データベース・システム 57－4
（１９８７．１．１９）

# 知識ベースマシンにおける単一化専用装置の処理方式とその評価

森田　幸伯，小黒　雅己*，伊藤　英則

（財）新世代コンピュータ技術開発機構，* ＮＴＴ電気通信研究所

本報告では、単一化専用装置（単一化エンジン）の評価について述べる。

単一化エンジンは、単一化検索演算（ＲＢＵ演算）のための専用装置である。関係型知識ベースモデルは、知識ベースのひとつのモデルであり、そのなかでは、知識は項の関係として表現されて、単一化演算を検索に用いる。単一化エンジンは、データをストリーム状に流す方式を採っているため、その性能はデータ転送量に大きく左右される。

本報告では、単一化エンジンの一部である単一化処理部に流れるデータ転送量を少なくする方式についても述べる。

## Design and Evaluation of Unification Engine for a Knowledge Base Machine

Yukihiro Morita, Masami Oguro*, Hidenori Itoh.

Institute for New Generation Computer Technology
*NTT Electrical Communications Laboratories

In this paper we describe evaluation of performance of a unification engine (UE) for a knowledge base machine.

The UE is dedicated hardware that performs the retrieval-by-unification operation (RBU operation) on a relational knowledge base model. Relational knowledge base model is a conceptual model for a knowledge base in which the knowledge is represented by relations of terms and unification operation on terms is used as the retrieval mechanism. The performance of the UE mainly depends on the size of its data stream.

We propose a method which reduces the data stream for the unification unit, one component of the UE.

## 1．はじめに

新世代コンピュータ技術開発機構では、関係データベースマシンを拡張した関係型知識ベースマシンの開発を行っている［伊藤86a ］［伊藤86b ］。本マシンは、知識を変数を含む論理的構造体である項(term)で表現することを仮定し、関係の属性値を定数だけでなく、項に拡張した項関係の形で項集合を２次記憶に格納する。このため、２次記憶に格納された項の検索の高速化が必要となる。項の検索に対しては、従来の関係演算に単一化の概念を加えたＲＢＵ演算(Retrieval By Unification)［横田85］を用いる。さらに、ＲＢＵ演算を高速に実行する専用装置を複数用いて、検索の高速化を図る［酒井86］［柴山86］。この専用装置を単一化エンジンと呼ぶ。　本稿では、第２章で、単一化エンジンの構成方式を述べ、第３章で、ソフトウェア・シミュレータにより、幾つかのデータを用いてエンジンの基礎的な性能について述べる。また、第４章では、評価結果を基に、処理高速化の方式を提案し、その性能を評価する。

## 2．単一化エンジンの構成方式［森田86a ］

単一化エンジン( ＵＥ:Unification Engine)の構成を図１．に示す。エンジンの入力は、項のストリング表現、出力は、単一化結合に成功した結果のタプルのストリング表現である。ＵＥは、大きくソータ （ ＳＵ:Sort Unit）、ペア生成部( ＰＧＵ:Pair Generation Unit)、及び単一化処理部( ＵＮＵ:Unification Unit)に分類される。これら各部、及び各構成要素の機能について以下のようになっている。

タプル記憶部：入力したタプルの格納場所

前処理部　　：項関係ｋｂ１（Ａ，Ｂ）とｋｂ２（Ｃ，Ｄ）の間で、ジョイン対象属性をＢ、Ｃとし、結果として属性＜Ａ，Ｂ，Ｄ＞からなるタプルの出力を行う単一化結合では、タプルのＢ属性値、Ｃ属性値のみをソータへ送り、出力属性値Ａ、Ｂ、Ｄをタプル記憶部へ送る。

ソートセル　：2way-merge-sort 法を用い、項をオーダリングにより、ジェネラリティ順に並べ変える。この場合、入力される項は可変長の文字列で表現される。また、Ｔｒｉｅ化により、ソートセルの流量を減らす。

ペア生成部　：オーダリングにより整列された２つの項の記号列を受け取り、いずれかの項の変数の前までの２項の記号列の一致性をチェックする。一致した項組を単一化成功の可能性のあるペアとして、単一化処理部へ送る。一方、ペアに対応するタプルの、タプル記憶部上の格納番地を出力生成部へ送る。

出力生成部　：タプル記憶部を参照して、単一化結合の結果として出力するタプルを生成する。

単一化セル　：記号列の食い違いを検出し、ペアの１つの変数に対する置換を求め、置換えセルへ送る。１つの食い違いの検出を、単一化セル１つで行い、残りの処理は、次段セル以降で行うパイプラインである。

置換えセル　：単一化セルで求まったひとつの変数に対する置換を、対応する（出力指定属性からなる）タプルに適用する。置換が求まる毎に随時適用を行うためこの単一化処理方式を置換随時適用方式（ＳＡＭ：Sabstitution Apply Method ）と呼ぶ。

後処理部　　：データを標準出力形式に変換する。

全体で見ると、これら各部がパイプラインを行っている。

以上の構成をＣ言語を用いて作成されたプログラムを使用してシミュレートし、各セル間のデータの流れる時間を測定することにより単一化エンジンの性能を評価する［酒井86］［小黒86］。なお、本評価では、１回の１Word (4byte) のデータに対する操作(read/ write) に要する時間を１クロックとする。

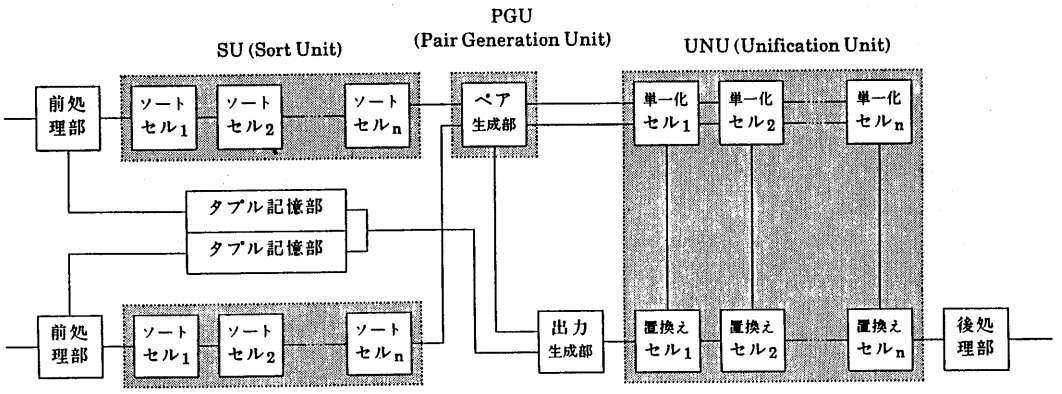


図１．ＵＥの構成図

## 3．単一化エンジンの性能

今回は以下のようなデータを用いて、ＲＢＵ演算の中で最も処理が重い単一化結合演算時の性能を評価した。

Ⅰ．網羅的なデータ

項を木で表現した時の木の形と各レベルに出現しうる関数子を定めたときに、その可能なすべての項の集合。

単一化エンジンは入力データや出力データの量や単一化可能な項の数、ペア生成部で削減可能な候補の数など様々な要因の影響を受けると思われる。網羅的なデータはパラメータも多く様々な項集合を簡単に生成できる。今回は、以下のようなデータを用意した。

（１）木構造を一定にし、図２．に示す各ノードに現われる関数子の種類数を変える。

①根ノードの関数子のみの種類数変化(PARA1)

②中間ノードの関数子のみの種類数変化(PARA2)

③葉ノードの関数子のみの種類数変化(PARA3)

（２）木構造を変える。

根、葉からなる木構造において、根ノードの関数子がとる引数の数を変化(PARA4)

本評価では、主にこの４つのパラメータ別に測定を行う。但し、今回は、簡単のために、属性数１の項関係として測定を行った。

Ⅱ．ＤＣＫＲサンプルデータ

ＤＣＫＲ［田中86］で表現された知識を項関係の形に書き直したもの。

ＤＣＫＲで表現された知識を項関係に書き直したもののＲＢＵ演算を用いた知識検索については［村上86］を参照。今回用いたサンプルデータは、属性数２タプル数８１の項関係で表現されたis aとhas a を含んだ知識である。

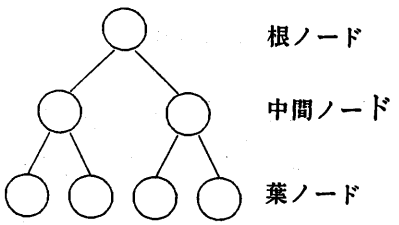


図２．項の木表現

### ３．１　ＵＥの処理概要

各部の処理の全体処理への影響を見る。単一化エンジンの主な処理ブロックは、ＳＵ、ＰＧＵ、ＵＮＵであり、全体でパイプラインを行う。このため、処理負荷が一番重い部分が全体の処理を決めると考えられる。図３．に、網羅的なデータ(PARA1) の場合の各部の処理量と単一化エンジン全体の処理量の関係を示す。図３．より、単一化エンジン全体の処理量が、ＵＮＵの処理量にほぼ一致していることが分かる。これは、他の網羅的なデータに対しても同様である。一方、ＵＮＵの処理を決める要因である入力量は、ＰＧＵの出力量である。ＰＧＵの出力量は、ＰＧＵの単一化失敗項組の削減機能（以降、候補削減能力と呼ぶ）により定まる。網羅的なデータに対してはＰＧＵの候補削減能力が低いため、ＳＵでの項の整列やペアの生成に要する処理は、ＵＮＵでの単一化処理より軽くなっているからである。

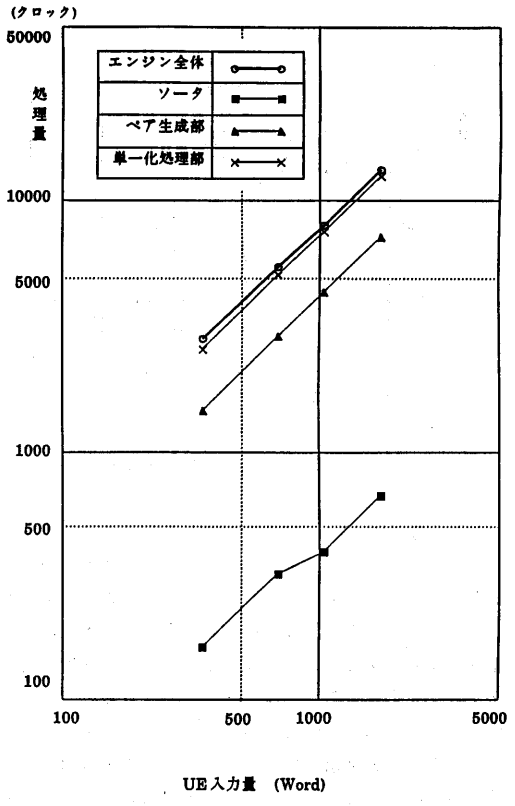


図３．各部の処理量(PARA1)

一方ＤＣＫＲの例では、ＳＵ，ＰＧＵ，ＵＮＵの処理時間は、それぞれ全体処理時間の60%, 90%, 20～50% になっている。これは、逆に、ＰＧＵの候補削減能力が高いため、ＵＮＵの処理負荷が軽くなっているためである。

以上より、エンジン性能は、ＰＧＵの候補削減能力とＵＮＵの単一化処理能力に左右される。以下、各部の性能を評価する。

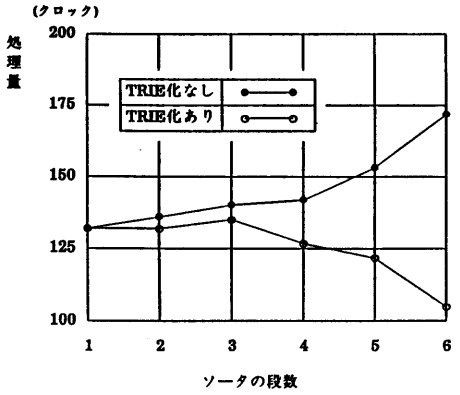


図4．Ｔｒｉｅ化の効果

### ３．２　ソート処理

ソート処理は、パラメータによらず、入力量にのみ影響を受ける。このため、前述PARA1 ～PARA4 のパラメータは無視して、ソータの評価を行う。図３．で、入力量と処理量が完全に比例関係にならないのは、Ｔｒｉｅ化の効果である。Ｔｒｉｅ化の効果が大きいほど、セルに流れる量は少なくなり、処理が軽くなる。網羅的なデータでは、一般的に、関数子の種類が増えると、Ｔｒｉｅ化の効果が大きくなる。図４．に、Ｔｒｉｅ化の効果例を示す。Ｔｒｉｅ化を行うと、ソートセルの後段に行くに従い処理が軽くなっている。2way-merge-sort では、後段に行くに従い、メモリの容量が増える（$2^n$ タプル分　ｎ：段数）ため、Ｔｒｉｅ化を行わないと、図４．に示すように、項列をメモリにセットするまでの時間が多くかかる。

### ３．３　ペア生成処理

ペア生成部の処理は、パラメータPARA1 ～PARA4 により傾向が異なっているため、別々に評価を行う。

#### ３．３．１　ペア生成部の出力と処理量

網羅的データを使用した例では、ペア生成部の出力量は、入力量より大きくなる。このため、ペア生成部の出力量のペア生成処理への影響を考える。図５．に、ペア生成部処理量／出力量関係を示す。図５．から明らかなように、処理量は、出力量と比例関係にある。従って、出力量が入力量より十分大きければ、出力処理がペア生成部の処理時間となりペア生成の処理は出力処理に隠れる。これは、ペアを生成する処理は断続的であり一度に複数組を生成すため、一度に複数生成されたペアを流す間に次のペアの集合を生成してしまうからである。

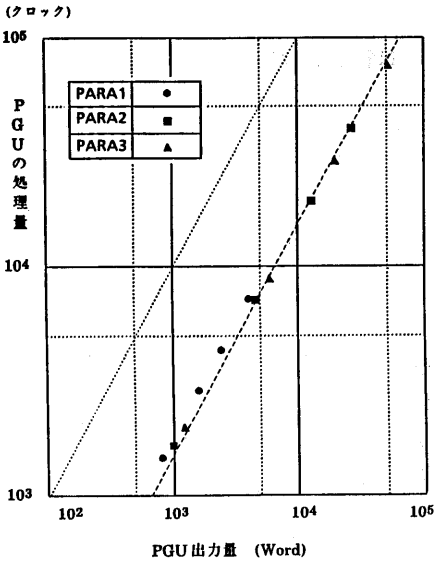


図５．ペア生成部処理量／出力量

#### ３．３．２　候補削減能力

ＰＧＵの性能として、明らかにすべき候補削減能力について評価する。表１．にＰＧＵ出力項組数／単一化成功項組数特性を示す。また、表２．には、エンジン入力に含まれる単一化失敗項組をＰＧＵでふるい落とした割り合いを示す（削減率）。木構造が複雑になるか、または木構造の中間ノードや葉ノードの関数子の種類数が多く存在する程、単一化成功項組数に対して、ＰＧＵの出力項組数が多くなる。一方、削減率についても同様であり、このような場合

表１．候補削減能力（１）
ＰＧＵ出力項数／単一化成功項組数

| 種類数が変化するノード | 関数子の種類数 | | |
|---|---|---|---|
| | 1 | 2 | 3 |
| 根 | 1.08 | 1.08 | 1.08 |
| 中間 | 1.08 | 1.86 | 2.63 |
| 葉 | 1.08 | 1.69 | 2.73 |

(a) 木構造を一定にした場合

木構造は、深さ２
根ノードがｎ引数関数１種
葉ノードが定数３種と変数

| n | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| | 1.00 | 1.35 | 2.08 | 3.13 |

(b) 木構造を変化させた場合

表２．候補削減能力（２）

削減率

（％）

| 種類数が変化するノード | 関数子の種類数 | | |
|---|---|---|---|
| | 1 | 2 | 3 |
| 根 | 93.0 | 97.5 | 98.5 |
| 中間 | 93.0 | 84.0 | 86.1 |
| 葉 | 93.0 | 75.3 | 72.8 |

(a) 木構造を一定にした場合

木構造は、深さ２
根ノードがｎ引数関数１種
葉ノードが定数３種と変数　　（％）

| n | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| | 100.0 | 88.6 | 79.4 | 73.2 |

(b) 木構造を変化させた場合

に、ＰＧＵの候補削減能力が低下し、エンジン全体の性能が低下する。

ＤＣＫＲの例では、エンジンに入力された項集合中に、単一化可能項組が含まれる割り合いが、網羅的データに比べて小さい。このため、ＰＧＵ出力項組数／単一化成功項組数特性は網羅的なデータの場合それほど大差はない。しかし、削減率（ＰＧＵによりふるい落とした単一化失敗項組／ＵＥ入力内の単一化失敗項組）は、測定データの最悪値が約９９．１％であり、削減率は、網羅的データの場合に比べて良い。

候補削減能力を向上させるためにはハッシング［大森86］やスーパーインポーズドコード［森田86b ］などを用いる方法などが考えられる。これらは、複数台のＵＥを並列に実行させるときの問題分割のとき用いることができ、その方式については現在検討中である。

## ３．４　単一化処理部

### ３．４．１　パイプライン処理

ＵＮＵの各エレメントの処理は、パイプラインを行っている。このパイプラインの効果を、UNION-FINDメモリを用いて、１組の項の単一化の高速化を行った［安浦84］と処理クロックを比較して、表３．に示す。一組の項の単一化では、本方式は処理が遅くなるが、複数組の項の単一化には、パイプラインが効果的である。

### ３．４．２　単一化処理部と入力量

ＵＮＵに対する入力は、単一化セルに送られるジョイン対象属性と置換えセルに対する出力指定属性の２つがある。単一化処理部の置換えセルへの入力量とＵＮＵの処理量の関係を図６．に示す。図６．より、処理クロックは、置換えセルへの入力量にほぼ比例することが分かる。網羅的なデータに対しては、単一化セルに流れる量より置換えセルを流れる量が多く、単一化セルの処理量が隠れている。一方、置換えセル及び単一化セルの入力量は、ＰＧＵの候補削減能力で決まるため、ＰＧＵの候補削減能力が、前述のように低下すると処理に時間を要することになる。ＰＧＵの候補削減能力が低下するとは、ＵＮＵ入力中に、単一化失敗項組が多く含まれている事である。このため、ＰＧＵの候補削減能力低下に対して単一化処理を高速に実行するために、以下のような手法があげられる。

①単一化処理を高速に実行できる新アルゴリズムの提案
②失敗組に対する単一化処理の省力化
③入力属性タプルへの変数に対する適用の高速化

今回は、②による高速化の検討を行った。単一化処理の高速化の考察を次章で述べる。

表３．パイプラインの効果

t1=f(a1,a2,...,ak)とt2=f(X1,X2,...,Xk)の単一化

| k | 単一化対象項数 | | | | |
|---|---|---|---|---|---|
| | パイプライン | | | | 文献３ |
| | １：１ | ５：５ | 20：20 | 50：50 | １：１ |
| 1 | １８ | ９ | ８ | ６ | １０ |
| 2 | ２１ | １２ | ９ | ８ | １３ |
| 10 | ５６ | ３７ | ２７ | ２６ | ３７ |
| 25 | １３１ | ７６ | ５６ | － | ８２ |

単位：クロック／項組

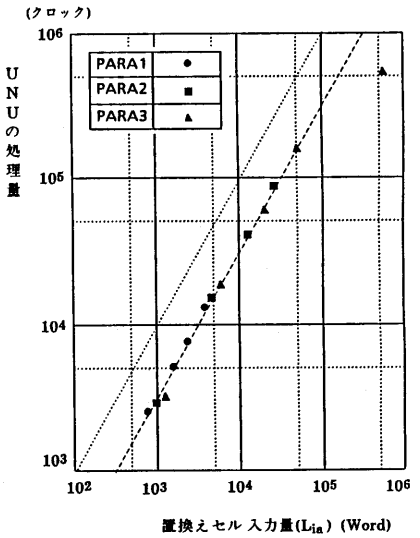


図６．ＵＮＵ処理量／入力量

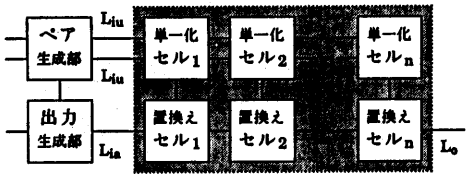


図7．置換随時適用方式（SAM）

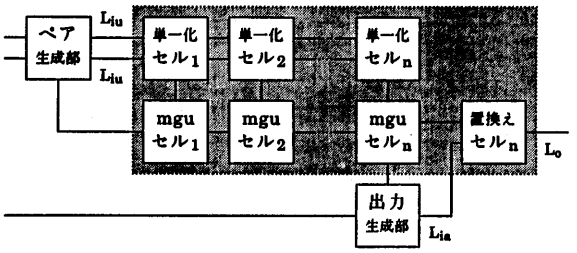


図8．mgu適用方式（MAM）

## 4．単一化処理部の考察

### 4．1　原理

置換随時適用方式（SAM）は、以下のような特徴を持つ（図7．）。

①単一化セルには、単一化結合対象属性が流れ、変数に対する置換を随時求める。各セルでは、項間に食い違いが検出されるまで、記号列を検査し、検出すれば、まだ検査されていない部分のみを、次段以降に送るため、各セルの入力量は、後段に行くほど減少する。

②置換えセルには、出力指定属性からなるタプルが流れ、変数に対する置換の適用が単一化セルで求まるとすぐ適用を受ける。後段に行くほど、置換の適用を多く受けたタプルが流れるため、各セルの入力量は増加する。

SAMの問題点として、以下があげられる。

タプルへの置換の適用は、随時行われるため、m段目（m＜n）でペアの単一化失敗が判明したときに、m－1段目までに求まった置換の適用は既に行われており、置換えセルに無駄な量が流れていることになる。3章で述べたように、単一化処理は、置換えセルに流れる量で決まるため、無駄な量が存在することは問題である。

この問題を除去するために、図8．に示すmgu適用方式（MAM：Most-general-unifier Apply Method ）についての考察を行う。MAMには、以下の特徴がある。

(1) 単一化セルは、SAMと同様の機能を持つ。

(2) mguセルでは、単一化セルで求まった変数に対する置換を集合にして流す。このとき、流れてきた置換集合の中に、置換を適用すべき変数が含まれている場合は、置換の適用を行う。例えば、流れてきた置換集合が$\theta$＝｛f（a）／X，Z／Y｝であるとき、単一化セルで｛a／Z｝なる置換が求まれば、Zに対する置換の適用を受けて、$\theta'$＝｛f（a）／X，a／Y，a／Z｝が出力される。mguセルの最終段出力は、ペアに対するmgu（最汎化作用素）である。

(3) 置換えセルでは、ペアの単一化が終わった後に、単一化成功（mguが求まった）ペアに対してのみ、mguの適用を行う。置換えセルには、単一化成功ペアに対する出力属性からなるタプルのみ流すため、置換えセルの処理量に無駄な量は含まれない。

MAMは、以上のように、UNUの入力に単一化失敗ペアが多く存在する場合には、非常に効果的であると思われる。次節でSAMとMAMの比較をシミュレータにより行う。

### 4．2　実験結果

SAM、MAMの比較を次の条件の下で行う。

①入力されるペアの種類、数を一定にする。

②入力ペアに含まれる単一化失敗項組の数を変化させる。このときのUNUの処理量／出力量の関係を測定する。

測定結果を図9．に示す。UNUへの入力量は、PGUから単一化セルへ（$L_{iu}$）は1728×2Word一定で、出力生成部から置換えセルへ（$L_{ia}$）はSAMでは3265Word一定で、MAMでは単一化失敗項組の数による。点線は、SAMで、単一化失敗項組が無い場合のUNUの処理量／出力量関係である（この場合は入力量を変化させている）。図9．より以下が観察される。

Ⅰ．UNUの処理量は、出力量による。点Pの出力データ長以上では、2方式の処理量は、ほぼ一致し、それは出力量による。

Ⅱ．失敗項組が多いとき、両方式とも、出力量によらず、UNUの処理量が一定になり、しかも、MAMのそれは、SAMのそれより小さい。

これらより、次のことが考えられる。

Ⅰは、失敗組が少ないとき、UNUの出力量（$L_0$）が、他のどのセルを流れる量よりも多くなり、処理時間が$L_0$で決まるからである。また、失敗組が多いとき、SAMでは、$L_0$よりも、最終段の置換えセルの入力量、MAMでは、$L_0$よりも、最終段のmguセルから出るmguの量の方が多くなり、処理時間がそれぞれの量で決まる。Ⅱは、これらの量が、UNUの入力一定の時、ほぼ一定の量になるためである。

さらに、一般的に置換えセルに流れるタプルの量は、mguセルを流れる置換集合の量よりも大きい。

以上より、置換集合の量及びジョイン対象属性の量が出

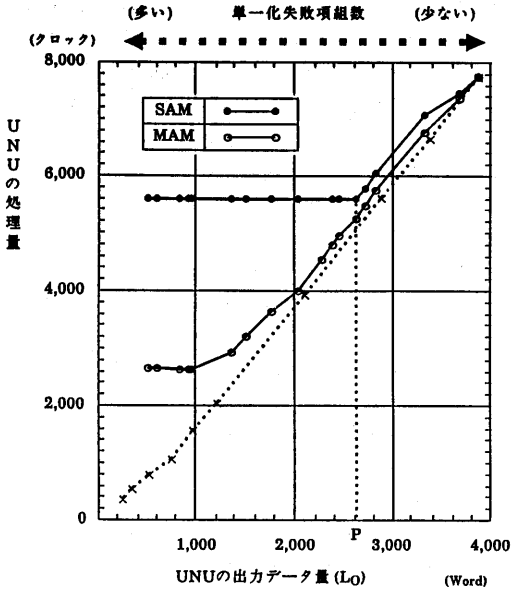


図９．単一化失敗項組の数による両方式の違い

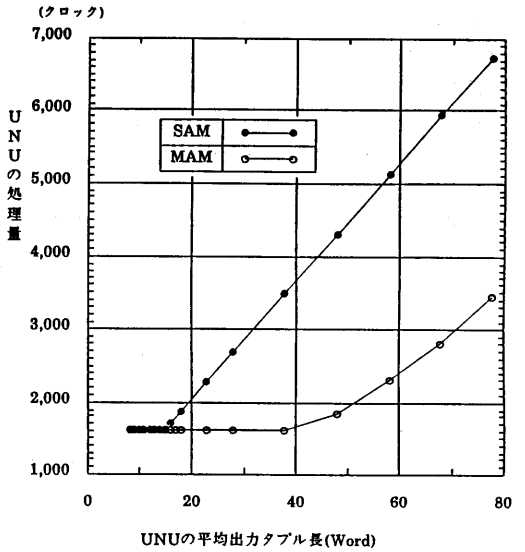


図１０．出力タプルの長さによる両方式の違い

力指定属性の量より小さいことを前提にすると、ＵＮＵ入力に失敗項が多く含まれる場合は、ＵＮＵの処理は、ＳＡＭよりもＭＡＭが優れる。網羅的なデータに対しては、ＵＮＵ入力中に単一化失敗項組を含む率は、30～60％程度であるため、ＭＡＭを用いたほうが処理が速くなる。

次にＤＣＫＲの例に対してこの２つの方式を比較してみる。ＤＣＫＲの例では、削減率は高いが、ＰＧＵ出力項組中に含まれる単一化失敗項組の数は多い。このため、これまでの検討から、単一化処理方式としてＭＡＭが、優れていることが予想される。しかし、その結果単一化処理のクロックは、ＭＡＭでは１６２９クロック、ＳＡＭでは１６２８クロックとなり、方式による違いが顕著にみられなかった。これは、$L_{iu}$のほうが$L_{ia}$より大きく、ｍｇｕの計算処理のほうがｍｇｕの適用処理よりも負荷が重くなったためである。（ＤＣＫＲの例ではジョイン対象属性を出力属性として指定していない。）ＭＡＭの方が処理時間が長いのは、４．１の(3) により、置換え処理をｍｇｕの計算と並列して行えなくなったためである。しかし、一つの変数に対する置換えは、置換１つでもｍｇｕでも同じ時間で行なえるため１クロックの遅延で済んでいる。逆に、出力属性を増やすなどして$L_{iu}$一定で$L_{ia}$の量を増加させると図１０．のようにＭＡＭの方が処理時間が短くなる。ただし、図１０．で横軸は$L_{ia}$がＭＡＭで単一化失敗組数によるため平均出力タプルを用いており、$L_{iu}$の平均タプル長は１３．７wordである。

## ５．そのほかの改良点とその評価

評価を行う過程で各処理部の動作を細かく検討した結果以下のような点を改良した。

（１）ペア生成部

・ペア生成部内部の並列動作

ペア生成部では関数記号スタックと項スタックが用いられている［森田86a ］。これらのアクセスに関して同時に行えるものは同一クロック内に処理できるようにした。

・ペア生成部の終了処理

処理の終了条件を一方のスタックが空になったときとした。これは、ＤＣＫＲの例のように項をソートしたとき単一化可能な項の集合が一部に固まって現れる場合に効果的である。

（２）単一化処理部

ｍｇｕ適用方式ではｍｇｕが求まってから適用すべきタプルを項記憶部から取り出し適用を行うまで次のｍｇｕが受け取れず、結果として単一化セルが停止するようになっていた。そこで、置換えセルにバッファを持たせこれを改善した。

これらの結果（新ＭＡＭ）と改良前の結果（ＳＡＭ）との比較を図１１．に示す。全体処理時間及び単一化処理部の処理時間が約６５％単一化処理部の処理時間が約７８％短縮されている。ソータに関しては、改良を行っていないので変化が無い。

## 6．おわりに

ＰＧＵでは、項の構造が複雑になると候補削減機能が低下し、単一化失敗項組が多く残るため、ＵＮＵに、置換随時適用方式を採用すると性能が低下し、エンジン全体の性能低下につながる。このため、ＵＮＵに、ｍｇｕ適用方式を採用することで、単一化失敗項組が多く含まれるデータに対するＰＧＵの候補削減能力低下を補償できた。

一方そのほかの改良により網羅的なデータに対してＵＥの処理速度は約６５％向上した。

今後の課題には、シミュレーション解析を反映して、単一化エンジンの詳細設計を行い、クラスタリングを導入した複数ＵＥの並列処理方式の検討を行いたい。


## 謝辞

評価データをまとめるに当り御協力頂いた日科技研の白瀬勝次氏、高橋正寿氏に感謝します。さらに、熱心に議論して頂いた第３研究室の諸氏、及びVLKB会議メンバに感謝致します。


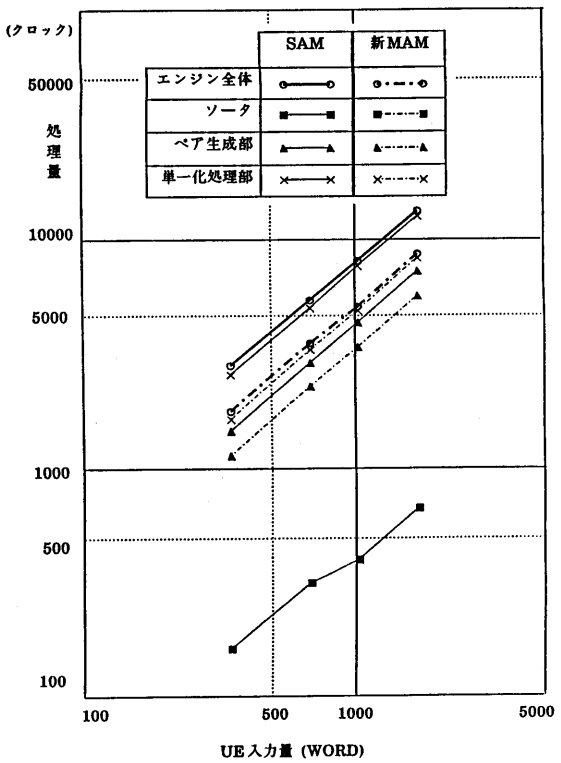


図１１．改良前後の処理時間（PARA1）

## 参考文献


［伊藤　86a ］Itoh, H, "Resarch and development on knowledge base systems at ICOT",12th Int. Conf. on VLDB, pp.437-445, August, 1986

［伊藤　86b ］伊藤他「大規模知識ベースマシン実験機の開発（１）」第３３回情処全大(1986) 3B-1

［大森　86］大森他，「推論機能と関係データベースの融合方式」信学技報 AI86-21

［小黒　86］小黒他，「大規模知識ベースマシン実験機の開発（４）」第３３回情処全大(1986) 3B-4

［酒井　86］酒井他，「大規模知識ベースマシン実験機の開発（３）」第３３回情処全大(1986) 3B-3

［柴山　86］柴山他，「大規模知識ベースマシン実験機の開発（２）」第３３回情処全大(1986) 3B-2

［田中　85］田中他，「Definite Clause Knowledge Representation」, The Logic Programming Conference 85, 1985

［村上　86］村上他，「単一化検索言語による知識ベースソフトウェアの記述」第３２回情処全大(1986) 1M-9

［森田　86a ］Morita, Y., et al, "Retrieval-BY-Unification on a Relational Knowledge Base Model",Proc. 12th Int. Conf. on VLDB, pp.52-59, August, 1986

［森田　86b ］森田他，「スーパーインポーズドコードを用いた構造体の検索方式」第３３回情処全大(1986) 6L-8

［横田　86］Yokota, H., et al, "A Model and Architecture for a Relational Knowledge Base",13th Int. Sym. on Computer Architecture, pp.2-9, June, 1986

［安浦　86］安浦他，「論理型言語の単一化操作のためのハードウェアアルゴリズム」信学技報 EC84-67