメンバーの吉田至長およびKC打合せメンバーの方々、また、文献(6)を基に有益なご示唆を戴いた古川至長、竹内両氏に感謝します。

参 考 文 献

- (1) K.Murakami, et al. A Relational Database Machine: First Step to Knowledge Base Machine, Proc. of 10th Symp. no Comp. Archi. June 1983.
- (2) H. Itoh, C. Sakama, et al. Parallel Control Techniques for Dedicated Relational Database Engines, ICOT TR-182.
- (3) 伊藤, 他 KBMS PHI(1) 情報処理第32回全国大会
- (4) 横田, 他 MPPHの構成方法 電子通信 S80 全国大会
- (5) H. Yokota, H. Itoh. A Model and an Architecture for a Relational Knowledge Base, The 13th Annual International Symp. on Comp. Archi. June 1986.
- (6) A. Takeuchi, K. Furukawa. Partial Evaluation of Prolog Programs and its Application to Meta Programming, ICOT TR-126.
- (7) H. Gallaire. Logic Databases vs Deductive Databases. Logic Programming Workshop 1983 ALB.-Portugal.
- (8) 宮崎, 阿比留, 他 KBMS PHI(2) 情報処理第32回全国大会
- (9) 村上, 他 知識ベースマシンの知識コンパイラ (4)に同一
- (10) H. Yokota, et al. An enhanced inference mechanism for generating relational algebra queries. ICOT TR-026
- (11) H. Yokota, et al. Deductive Database System on Unit Resolution, The 2nd Data Eng. Comp. Conf. pp. 228-235 Feb. 1986
- (12) N. Miyazaki, et al. Compiling Horn Clause Queries in Deductive Databases ICOT TR-183.