

TR-225

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

小黒雅己 (NTT)
森田幸伯, 伊藤英則

January, 1986

©1987, 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

知識ベースマシンにおける 単一化専用装置の処理方式とその評価

森 田 壽 伯, 小 黒 雄 三*, 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とっているため、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 Iton.

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、知識を意味を含む論理的構造体である項(terml)で表現することを規定し、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、従来の関係演算に統一化の概念を加えたRBI演算(Retrieval by Unification)〔渡田85〕を用いる。さらに、RBI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〔堀井86〕〔黒田86〕。この専用装置を統一化エンジンと呼ぶ。本稿では、第2章で、統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. 統一化エンジンの構成方式 黒田86a

統一化エンジン(以下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、統一化結合に成功した結果のタブ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(PG:Pair Generation Unit)、及び統一化処理部(UU: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。

タブル記憶部：入力したタブルの格納場所

前処理部：項関係Kの1(A, B)とKの2(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>からなるタブルの出力を行う統一化結合では、タブルの属性値、C属性値

のみをソータへ送り、出力属性値A, B, Dをタブル記憶部へ送る。

ソータセル：2way-merge-sort法を用い、項をオーダリングにより、ジェネラリティ項に並べ変える。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、Ternary化により、ソータ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。一致した項給を統一化成功の可能性のあるペアとして、統一化処理部へ送る。一方、ペアに対応するタブルの、タブル記憶部上の格納番地を出力生成部へ送る。

出力生成部：タブル記憶部を参照して、統一化結果の結果として出力するタブルを生成する。

統一化セル：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、統一化セル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：統一化セルで求め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)タブルに適用する。置換が求まる毎に高速適用を行うためこの統一化処理方式を置換適用方式(SAM: Substitution 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、各セル毎のデータの流れる確率を測定することにより統一化エンジンの性能を評価する〔堀井86〕〔黒田86〕。なお、本評価では、1回の1Word(4byte)のデータに対する読取(read/write)に要する時間を1クロックとする。

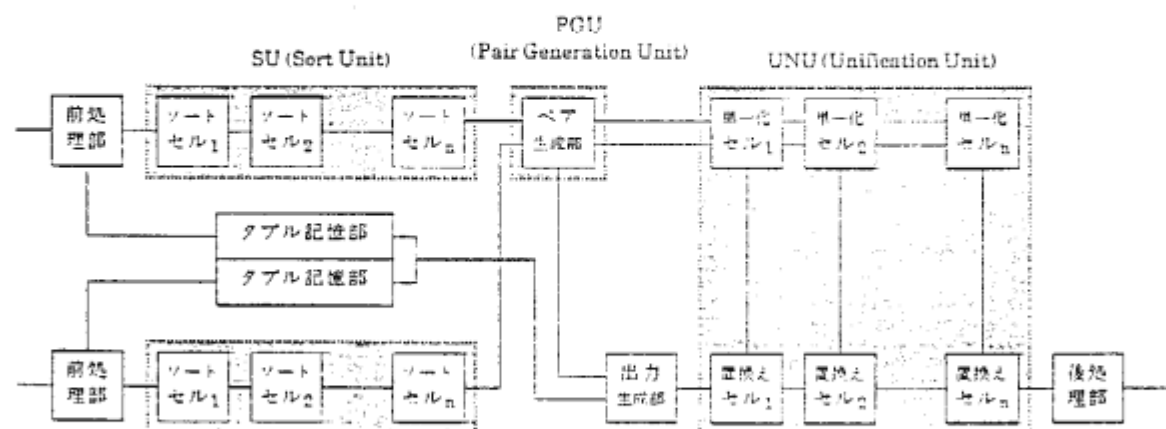


図1. UEの構成図

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する項数を定めたときに、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項数の種類数を変える。

- ①根ノードの項数のみの種類数変化(PARA1)
- ②中間ノードの項数のみの種類数変化(PARA2)
- ③葉ノードの項数のみの種類数変化(PARA3)

(2) 木構造を変える。

根、葉からなる木構造において、根ノードの項数が増える引数の数を変化(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、特性数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、特性数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。

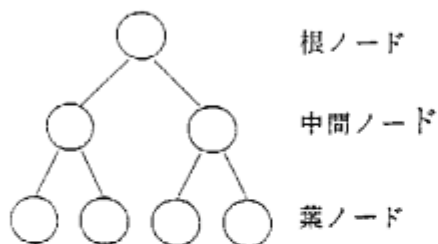


図2. 項の木表現

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

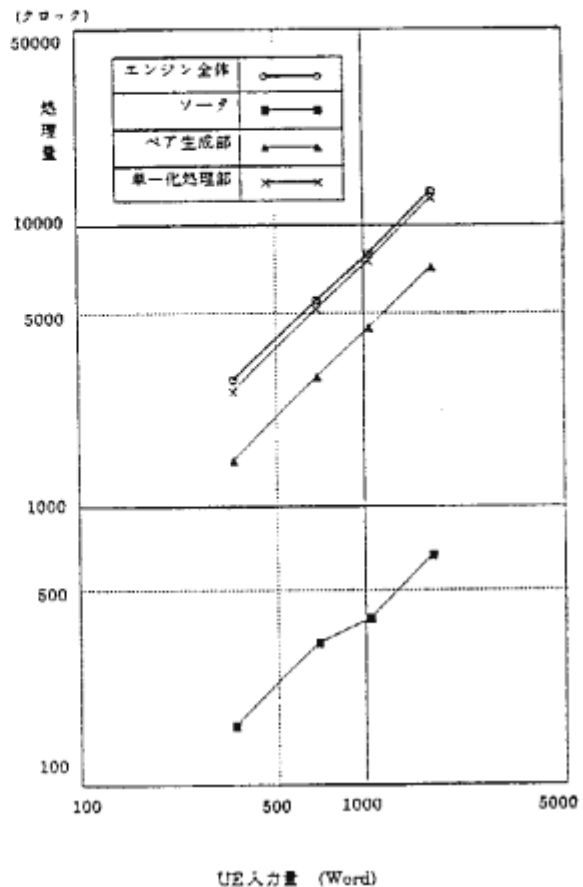


図3. 各部の処理量 (PARA1)

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の処理量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の単一化失敗候補の削減能力（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。

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。

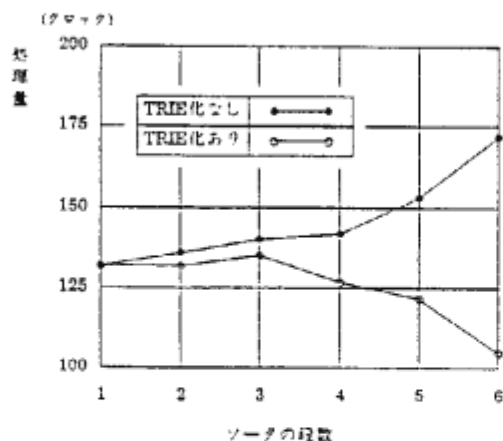


図4. Trie化の効果

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化の効果が大いほど、セルに流れる量は少なくなり、処理が軽くなる。綺麗なデータでは、一般時に、葉ノードの種類が増えると、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なる。

3. 3 ベア生成処理

ベア生成部の処理は、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 ベア生成部の出力と処理量

綺麗なデータを使用した例では、ベア生成部の出力量は、入力量より大きくなる。このため、ベア生成部の出力量のベア生成処理への影響を考える。図5. に、ベア生成部処理量/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、出力処理がベア生成部の処理時間となりベア生成の処理は出力処理に遅れる。これは、ベアを生成する処理は逐次的であり一度に複数組を生成するため、一度に複数生成されたベアを流す間に次のベアの生成を生成してしまうからである。

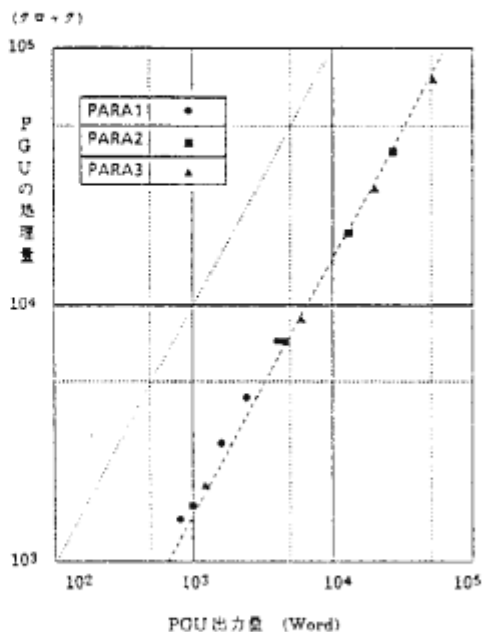


図5. ベア生成部処理量/出力量

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出力項数/単一化成功項数削減率を示す。また、表2. には、エンジン入力に基く単一化失敗項数をPGUでふるい落とした割合を示す(削減率)。本構建が理想になるか、または本構建の中間ノードや葉ノードの葉ノードの種類数が多く存在する程、単一化成功項数に対して、PGUの出力項数が多くなる。一万、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(%)

PGU出力項数/単一化成功項数

種類数が変化 するノード	葉ノードの種類数		
	1	2	3
根	1.08	1.08	1.08
中間	1.08	1.86	2.63
葉	1.08	1.69	2.70

(a) 本構建を一定にした場合

本構建は、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
削減率	1.00	1.35	2.08	3.13

(b) 本構建を変化させた場合

表2. 候補削減能力 (2)

削減率

種類数が変化 するノード	削減率の種別数 (%)		
	1	2	3
根	93.0	97.3	98.5
中間	93.0	84.0	86.1
葉	93.0	75.3	72.8

(a) 不構造を一定にした場合

不構造は、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
削減率 (%)	100.0	88.6	79.1	73.2

(b) 不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。

DOCKR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合が、構造的データに比べて小さい。このため、PGU出力項集合が単一化成功項組の割合は構造的データの場合それほど大差はない。しかし、削減率 (PGUによりふるい落とされた単一化失敗項組/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、構造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を用いることができ、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 バイプライン処理

U/Uの各要素の処理は、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-FIXメモリを用いて、1組の項の単一化の高速化を行った [安藤84] と処理クロックを比較して、図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

U/Uに対する入力量は、単一化セルに送られるジョイン対象項性と置換えセルに対する出力指定項性の2つがある。単一化処理部の置換えセルへの入力値とU/Uの処理部の関係を図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。構造的な

表3. バイプラインの効果

 $t1=f(a1,a2,\dots,ak)$ と $t2=f(x1,x2,\dots,xk)$ 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

データに対しては、単一化セルに送れる量より置換えセルを送れる量が多く、単一化セルの処理量が溢れている。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、U/U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 失敗項に対する単一化処理の省力化

③ 入力属性テーブルへの変数に対する運用の高速化

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有効性を次章で述べる。

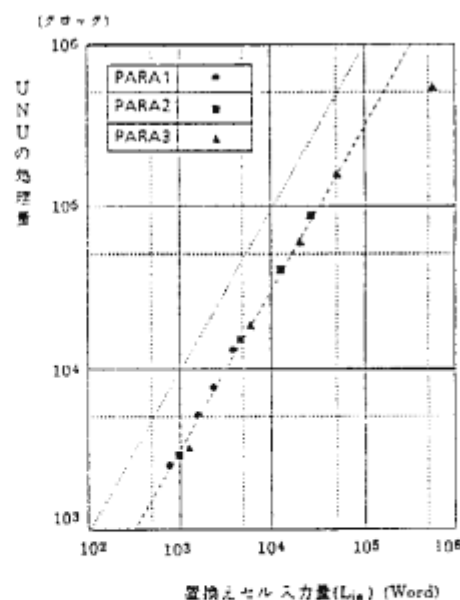


図6. U/U処理量、入力量

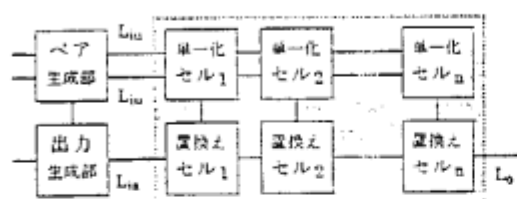


図7. 置換順時適用方式 (SAM)

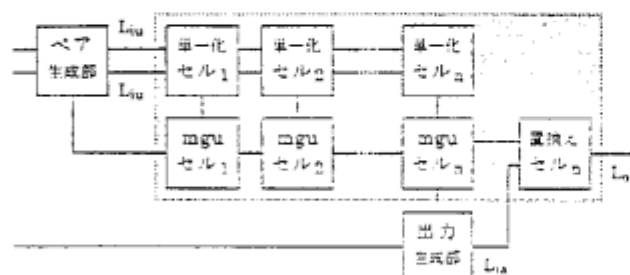


図8. mgu適用方式 (MAM)

4. 単一化処理部の概要

4.1 原理

置換順時適用方式 (SAM) は、以下のような手順を踏む (図7.)。

①単一化セルには、単一化組合対象属性が流れ、変数に対する置換を逐次求める。各セルでは、順番に古い値が検出されるまで、置換列を探索し、検出すれば、まだ探索されていない部分のみを、次段以降に進めるため、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め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値が流れていることになる。3 段で述べたように、単一化処理は、置換えセルに流れる量で決まるため、無駄な値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機会にして流す。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、2 段の適用を行う。例えば、流れてきた置換集合が $\theta = \{x/a\}$ ($X, Z \neq Y$) 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、 $\theta' = \{x/a\} \cup \{X, a/Y, a/Z\}$ 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、ペアの単一化が完了した後に、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値は含まれない。

MAMは、以上のように、し\じ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期待される。次段で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。
- ③このときのし\じの処理量と出力量の異同を測定する。

測定結果を図9. に示す。し\じへの入力量は、PGU から単一化セルへ (L_{1u}) は 1728×240 であり、出力生成部から置換えセルへ (L_{1a}) はSAMでは3265 Here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し\じの処理量・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. し\じの処理量は、出力量による。点Pの出力データ量以上では、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、し\じ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、失敗項組が少ないとき、し\じの出力量 (L_o) が、他のどのセルを流れる量よりも多くなり、処理時間 L_o で決まるからである。また、失敗項組が多いとき、SAMでは、 L_o よりも、最終段の置換えセルの入力量、MAMでは、 L_o よりも、最終段のmguセルから出るmguの値の方が多くなり、処理時間がそれぞれの値で決まる。II は、これらの量が、し\じの入力一定の時、ほぼ一定の値になるためである。

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

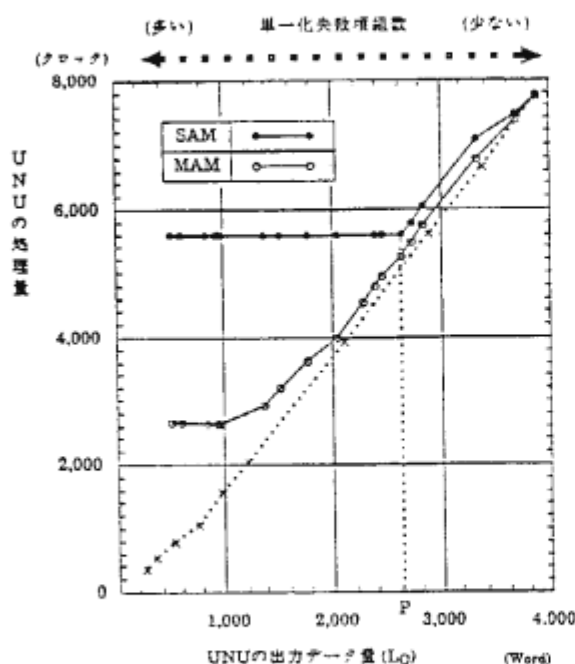


図9. 単一化失敗項数の数による両方式の違い

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が速れる。稀薄的なデータに対しては、UNU入力中に単一化失敗項を含む率は、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DOCKRの例に対してこの2つの方式を比較してみる。DOCKRの例では、割減率は高いが、PGU出力項組に含まれる単一化失敗項の数は多い。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、 mg_u の計算処理のほうが mg_u の置換処理よりも負荷が大きくなったためである。(DOCKR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 mg_u の計算と並列して行えなくなったためである。しかし、一つの要素に対する置換えは、置換え1つでも mg_u で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で単一化失敗項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。

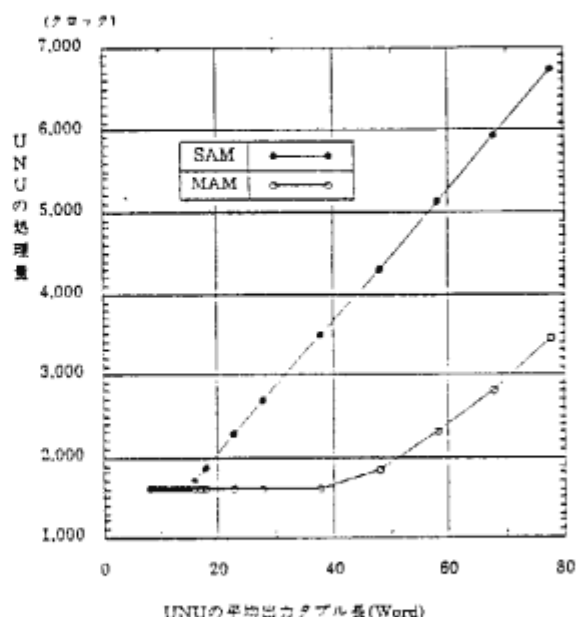


図10. 出力タプルの長さによる両方式の違い

5. そのほかの改善点とその評価

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

ペア生成部では簡記符号スタックと項スタックが用いられている〔森田88a〕。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、DOCKR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留まって残れる場合に効率的である。

(2) 単一化処理部

mg_u 適用方式では mg_u 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 mg_u 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6.5%単一化処理部の処理時間が約7.8%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

P GUでは、項の構造が複雑になると誤推測認識能力が低下し、単一化失敗項が多く出るため、しししに、意味認識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、しししに、m g u適用方式を採用することで、単一化失敗項が多く含まれるデータに対するしししの誤推測認識能力低下を補償できた。

一方そのほかの改良により複雑なデータに対してししの処理速度は約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しし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田端勝次氏、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VLK3会議メンバに感謝致します。

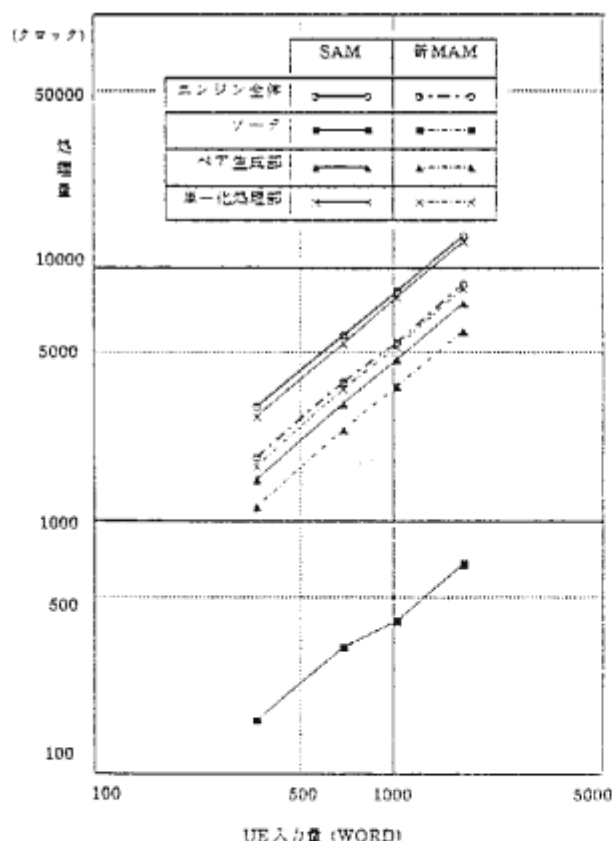


図 1.1. 改良前後の処理時間 (PAR1)

参考文献

- [伊藤 85a] 伊藤, H., "Research and development on knowledge base systems at ICOT", 12th Int. Conf. on VLDB, pp.437-445, August, 1986
- [伊藤 86a] 伊藤他, "大規模知識ベースマシン実験機の開発(1)", 第33回情報学会(1986) 36-1
- [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回情報学会(1986) 38-4
- [福井 86] 福井他, "大規模知識ベースマシン実験機の開発(3)", 第33回情報学会(1986) 36-3
- [栗山 86] 栗山他, "大規模知識ベースマシン実験機の開発(2)", 第33回情報学会(1986) 38-2
- [田中 85] 田中他, "Definite Clause Knowledge Representation", The Logic Programming Conference 85, 1985
- [村上 86] 村上他, "単一化演算言語による知識ベースソフトウェアの記述", 第32回情報学会(1986) 38-9
- [森田 86a] Morita, Y., et al., "Retrieval-DB-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文木の検索方式", 第33回情報学会(1986) 36-8
- [横田 86] Yokota, H., et al., "A Model and Architecture for a Relational Knowledge Base", 10th Int. Sym. on Computer Architecture, pp.2-9, June, 1986
- [安部 86] 安部他, "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