

TM-0471

知識ベースマシン

伊藤英則

March, 1988

©1988, 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**

## 知識ベースマシン

### [ 1 ] 発展形態の動向

知識ベースマシンとはある程度量的にまとまりのある知識集合の操作・管理を効率よく処理するためのマシンである。知識ベースマシンアーキテクチャについては、扱うオブジェクトとその知識表現言語の種別、知識集合のシステム内での論理的・物理的位置および格納するオブジェクト量・質とその物理的媒体などの個々の組合せによりそれぞれ異なる。これらを適応分野ごとに整理した、アーキテクチャとしての必要十分条件の明確化は今後の課題であろう。

このような状況下で、従来のデータベースマシン（データベースマシン自身の定義そのものも必ずしも明確とは言えないが）を基本として知識情報処理機構を付加した知識ベ-

スマシンの発展形態がある。また一方では、知識処理向き言語といえるLisp, Prologなどの高級言語用マシンがより普及し、これを基本として種々の応用分野から知識ベースの構築が進むと同時に、知識ベースの操作・管理機能の高級化・高速化を狙って知識ベースマシンへと発展してゆく方向がある。さらに、処理の高速化のために並列マシンアーキテクチャ上での研究開発も行なわれている。さらには、最近では超並列コネクションマシンおよび脳神経細胞モデルとしてのニューロンマシン等の研究開発も盛んである。いずれにしても、ある程度応用分野と機能レベルを限定するなどした知識ベースマシンアーキテクチャの研究成果を反映しつつ、より汎用高級マシンへ融合または専用高級マシンへ特殊化してゆくであろう。

なお、文献(清木 1987)はデータベースマシンに関連する最近の新しい研究分野として、関係データベースマシン、演繹データベ

ースマシン、メインメモリデータベースマシン、汎用性を重視した大量データ処理用マシン、高機能二次記憶システム、データベースマシンのオペレーティング・システム等を挙げている。

ここでは、知識（オブジェクト）は論理式であり、知識ベースはその集合であり、その操作・管理にも知識（メタ）を用いて論理式を基本として実行することを知識ベースの前提条件にする。そのため論理式を効率的に扱える高級言語マシンアーキテクチャを基本とする演繹データベースマシンを起点として、その拡張などに重心をおいて議論を進める。

## 〔2〕知識表現言語と

### 知識ベース操作基本機能

知識を表現する言語としては、エキスパートシステムの開発で一般に使用されているプロダクションルール、フレーム、セマンティックネットワーク、さらにはこれらを拡張し

たものから自然言語に近いものまでである。

ここでは文献 (Gallaire 1984) に従い、一階述語論理のうちで図 1 に示す確定節の部分を知識表現言語とする。これは、上述のユーザプログラミング言語の理論的基盤であり広い機能範囲を包含しているので、これを知識ベースマシンで意識する中間言語とする。このため知識ベースマシンで具備すべき知識の蓄積・検索／探索・管理機能などの諸機能は確定節で表現された知識集合を処理の対象とすることになる。このことにより、知識ベースとその検索／探索・管理等の諸プロセスとは共に確定節を基本として記述するので、これら双方は共通の枠組みでとらえることができる。この利点を活かしたプロセス指向の知識ベースの概念が派生し、オブジェクト指向知識ベースシステム構築に新しいパラダイムを与えることが期待される。

また、大量の知識ベース内から要求条件を満足する知識全てに対して処理（全解収集処

理) が可能であることと、処理効率化を目的としたオブジェクト知識の整列・ハッシング・インデッキングなどの諸機能が必須である。知識ベースマシンのオペレーティング・システムへの要求条件としては、知識演算処理、知識ベース管理機能の実現性を保証すること、および、処理過程で処理対象グラニアリティの動的変化に対応する、プラグマティックな実行管理・資源管理機能を充実させることが挙げられる。さらには、[3]で述べる並列マシンアーキテクチャ上での機能実現は今後の大きな研究課題である。

### [3] 知識ベースマシンモデルの例

知識ベースマシンの基本的なモデルの一つとして、図2に示すような推論処理機構、主記憶機構および外部記憶機構からなるものが考えられる。特に、推論処理機構はWAM(Warren 1983)コードを効率よく実行するためのヒープ、スタックさらにはトレール機能を

具備する。なお、外部記憶機構を持たない推論処理機構と主記憶機構だけからなるメインメモリ知識ベースマシンも実在する。また、必須ではないが主記憶機構と外部記憶機構の間にデータストリームをパイプライン的に処理する知識演算専用装置が処理効率のために要求される。ここで、外部記憶機構に格納する知識の論理的構造形式に依存して知識演算専用装置が具備すべき機能は異なる(Itoh 1986)。例えば、図1のタイプ1(事実)の集合のみを格納する場合は、事実の集合を関係データの集合とみなすことができるので関係代数演算の専用装置(Murakami 1983)(岩田1987)となる。図3に、2ウェーマージソートアルゴリズム(Todd 1978)により実現したダブルバッファを持つソートユニットと関係代数演算ユニットからなる関係代数演算専用装置を示す。また、図1のタイプ3(ルール)の集合の場合は知識検索の基本に単一化(unification)機能を取り込めば、大量の知識に

浅い推論を繰返し実行し幅優先(breadth first)で解の候補を絞り込むような推論機構の処理負担を軽減できる。このために、知識集合間を対象として関係代数演算と単一化演算を融合させることにより知識を演繹検索できる単一化検索手法(RBU: retrieval by unification)が提案されている。このモデルは、従来の関係データモデルで定義されていたデータを項(term)へ拡張したものであり、関係項モデル(Yokota 1986)と呼ばれている。この上での基本演算である単一化制約、単一化結合演算をそれぞれ図4、図5に例示する(伊藤 1987)。図6に単一化関係演算を使って演繹検索ができるアルゴリズムの例を示す。これらの演算によりデータストリーム形式でパイプライン処理するために、図7に示すような単一化検索専用装置が試作実験されている(Morita 1986)。また、単一化関係演算を効率よく実行させるために推論機構内で実現されているクローズインデッキング手法を



拡張した、項集合の順序付け、整列、索引付け、重ね合わせコード (superimposed code) およびトライ (trie) 手法についても種々提案されている (Morita 1988) (Nakase 1988)。

これまでに述べた知識ベースマシン基本モデルの拡張としてはその他、(i) 推論機構の並列化、(ii) 専用装置の並列化 (Itoh 1987)、(iii) 外部記憶機構への同時多重アクセスを許すネットワーク化 (Tanaka 1984) (Monoi 1988) などがあげられる。

推論機構のOR並列化には、バックトラック時のタスク切り換え処理と変数束縛処理の負荷バランスを考慮した S R I、A R G O N N E、M A N C H E S T E R、S I C S モデル等各種のマシンが研究開発されている (Warren 1987)。また、AND 並列化としては、AND 並列論理型核言語をもつ ICOT、およびワイツマン、インベリアルモデル等がある。与えられた問題の解決に手分けして互いに独立に解を求める手法と高密度の通信をしながら互い

に協調して解を求める手法とがある。前者はOR並列処理向きであり、後者はAND並列処理向きである。当面の知識ベースマシンとしてはある程度まとまりのある知識集合から全解を収集するために前者を基本としてより高度な探索機能のために後者を取り込んだものとなろう。将来の知識ベースマシンの理想形態としてはこれら双方の利点をバランスよく整合した抽象マシンの処理系とマシンアーキテクチャの研究成果を取り込んだマシンへ発展してゆくであろう

#### 〔4〕演繹処理

〔2〕で述べた確定節を知識表現言語とするのでこれを処理する代表的な基本機能の一つとして演繹がある。演繹ができるので従来のデータベースマシンより柔軟で高度な処理が可能となる。以降の議論では特に断らない限りこれを演繹データベースと呼ぶ。

演繹データベースは従来の関係モデルとは

異なり知識を論理式の形で表現でき、しかも  
ルールの再帰的な表現の論理式も許される。  
このようなルールに対する問合せ処理は演繹デ  
ータベースの特徴であり、特に“再帰問合せ  
(recursive query) 処理”と呼ばれている。  
この処理問題は、この数年来の演繹データベ  
ースにおける中心的テーマの一つであり、つ  
ぎつぎに完全性と停止性を保証する新しいア  
ルゴリズムが提案されてきている (Bancilhon  
1986)。

再帰問合せ処理の方法には二種類に大別さ  
れる。ひとつは、Prologのように問い合わせ  
(ゴール) に対して逐次 (通常は導出原理に  
基づいて) サブゴールへの展開ルールを適用  
していき、最終的にEDB (Extensional Data B  
ase: 事実の集合) の検索を行い解を求めるト  
ップダウン型の処理である。これに対して、  
EDB の要素から出発して、それからルールに  
よって導入される事実の集合を集めつつ繰返  
しルールを評価して最終的に問い合わせに対

応する解を求めるボトムアップ型がある。この両者の方法はいずれも一長一短がある。

トップダウン型の処理ではゴールをサブゴールに展開するとき基本的に一度に一タブルずつ処理を行うため処理効率が悪い。

一方、ボトムアップ型の処理では、EDB 全体を一度に処理できるという点ですぐれているが、元々の問い合わせとは無関係な事実も中間的に生成してしまうので、途中で無駄な計算を繰返してしまう。このボトムアップ型の処理の中に、トップダウン型処理の長所を取り入れようとしたアルゴリズムがマジックセット法 (Bancilhon M. 1986) (Beerl 1986) やアレキサンダ法 (Rohmer 1986) として知られている。これらの方法は共に、問い合わせに現れる定数に着目し、ルールによる導出の際にどの変数が束縛されるかという変数の束縛情報をもとにして、導出される実事の絞り込みをおこなっている。

例として、次の簡単なプログラム (IDB: int



```

magic __bf(X)

R2: ancestor__bf(X,Y) ← parent(X,Z),

    magic __bf(X),

    ancestor__bf(Z,Y).

R3: ancestor__bf(X,Y) ← parent(X,Y),

    magic __bf(X).

R4: query __f(X) ← ancestor__bf(太, X).

R5: magic __bf(太).

```

上の変換されたルール集合とEDBから、その不動点をボトムアップ的に計算することにより元の問い合わせの解を求めることになる。素朴なボトムアップ法では、問い合わせに無関係な計算（上の例ではpeterの先祖等）を行ってしまうが、ここではmagic \_\_bf等のアトムを追加することによりそれを回避している。これによって、トップダウン型の評価の場合と同じ変数の束縛の伝播を行う事ができ、ボトムアップ型の計算でトップダウン型の実

行を模擬していることになる。

その他、与えられたIDBに部分計算(二村 1983)を施すことによって無駄な計算を削除するホーン節変換アルゴリズムも提案されている(宮崎 1988)。

また、問い合わせを集合で扱い、その処理に部分計算の概念を導入して効率化を図ったアルゴリズムも提案されている。例えば、問い合わせ集合が与えられた場合、

その最小汎化(least generalized query)で評価を行なう。これにより、共通部分の評価は一度だけ計算すればよいので全体の計算量を削減できる。最小汎化法は与えられた問い合わせ集合に類似のもの(compatible)が多い場合に有効である。

また、問い合わせ集合の述語の順序関係の低位のものから順に評価すれば上位の述語からなる問い合わせは低位の評価結果を利用できる。このような最大汎化法 MGQ(most generalized query)は与えられた問い合わせ集

合に階層的(hierarchical)なものが多い場合に有効である。

なお、最小汎化と最大汎化を組み合わせた簡単な方式の実験による処理効率の評価が(Sakama 1987)で議論されている。

つぎに、多重世界における知識の連想処理モデルの一例について述べる。一般に多重世界(multiple world)を考えた場合、知識 $K$ がそれぞれの世界 $W$ で異なる解釈を持つ場合が起こり得る。例えば、世界 $w_1$ では知識 $K$ は $K_{w_1}$ と解釈されるが、別の世界 $w_2$ では $K_{w_2}$ と解釈される場合を考える。このとき、知識 $K$ を各々の世界で解釈する $K_{w_i}$ を知識ベースに格納させると、一般には冗長である。

例えば、 $K = p(X, Y)$ 、 $K_{w_1} = p_{w_1}(X, Y)$ である場合、 $p = p_{w_1}$ の同値関係が定義されていれば $K$ から $K_{w_1}$ を導出できる。

このように知識 $K$ を多重世界 $W_i$ における代表元として、各世界での同値知識を処理すれば、ある知識の集合に対して世界によって異



なる解釈を与えることもできる (Sakama 1988)。

このように最小構成要素を代表元として世界によって異なる解釈を与える処理をある種の連想と見ることもでき、今後の柔軟な知識ベース管理の要素機能として期待される。

#### [6] 演繹データベースの拡張

演繹データベースは、普通確定ホーン節から成るデータベースでは関数記号は定数のみであるクラス (Datalog と呼ばれている) を対象にしている。これをさらに関数記号を含む場合や否定を含む節にまで扱えるような拡張も考えられつつある。

演繹データベースでは知識は論理式で表現されるが、ここでどの程度の範囲の論理式を許すかは、問い合わせ処理等の複雑さとのトレード・オフの問題となる。演繹データベースの研究の初期の頃は、確定節から成るデータベースがよく扱われていた。この場合は否

定情報が明示的には表現できないので、何らかの否定情報を導く形式化が必要になってくる。通常はReiterの閉世界仮説 (Closed World Assumption, CWAと略する) によることが多い。しかし、確定節から一般の節への拡張するとCWAは矛盾してしまい適用できなくなるので、その改善策として一般化されたCWA(GCWA)(Minker 1982)などが提案されている。このような概念はCircumscription(McCarthy 1980)と密接に関連していることが知られている。

確定節を不確定な情報が生じないように拡張したクラスとして、“層状(stratified)”プログラム(Apt 1986)がある。これはプログラムに出現する述語記号間にある階層(レベル)が存在して、どの述語についてもその述語を定義するルールにおいては、否定情報は定義される述語のレベルより下のレベルの述語によるもののみが用いられるプログラムである。このような層状プログラムでは、確定節の場

合と同じ様に不動点セマンティックスを用いて不確実な情報を生じることなしにその意味を定義できる。また、層状データベースに対する問い合わせ評価アルゴリズムも提案されつつある (Seki 1988)。

否定情報と同様に知識ベースの表現能力拡張の方向として、集合概念が扱えるようにデータベース言語を拡張する研究も MCC (LDL) (Beerl, N 1987) や INRIA (COL) (Abiteboul 1987) 等で行われている。これは、集合と論理プログラミングのパラダイムを結びつけようとする動きであり、データベース言語 (Codd 1972) 設計の新しい中心的課題の一つと考えられる。

つぎに、演繹データベースの拡張として例外を含む性質継承処理をとらえる。例えば、

- ・ 象は灰色である。
- ・ ロイヤル象は象であるが、灰色でない。
- ・ clyde はロイヤル象である。

のように、性質継承に例外を含むような場合、

一階述語論理では例えば以下のように現すことができる。

$$\begin{aligned}
 W = [ & \forall x \text{Royal-Elephant}(x) \\
 & \supset \text{Elephant}(x). \\
 & \forall x \text{Royal-Elephant}(x) \supset \neg \text{Gray}(x). \\
 & \forall x \text{Elephant}(x) \\
 & \quad \wedge \neg \text{Royal-Elephant}(x) \\
 & \quad \supset \neg \text{Gray}(x). \\
 & \text{Royal-Elephant}(\text{clyde})]
 \end{aligned}$$

ところが、Wに例えば $\text{Elephant}(\text{taro})$ を加えると、 $\text{taro}$ が $\text{Royal-Elephant}$ でない場合、 $\text{Gray}(\text{taro})$ を演繹するためには $\neg \text{Royal-Elephant}(\text{taro})$ もWに表現しておかねばならない。このように、一階述語論理で非単調な性質継承を表現しようとする、それらを陽に記述しておかねばならないという表現量の問題がある。

そこで、このような非単調な性質継承の形

式化としてデフォルト論理を用いた方法が提案されている(Etherington 1983)。

また、この様な意味ネットワークにおける並列性質継承アルゴリズムの研究も最近活発に行われている(Touretzky 1986)(Sakama, 1988)。さらに、多重継承における性質継承のための並列アルゴリズムも提案されつつある。

最後に、データベースマシンおよび知識ベースマシンの事例の枚挙については文献(清木1987)に、評価方式、データ構造、アーキテクチャ構成、ソフトウェア構成、並列性について詳しく網羅的に整理されているのでここではそれを割愛した。

## 参考文献

Abiteboul, S. and Grumbach, S., "COL : a Logic-Based Language for Complex Objects." INRIA-TR, 714, Sept. (1987)

Apt, E.R., Blair, H., and Walker, A., "Towards A Theory of Declarative Knowledge." in Proc. of Workshop on Foundations of Deductive Databases and Logic Programming (J.Minker, editor), pp.548-828 (1988) Washington, DC.

Bancilhon, F. and Ramakrishnan, R., "An Amateur's Introduction to Recursive Query Processing Strategies." in Proc. of the ACM-SIGMOD Conference, pp.16-52 (1986) Washington, DC.

Bancilhon, F., Maier, D., Sagiv, Y., and Ullman, J., "Magic Sets and Other Stranger Ways to Implement Logic Programs." in Proc. of The 5th ACM-SIGMOD Symposium on Principles of Database Systems (1986)

Beeri, C. and Ramakrishnan, R., "On the Power of Magic." in Proc. of The 5th ACM Symposium on Principles of Database Systems, pp.269-284 (1986)

Beeri, C., Naqvi, S., Shmueli, O., and Tsur, S., "Sets and Negation in a Logic Database Language (LDL), Rev.1," MCC Technical Report No.DB-375-86, Rev.1, July (1987)

Codd, E.F. "Relational Completeness of Data Base Sublanguages." In Courant Computer Science Symposium of Data Base Systems (1972)

Etherington, D.W. and Reiter, R., "On Inheritance Hierarcies with Exception" AAAI'83, pp.104-108 (1983)

Gallaire, H., Minker, J., and Nicolas, J.-M., "Logic and Data Base. A Deductive Approach." Computing Surveys, 16, pp.153-185 (1984)

Itoh, H. "Research and Development on Knowledge Base System at ICOT," The 12th Int. Conf. on VLDB PP.437-445 (1986)

Itoh,H. and Sakama,C. "Parallel Control Techniques for Dedicated Relational Database Engines." The 3rd International Conference on Data Engineering (1986)

McCarthy,J.A. "Circumscription-a form of non-monotonic reasoning. Artificial Intelligence 13, pp.27-39 (1980)

Minker,J.. "On indefinite data bases and the closed world assumption." in Proc. 6th Conference on Automated Deduction, pp.292-308. Lecture Notes in Computer Science 138, Springer, Berlin (1982)

Monoi,H., Morita,Y., Itoh,H., Shibayama,S., and Sakai,H. "Parallel Control Technique and Performance of an MPPM Knowledge Base Machine." The 4th International Conference on DE (1987)

Morita,Y., Oguro,M., Sakai,H., Shibayama,S., and Itoh,H. "Performance Evaluation of a Unification Engine for a Knowledge Base Machine." to appear The 15th International Symposium on Computer Architecture (1988)

Morita,Y., Yokota,H., Nishida,K., and Itoh,H., "Retrieval-by-Unification, on a Relational Knowledge Base Model." The 12th International Conference on VLDB , (1986)

Murakami,K., Kakuta,K., Miyazaki,T., Shibayama,S., and Yokota,H "A Relational Database Machine; First Step to Knowledge Base Machine." Proc. of 10th Symp. on Computer Architecture (1983)

Nakase,A., Shibayama,S., Sakai,H., Morita,Y., Monoi,H., and Itoh,H., "Structural Superimposed Codeword as an Indexing Scheme for Terms," to appear The 14th International Conference VLDB (1988)

Rohmer,J., Lescouer,R., and Kerisit,J.M., "The Alexander Method - A Technique for the Processing of Recursive Axioms in Deductive Databases." New Generation Computing, 4 (3) : 273-285 (1986)

Sakama.C. and Itoh.H., "Partial Evaluation of Queries in Deductive Databases .," IFIP Workshop on Partial Evaluation, Denmark, Oct. (1987) and also to appear in New Generation Computing, Springer-Verlag

Sakama.C. and Itoh.H. : "Handling Knowledge by its Representative,"The 2nd Int. Conf. on Expert Database Systems, Virginia, Apr.(1988)

Seki.H. and Itoh.H., "An Evaluation Method for Stratified Programs under the Extended Closed World Assumption," (1988) ICOT TR-337

Tanaka.Y., "MPDC : Massine Parallel Architecture for Very Large Databases," International Conference on 5th Generation Computer Systems, pp.113-137 (1984)

Todd.S. "Algorithms and Hardware for Merge Sort Using Multiple Processors," IBM Journal Research and Development, Vol.22, No.5, Sep. (1978)

Touretzky.D.S., "The Mathematics of Inheritance Systems," Research Notes in Artificial Intelligence, Pitman, London (1986)

Warren.D.H.D., "An Abstract Prolog Instruction Set," Technical Note 309, SRI International (1983)

Warren.D.H.D., "Or-Parallel Execution Models of Prolog," TAPSOFT'87, Springer-Verlag, pp.243-259 (1987)

Yokota.H. and Itoh.H. "A Model and an Architecture for a Relational Knowledge Base," The 13th Architecture for a Int. Symp. on Computer Architecture (1986)

伊藤；並列論理型核言語に基づく知識ベースマシン，人工知能学会誌，Vol.2, No. 4 (1987)

岩田，柴山，酒井，伊藤，村上：関係台数専用装置の評価，情報処理学会誌Vol.28 No.7 (1987)



宮崎，羽生田，伊藤：ホーン節変換：演繹データベースにおける部分評価の応用  
情報処理学会誌 Vol.29 No.1 (1988) ICOT TR-258

藤野，“演繹データベースの動向”，アドバンスド・データベース・システム  
情報処理学会，昭和62年12月

清水，“データベースマシンの動向”，アドバンスド・データベース・システム  
情報処理学会，昭和62年12月

二村：プログラムの部分計算法，電子通信学会誌，  
Vol.66, No.2, pp.157~165 (1983)

図1 知識表現言語

| $K \setminus L$ | 0    | 1                                                                                                                                                                 | $>1$ |
|-----------------|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|
| 0               |      | <div style="border: 1px solid black; padding: 5px; display: inline-block;">                     タイプ1: 定数<br/>                     タイプ2: 変数                 </div> | タイプ5 |
| $\geq 1$        | タイプ4 | タイプ3                                                                                                                                                              | タイプ6 |

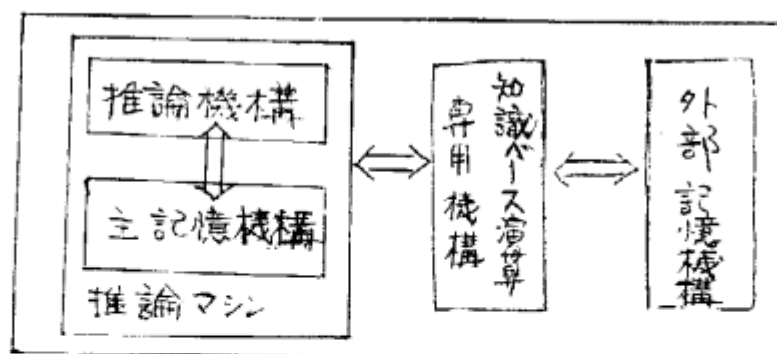
一階述語論理式

$$P_1 \wedge P_2 \wedge \dots \wedge P_k \longrightarrow Q_1 \vee Q_2 \vee \dots \vee Q_l$$

知識表現言語 ; タイプ1~3, 4

問合せ言語 ; タイプ4

図2 知識ベースマシンモデルの例



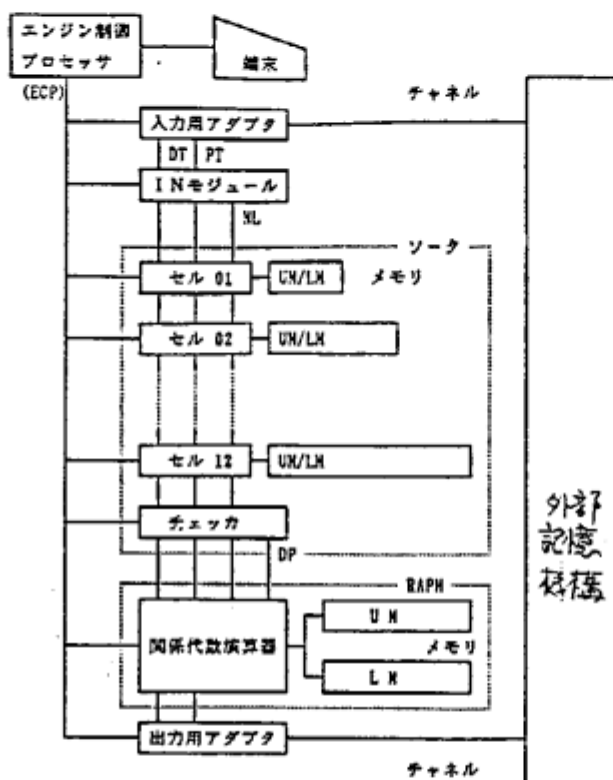


図3 関係代数演算専用装置の構成概念図

$$\begin{array}{ccc}
[1, 2] & [1', 2'] & [1'', 2'', 3''] \\
\left\{ \begin{array}{l} (f(a, y), g(x, y)) \\ (f(b, x), g(a, a)) \\ (f(v, v), g(b, y)) \end{array} \right\} & \begin{array}{l} \bowtie \\ 2 \diamond 1' \end{array} \left\{ \begin{array}{l} (g(a, x), h(x, b)) \\ (g(b, w), h(y, c)) \end{array} \right\} & = \left\{ \begin{array}{l} (f(a, x), g(a, x), h(x, b)) \\ (f(b, a), g(a, a), h(a, b)) \\ (f(a, w), g(b, w), h(y, c)) \\ (f(v, v), g(b, w), h(y, c)) \end{array} \right\}
\end{array}$$

図 5 単一化結合演算の例

$$\begin{array}{ccc}
[1, 2] & [1', 2'] \\
\sigma f(a, x) \diamond 1 \left\{ \begin{array}{l} (f(a, y), g(y, x)) \\ (f(b, x), g(a, x)) \\ (f(w, w), g(c, w)) \end{array} \right\} & = & \left\{ \begin{array}{l} (f(a, x), g(x, x)) \\ (f(a, a), g(c, a)) \end{array} \right\}
\end{array}$$

◇ : 単一化制約

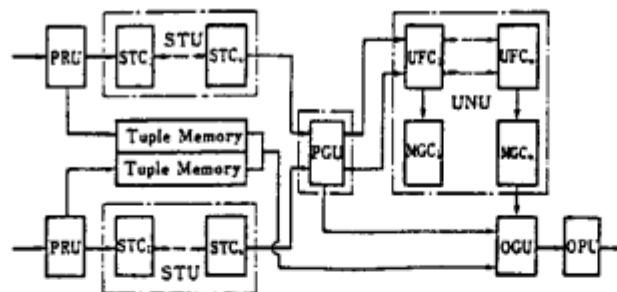
図 4 単一化制約演算の例

```

(R : result) ←  $\phi$ 
 $K_0 \leftarrow \sigma_{goal} \odot head (KB : term relation)$ 
 $i \leftarrow 0$ 
while  $K_i \neq \phi$  do
begin
   $R \leftarrow (K_i \text{ の body が } [] \text{ の } K_i \text{ の head}) \cup R$ 
   $K_{i+1} \leftarrow \prod_{K_i} \text{の head, KB の body } (K_i \text{ body } \odot head KB)$ 
   $i \leftarrow i + 1$ 
end (ただし,  $\prod$  は通常の射影)

```

図 6 単一化検索アルゴリズムの例



PRU: 前処理ユニット    PGU: ペア生成ユニット    OGU: 出力項生成ユニット  
 STC: ソートセル    MGC: 最汎単一化処理セル    OPU: 後処理ユニット  
 STU: ソートユニット    UNU: 単一化ユニット

図 7 RBU 専用装置の構成概念図