

TR-112

論理型言語 ESP のプログラム特性評価

山本 明, 三井正樹, 吉田裕之 (沖電気)

中島克人 (三菱電機)

横田 実 (ICOT)

July, 1985

©1985, 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

論理型言語 ESP のプログラム特性評価

山本 明* 三井 正樹* 吉田 裕之* 横田 実** 中島 克人***
* (沖電気) ** (新世代コンピュータ技術開発機構) *** (三菱電機)

This paper describes static and dynamic characteristics of programs written in ESP (Extended Self-contained Prolog) running on PSI (Personal Sequential Inference machine). All the software modules on PSI are written in ESP. Four actual programs are chosen to measure the numbers of arguments, alternatives, the ratios of built-in predicate calls to all other calls, and so forth. The results of this measurement are summarized and evaluated.

1. はじめに

最近論理型言語への関心が高まってきており、論理型言語の処理系開発がさかに行なわれている。しかし、論理型言語で記述されたプログラムの特性について評価したデータはあまり多くないのが現状である。したがって、我々は論理型のシステム記述言語ESP (Extended Self-contained Prolog) で記述された各種プログラムの静特性と動特性についてデータを収集し解析した。

ESP は第五世代コンピュータシステム研究開発プロジェクトの一環としてICOTで開発された逐次型推論マシンPSI (Personal Sequential Inference machine) 上で実行される。

本稿ではまず収集したデータを基にして、評価対象プログラムの論理型言語の特性を評価した結果を述べ、つぎにその結果を基に論理型言語で記述された実用的なプログラムの特性についてアーキテクチャの視点から考察した結果について述べる。

2. 評価の概要

プログラミングスタイル及びそのプログラムが実際にどのように動作しているかを知ることはコンパイラおよびマシンの設計者にとって大きな関心事である。実用的プログラムのプログラム特性は、コンパイラ、マシンの最適化を図るための重要な要素であるからである。又、実用的なプログラムの特性を知ることにより設計者にとってはコンパイラ又はマシンの実際の性能を予測することが容易になる。又、ユーザにとっても実用的プログラムの実行速度について正しい認識を持つことが出来るようになる。

本評価ではプログラムのリスト上に現われる特性 (静特性) と実際にそのプログラムをマシン上で実行した際のプログラムの動きに現われる特性 (動特性) について解析を行なった。

評価は各評価対象プログラムの静特性、動特性について行なった後、静特性と動特性との比較検討を行なう。

2.1 ESP プログラム

ESP は論理型のシステム記述言語であり、prologと同様にユニフィケーションによるパラメータパッシング、バックトラックによるAND-ORトリの縦型探索の機構を持っている。さらにESP はオブジェクト指向の特徴とマクロ表現メカニズムをも持つ言語でもある。ESP のオブジェクトは論理型言語における公理系に対応し、オブジェクトへのメッセージパッシングは論理型言語におけるユニフィケーションに対応する。また、マクロ表現メカニズムはプログラミングを容易にするのに役立つ。

本評価ではESP の特徴のうち主に論理型言語の特徴について評価した。

評価の対象として、実際にPSI 上で使用されている以下の4種類のプログラムを選び、各種のデータを収集した。

- 1) 8PUZZLE
- 2) BUP トランスレータ
- 3) WINDOWシステム
- 4) HARMONIZER

1) 8PUZZLE

3×3の盤面上に小さな数字から大きな数字の順に並べられたタイルを、ランダムな順に移動した後、そのタイルを最短ルートで初期状態に戻すための解を求める。

解は横型探索の戦略で求められる。すなわち、1手動かすことで解が見つかるか否かを調べだめなら2手動かして

INPUT PATTERN

8	7	6
1	*	5
2	3	4

OUTPUT PATTERN

(initial position)

1	2	3
8	*	4
7	6	5

fig.1 8PUZZLE

解が見つかるか否かを調べる。このように順に手数をふやして行き解が見つかるまで続ける。

本評価プログラムでは入力データは図-1のパターンに固定している。

このプログラムは97クローズで構成されている。

2) BUP トランスレータ

DCG (Definite Clause Grammar) の形式で記述された文脈自由文法の規則を、ボトムアップ方式にパーシングするESP プログラムに変換する。

本評価プログラムでは、固定入力データとして単純な日本語の文法を使用している。

このプログラムは 110クローズで構成されている。

3) WINDOWシステム

ビットマップディスプレイ上に新しいウィンドウの生成、文字の表示、ウィンドウの消去を行なう。処理は以下の手順で行なわれる。

- (1) ウィンドウをビットマップメモリ上に生成し、スクリーンメモリに転送する。
- (2) ウィンドウに1文字を表示し、カーソルを移動する。
- (3) ウィンドウを画面から消去し、ビットマップメモリを解放する。

このプログラムは2896クローズで構成されている。

4) HARMONIZER

ハーモナイザ・システムは、与えられたメロディーにハーモニーを付けて四声のコーラス曲に編曲する一種の音楽用エキスパートシステムである。与えられたメロディーから1音ずつを取り出しては生成規則により和音を付け、この和音と1つ前の音につけられたワーキンググループ音の組に対して進行規則を適用し正しさを確認する。もし正しければ、メロディーの次の音を取り出して再び生成規則により和音をつけ、進行規則で前の和音とのつづき具合の正しさを確認する。進行規則で許されない和音のつづき具合になっていれば、最後に生成された和音を捨て、もう一度生成規則に戻って和音の別解を求める。進行規則で許される和音が見つかるまでこれをくり返し、みつかったならばメロディーの次の音に進む。もしも和音の別解がなくなってしまう場合には、メロディーの1つ前の音に対する和音まで戻って、その別解を求めている。

このプログラムは 467クローズで構成されている。

2.2 評価方法

1) 静特性

ESP で記述されたプログラムはPSI の機械語である論理型言語KLO にコンパイルされる。このコンパイラはESP の

オブジェクト指向及びマクロ表現の機能を機械語KLO 上で実現するための変換を行なう。本評価では静特性の評価をこの機械語KLO で記述されたプログラムリストを解析することによって行なった。これはPSI 上で実際に測定したKLO プログラムの動特性との比較検討を行ないやすくするためである。

機械語KLO で記述されたプログラムはクローズの集合で表現される。クローズはヘッド部とボディ部とからなり、ボディ部はゴールの集合として表わされる。ゴールはシステムとして定義された述語(組込述語)またはユーザ自身が定義した述語(ユーザ定義述語)のいずれからなる。

静特性では以下の項目についてデータの収集を行なった。

a. クローズタイプ

クローズを以下の4種類のタイプに分類し、タイプ別にクローズの出現数及び出現率を集計した。又、extended head をもつクローズの出現数及び出現率についても集計した。

- unit clause H.
ヘッド部のみからなるクローズ。
 - pseudo unit clause H :- b, ..., b.
ボディ部が組込述語のみからなるクローズ。
 - transitive clause H :- B. 又は H :- b, ..., b, B.
ボディ部が1つのユーザ定義述語からなるクローズ、もしくは幾つかの組込述語の後に1つのユーザ定義述語が続くクローズ。
- このようにボディ部の最初のユーザ定義述語とヘッドとの間にある組込み述語列およびpseudo unit clauseのボディ部に出現する組込述語列をextended head と呼ぶ。
- regular clause H :- B, B', ..., B'.
上記以外の全てのクローズ。
- (H・・・ヘッド、B・・・ユーザ定義述語、b・・・組込述語、B'・・・ユーザ定義述語または組込述語)

• タイプ別のクローズ出現率

$$\frac{\Sigma (\text{タイプ別}) \text{クローズ数}}{\Sigma \text{クローズ数}}$$

• extended head を含むクローズの出現率

$$\frac{\Sigma \text{extended head を含むクローズ数}}{\Sigma \text{クローズ数}}$$

(Σ はプログラム中で対象となる述語又はクローズの総和)

b. OR関係

OR関係には同一ヘッド述語名を持ち、かつ引数の数も同一であるクローズ同志の関係(OR)と同一クローズ内で“;”で分けられたボディゴール列同志の関係(inner OR)

との2種類がある。

OR関係ではORの平均、inner ORの出現数及び出現率について集計した。

・OR平均

$$\frac{\sum \text{クローズ数}}{\sum \text{述語数}}$$

・inner ORの出現率

$$\frac{\sum \text{inner OR関係をもつクローズ数}}{\sum \text{クローズ数}}$$

c. AND 結合

同一クローズのボディ部で“,”で結合されたボディゴール同志の関係をいう。

AND 結合に関しては全相込述語数及び出現率とextended head として出現する相込述語数及び出現率についてデータを集計した。

・相込述語の出現率

$$\frac{\sum \text{相込述語数}}{\sum \text{ボディゴール数}}$$

・extended head 相込述語の出現率

$$\frac{\sum \text{extended head 相込述語数}}{\sum \text{相込述語数}}$$

d. 引数

ヘッドゴールに出現する引数の数の分布及び平均を集計した。又、引数として出現するデータタイプを変数、コンスタント、ストラクチャの3種類に分類し、それぞれの出現数、出現率をヘッドゴールとボディゴールごとに集計した。この集計ではatom、integer、floatingはコンスタント、compound term、vector、stringはストラクチャとした。

・ヘッド引数の平均個数

$$\frac{\sum \text{ヘッド引数の数}}{\sum \text{クローズ数}}$$

・ヘッド引数のデータタイプ別出現率

$$\frac{\sum (\text{データタイプ別}) \text{ヘッド引数の数}}{\sum \text{ヘッド引数の数}}$$

・ボディ引数のデータタイプ別出現率

$$\frac{\sum (\text{データタイプ別}) \text{ボディ引数の数}}{\sum \text{ボディ引数の数}}$$

e. 相込述語

相込述語を以下の9種類に分類してその出現率を集計した。

・データ操作相込述語

変数にバインドされている値の属性チェック、データ構造の生成及び構造体の要素又は部分要素をアクセスするための相込述語。

・算術演算相込述語

整数及び実数の算術演算を行なうための相込述語。

・論理演算相込述語

整数又はストリングの論理演算を行なうための相込述語。

・比較相込述語

2引数間の一致、大小関係を調べるための相込述語。

・実行制御相込述語

不要となった枝の検索を行なわないようにするためのカット、特定の変数に値がバインドされた際に、指定された処理を起動させるように指示するオンバックトラック等の拡張された実行制御用の相込述語

・OSインタフェース用相込述語

新しく生成するプロセスの実行環境の設定、ソフトウェアトラップのは発生、メモリ管理を行なうための相込述語。

・ハードウェア操作相込述語

入出力命令、内部レジスタのアクセスなど従来のハードウェア操作と同様の機能をもつ相込述語。

・オブジェクトサポート用相込述語

メソッド呼び出し、スロットアクセスのための相込述語。

・その他の相込述語

コンソールとの入出力を行なうためのもの、最適化のためのハッシュ関数の計算を行なうためのもの、及びデータタイプ変換用の相込述語

・相込述語のタイプ別出現率

$$\frac{\sum (\text{タイプ別}) \text{相込述語数}}{\sum \text{相込述語数}}$$

2) 動特性

ESP プログラムはK10 プログラムにコンパイルされた後、PSI 上のOS (SIMPDS) のもとで実行される。動特性はその際のK10 プログラムの動きを解析することにより得られた。

PSI はK10 をマイクロインタプリタと専用のハードウェアによって直接解釈実行するマシンである。動特性のデータ収集にあたっては、PSI のマイクロインタプリタの数ヶ所に測定点を設け、プログラムがその地点を通過する際に必要なデータを収集した。したがって、この測定方法により静特性に対応した動特性のデータをほぼ実時間で収集することが出来た。

動特性の解析のため以下の項目についてデータを収集した。

a. クローズタイプ

呼び出されたクローズ（バックトラックによる再呼び出しを含む）をunit clause、pseudo unit clause、transitive clause、regular clauseの4種類に分類し、各タイプごとに出現回数及び出現率を集計した。

・クローズタイプの出現率

$$\frac{\sum (\text{タイプ別}) \text{呼び出されたクローズ数}}{\sum \text{呼び出されたクローズ数}}$$

b. 非決定的クローズ

他の選択肢を持っているクローズを非決定的クローズと呼ぶ。呼び出されたクローズ（バックトラックによる再呼び出しを含む）が非決定的クローズである回数及びその出現率について集計した。

・非決定的クローズの出現率

$$\frac{\sum \text{他の選択肢をもつクローズ数}}{\sum \text{呼び出されたクローズ数}}$$

c. 実行順序

クローズはプログラムで記述された順に、ボディゴールは左から右へ順に実行され、ユニフィケーションに失敗（フェイル）した場合他の選択肢がある最近の肢まで戻って（バックトラック）再実行される。

実行されるボディゴールにはユーザ定義述語、組込述語、inner ORがある。

フェイルはユーザ定義述語及び組込述語実行時に発生することがある。バックトラックには自クローズにまだ実行されていない他のクローズが残っている場合のバックトラック（シャローバックトラック）と、呼び出し元又はさらにその先まで戻って行なうバックトラック（ディープバックトラック）とがある。

実行順序に関してはユーザ定義述語の出現回数及び出現率、組込述語の出現回数及び出現率、extended head 組込述語の出現回数及び出現率、inner ORの出現回数及び出現率、ユーザ定義述語のユニフィケーションの成功及び失敗回数、組込述語の成功及び失敗率、バックトラックがシャローである場合とディープである場合のそれぞれの出現回数及び出現率について集計した。（ユニフィケーションではバックトラックによる再実行を含む。）

・ユーザ定義述語の出現率

$$\frac{\sum \text{実行されたユーザ定義述語数}}{\sum \text{実行されたボディゴール数}}$$

・組込述語の出現率

$$\frac{\sum \text{実行された組込述語数}}{\sum \text{実行されたボディゴール数}}$$

・extended head 組込述語の出現率

$$\frac{\sum \text{実行されたextended head 組込述語数}}{\sum \text{実行された組込述語数}}$$

・inner OR出現率

$$\frac{\sum \text{実行されたinner OR数}}{\sum \text{実行されたボディゴール}}$$

・ユーザ定義述語成功（失敗）率

$$\frac{\sum \text{ユーザ定義述語成功（失敗）回数}}{\sum \text{実行されたユーザ定義述語数}}$$

・組込述語成功（失敗）率

$$\frac{\sum \text{組込述語成功（失敗）回数}}{\sum \text{実行された組込述語数}}$$

・シャロー（ディープ）バックトラックの出現率

$$\frac{\sum \text{シャロー（ディープ）バックトラック回数}}{\sum \text{バックトラック回数}}$$

d. ニフィケーション

実行時にユーザ定義述語のボディゴールに出現する引数の平均個数及び分布について集計した。

又、引数のデータタイプを変数、コンスタント、ストラクチャの3種類に分類し、実行されたデータタイプごとの出現回数及び出現率をボディゴール、ヘッドゴールについて集計した。（ヘッド引数にはバックトラックによる再呼び出しを含む）

・ユーザ定義述語のボディ引数の平均個数

$$\frac{\sum \text{実行されたユーザ定義述語のボディ引数の数}}{\sum \text{実行されたユーザ定義述語のボディゴール数}}$$

・ボディゴール引数のデータタイプ別出現率

$$\frac{\sum \text{実行された（タイプ別）ボディ引数の数}}{\sum \text{実行されたボディ引数の数}}$$

・ヘッド引数のデータタイプ別出現率

$$\frac{\sum \text{呼び出された（タイプ別）ヘッド引数の数}}{\sum \text{呼び出されたヘッド引数の数}}$$

ユニフィケーションは呼び出し元ボディゴールの引数と呼び出されたクローズのヘッド引数との間で行なわれる。ユニフィケーションに関してはボディゴールに出現する引数のデータタイプとその引数に対応してヘッドゴールに出現する引数タイプについて出現率を集計した。

- ・ヘッド引数データタイプとボディ引数データタイプの組み合わせによる出現率

$$\frac{\sum \text{呼び出された(組み合わせタイプ別)引数の数}}{\sum \text{呼び出されたヘッド引数の数}}$$

e. 組込述語

組込述語をデータ操作、算術演算、論理演算、比較、実行順序、OSサポート、ハードウェア操作、オブジェクトサポート、その他に分類し、その出現率についてデータを集計した。

- ・組込述語のタイプ別出現率

$$\frac{\sum \text{実行された(タイプ別)組込述語数}}{\sum \text{実行された組込述語数}}$$

f. クローズインデクシング

クローズ呼び出しの高速化を計るためにヘッドゴール中に出現する変数以外の引数をキィとしてインデクシングを行なうことが出来る。コンパイラは対象となるプログラムを静的に解析し、ヘッドの第1引数が変数以外で2種類以上のキィとなるシンボルを持ち、かつOR関係が4個以上のクローズ群に対してインデクシングコードを生成する。本評価では実行されたボディゴールに対するインデクシングの成功率について集計した。

- ・インデクシングの成功率

$$\frac{\sum \text{インデクシングが成功した回数}}{\sum \text{実行されたボディゴール数}}$$

3. 評価結果

静特性の解析結果を表1～3に載せる。

3.1 静特性

1) クローズの特性

8PUZZLE及びHARMONIZERではunit clauseの出現率が非常に高く、OR関係の平均値が高い(表-1)。OR関係の分布表によると8PUZZLEでは24及び44のOR関係を持つクローズが1種類づつあり全クローズの70%を占めており、また、HARMONIZERでは99以上のOR関係をもつクローズが3種類あり、全クローズの40%を占めている。両プログラムともOR関係の値の大きいクローズはunit clauseもしくはpseudo unit clauseで構成されており、データベースがプログラムに対して大きな割合を占めていることがわかる。なをデータベースにpseudo unit clauseが出現するのはヘッドの後に実行制御用の組込述語cutがあるためである。cutによって、データベースの1つが選択されると以後のバックトラックでデータベースの別の値が選択されないようにしている。

inner ORの出現率はプログラムによって異なるが比較的高いものがある(表-1)。これはinner ORはコーディングが容易であるためであると思われる。また通常のORよりもコード量が少ないと言う利点もある。

2) ボディ部の特性

ボディ部における特徴の1つは全てのプログラムにおいて組込述語の出現率がユーザ定義述語に比べて高い点である(表-1)。Prologプログラムの静特性[尾内 84-3]よりも組込述語の出現率が高くなっているのは、KL0にはデータ操作等のために豊富な組込述語が用意されていることに影響されていると思われる。又、全クローズ中でextended headを持つクローズは全プログラムとも比較的高く、BUPでは73%を占めている(表-1)。組込述語の中でextended headとして出現するものの比率についてもWINDOWでは43%であるが他のプログラムでは50%以上を占めている。BUPとHARMONIZERにおけるextended headは主に組込述語cutによるものである。

ボディ引数のデータタイプは全クローズとも変数が多い(表-1)。

3) ヘッド部の特性

ヘッド引数の平均個数は3.4個であり[尾内 84-3]とほぼ同じ結果が出ている。各プログラムのクローズ単位の分布を見るとHARMONIZER以外は2引数のクローズが多い(表-3)。HARMONIZERでは5引数が最も多いが、この5引数のクローズはデータベースである。また、WINDOWプログラム以外ではヘッド引数は10以下であるが、OSの一部であるWINDOWプログラムには10引数以上のクローズが含まれている点は特徴的である。

ヘッドゴールの引数として出現する引数のデータタイプをコンスタント、変数、ストラクチャに分類すると、8PUZZLE、HARMONIZERではコンスタントの出現率が高い(表-1)。これはデータベースであるユニットクローズの引数がコンスタントであることによる影響である。

4) 組込述語

出現した組込述語を9種類に分類した場合、プログラムによって各組込述語の出現率にはバラツキがある(表-2)。しかし、その中でも比較的多いのがデータ操作用(DATA)と実行順序制御用(CONTROL)の組込述語である。又、出現した個々の組込述語について調べた結果、cut及び四則演算の出現率が非常に高い。出現率の高い上位5種で全体の組込述語出現率の50%以上を占めており、使用される組込述語が偏っていることを示している。

以上よりESPプログラムの特徴をまとめる。

- ・プログラムによってはデータベースが全プログラム中で大きな比率を占める。
- ・ボディゴール中に出現する組込述語の比率は約70%でユ

ーザ定義述語よりも多く、さらにその半分がextended head である。

- ・ヘッド引数の平均個数は約3個である。
- ・プログラム中に出現する組込述語のうち出現率の高い上位5種程度で全組込述語出現回数の半分以上を占める。

3.2 動特性

動特性の解析結果を表4～8に載せる。

1) クローズ

実行されたクローズをタイプ別に分類した場合、全プログラムに共通してregular clauseの出現率が全体の50%以上を占めている(表-4)。したがって、半分以上のクローズが、呼び出されたクローズからの復帰に備えて環境のセーブを必要とする可能性あることを示している。

PSIでは処理の高速化のためクローズの呼び出しに際してクローズのインデクシングを行なっている。したがって、実際に実行されたclause数にその影響が表われており、データベースを持つ8PUZZLE、HARMONIZERではクローズインデクシングの成功率が高く、unit clause、pseudo unit clauseの出現率が静特性に比べてそれほど高くない(表-4)。

2) 実行制御

ボディ部に関する動特性で目立つものは、組込述語の出現率が実行されたボディゴールの約70%以上を占めている点である(表-4)。特に8PUZZLEでは全体の93%と非常に高い比率を占めている。組込述語がExtended head中に出現する割合は8PUZZLEでは低い但他はほぼ全体の50%を占めている。

実行されたボディゴールの引数の平均個数は4.2個である(表-4)。引数の分布を見ると全プログラムとも2及び3引数のクローズが多い(表-8)。その他に8PUZZLEでは6引数、WINDOWでは15引数、HARMONIZERでは8引数にもう一つの山があるのが特徴である。

呼び出されたクローズが他の選択肢を持っている割合はWINDOW以外はほぼ70%以上を占めており、バックトラックに備えての環境のセーブが必要となる可能性の高いクローズの比率が多い(表-4)。WINDOWに他の選択肢が少ないのはWINDOWはOSプログラムの一部であり、決定的な処理が多いためであると思われる。

呼び出されたクローズがユニフィケーションに失敗する割合は大きく、WINDOW以外では全体の約40%もある(表-5)。WINDOWプログラムはユニフィケーションにおいても失敗する割合は他のプログラムよりも少なく、プログラムの実行が方向性であることを表わしている。

しかし、WINDOWプログラムでは呼び出されたクローズの半分が他の選択肢を持っており、OSのプログラムは非常に決定的に記述されているだろうとの当初の予測に反してい

た。

組込述語の実行が失敗する割合は全プログラムとも低い(表-5)。これは組込述語には実行が失敗することの無いもの(cut等)があること、失敗する可能性のあるもの(算術演算等)でも実際に失敗するとは少ないことによると思われる。

ユニフィケーションの失敗がシャローバックトラックを引き起こす割合は、バツキはあるが平均で全バックトラックの約70%あり、BUPでは91%と非常に高い(表-5)。したがって、全プログラムとも本来のバックトラックというよりケースプランチとしてのバックトラックが多いと言える。

3) ユニフィケーション

実行中にボディ部に出現する引数の値のデータタイプはコンスタントとストラクチャとで平均75%と多く、ユニフィケーション実行時にはボディ引数の値が定まっている場合が多いことを示している(表-4)。これはボディゴール引数の平均個数が約4個であるとするで4引数のうち3引数は呼び出したクローズへのデータ転送用として、残りの1引数は呼び出したクローズからのデータ受信用として使用されるパターンに相当する。BUPではボディ、ヘッドともストラクチャの出現率が高く、プログラムの特徴が表われている。

ユニフィケーション実行直前の呼び出し元ボディゴール引数のデータタイプと呼び出されたクローズのヘッドに出現する引数のデータタイプの組み合わせを見ると(表-6)、BUPを除いて呼び出し側の引数がコンスタント又はストラクチャ、呼ばれた側の引数が変数である場合が多く、つづいて、コンスタント同志のユニフィケーションが多い。このことより、ユニフィケーションのパターンとしては呼び出し元ボディゴールから呼ばれたクローズへのデータバッシングが最も多く、次に同一データタイプのパターンマッチングが多いことがわかる。BUPでは変数同志のユニフィケーションが多く、次にストラクチャ同志のユニフィケーションが多い。BUPで変数同志のユニフィケーションが多いのは多段のネストの後、値が呼び出し元へ返されるようにプログラムが記述されているためである。

4) 組込述語

組込述語を静特性と同様に9種類に分類した場合、算術演算用(ARITH)と比較用(COMP)の組込述語が比較的多いが、OSプログラムプログラム以外は使用されている組込述語に偏りがある(表-7)。個々の組込述語の出現率について注目した場合、cut、算術演算の出現率が高い。又、各プログラムとも出現回数の多い上位3から4個の組込述語で全組込述語の50%を占める。

以上のことより、ESP プログラムの動特性をまとめると以下ようになる。

- ・実行されるボディゴールが組込述語である割合が全体の70%と非常に高い。
- ・実行されるクローズが非決定的である割合が高い。
- ・ユーザ定義述語のユニフィケーションが失敗する割合は全ユニフィケーションの約半分と多い。
- ・全バックトラックのうちシャローバックトラックの比率が75%でディープバックトラックよりも高い。
- ・ボディゴール引数の平均個数は約4個である。
- ・実行中に出現する組込述語のうち出現率の高い上位4種類程度で全出現組込述語の半分以上を占めている。

3.3 静特性と動特性の比較検討

動特性の各種データはPSIのOSであるSIMPOSの下で実行されている状態で収集された。したがってプログラム実行中にプロセススイッチが発生した場合、対象としているプログラム以外にSIMPOSのプロセス管理ルーチンが実行される。WINDOWプログラム以外のプログラムについてはそのような状態は発生しないため問題はないがWINDOWプログラムの動特性には自プログラム以外に上記の要素が含まれている。したがって、本来WINDOWプログラムの静特性と動特性の比較を行なうことは正しくない。しかし、OSプログラムの特性を比較するという意味ではそれほど不適当ではないと思われるので、以下の比較検討ではWINDOWも比較の対象とした。

1) クローズタイプの特徴

クローズタイプの出現率を比較すると動特性では全プログラムともregular clauseの出現が多いが、静特性ではそのような特性は見られない(表-1、表-4)。その原因としてテイルリカーシブにプログラムが記述されていること、クローズインデクシングのためデータベースの影響があまり現われていないことによるためであろう。

2) 実行制御の特徴

静特性におけるORの平均と動特性における実行された全クローズ数に対するバックトラックの比率を比較するとOR平均の小さいWINDOWはやはりバックトラック率が小さい(表-1、表-5)。8PUZZLE、HARMONIZERでのバックトラックの比率が小さいのはクローズインデクシングのためであると思われる。

3) ユニフィケーションの特徴

静特性ではヘッド引数のデータタイプとして大きなデータベースをもつ8PUZZLEとHARMONIZERではコンスタントが多く、他は変数が多いという特徴があったが、動特性では全て変数の出現率が高くなっている。これは静特性に比べてデータベースに実際にアクセスされる割合が少ないためコンスタントの出現率が低くなったものと思われる。

BUPにおけるストラクチャの出現率は静特性においても動特性においても高く、同じ傾向を示している。ボディ側引数のデータタイプについては静特性と動特性とは異なり、静特性では変数の出現率が非常に高いのに反して、動特性ではコンスタント又は、ストラクチャの出現率が高くなっている。これは、ボディゴールの引数の実行時には、変数に値がバインドされている比率が高いことによる。

4) 組込述語の特徴

組込述語の出現率は静特性、動特性の双方とも高い。また、静特性で出現率の高い組込述語は動特性においても出現率が高い。

静特性、動特性との間にはあまり相関関係が無く、特にデータベースの影響があまり動特性に現れていない。この理由に1つとしてはクローズインデクシングによる影響があると思われる。クローズインデクシングを行なわなければデータベース中のデータ検索回数等が増し、クローズタイプ、ヘッド引数タイプ、バックトラック率については静特性と動特性に同じ傾向が出たのではないと思われる。言い換えれば、クローズインデクシングの有効性が動特性に現れていると言える。

静特性と動特性との関係をまとめると以下ようになる。

- ・静特性のOR関係の出現率と動特性のバックトラックの頻度は同じ傾向を示す。
(インデクシングによりその傾向は変化する。)
- ・組込述語の出現率は静特性、動特性とも高い。
- ・出現率の高い組込述語は静特性、動特性とも共通しており、出現率の高い上位4種類程度で全出現組込述語の半分以上を占める。

4.アーキテクチャの考察

ヘッド引数の数はほとんど10個以下であることから、10個までの引数をもつクローズに注目しそれらのユニフィケーションの最適化を図る方式が考えられる。組込述語の出現率が非常に高いことから、組込述語の高速化は全体の速度に影響すると思われる。特に出現率の高い組込述語は四則演算、cut等非常に限られているためそれらに特に注目して高速化することにより全体の性能を向上することが出来ると思われる。

又、extended headとして出現する組込述語の割合が比較的高いため、extended headを含むhead処理を内部レジスタ(バッファ)内で行なう方法は有効である。

[山本 84-10]

ユニフィケーション時に呼び出し元引数の値が定まっている割合が高いことから、ヘッド引数の値が変数以外であった場合クローズインデクシングが有効であることが予想される。しかし、呼び出されたクローズに他の選択肢が少

ない場合、インデクシングに要する処理速度が必ずしもユニフィケーションの処理速度よりも速くないことがある。従って、場合によってはクローズインデクシングが有効ではない場合があるので注意が必要である。

呼び出されたクローズが他の選択肢を持つ場合、後のバックトラックに備えて再実行のための環境をセーブする必要がある。しかし、ユニフィケーションに失敗した場合及びextended head に組込述語cut が出現した場合にはバックトラックに備えての環境セーブが不要となる。動特性によるとバックトラックのうちシャローバックトラックの比率が高いこと、組込述語cut の出現する可能性が比較的高いことから、処理の高速化のためには、実行中のユニフィケーション及びextended head の処理が終了するまでバックトラック用の環境セーブを遅らせることは処理の高速化につながる。又、バックトラックが多いことから、一度呼び出し元の引数の値を呼び出されたクローズの環境にコピーしてからユニフィケーションを行なう方式（引数コピー方式）はバックトラック時にコピーした値を再利用出来るので高速化に非常に有効であると考えられる。

5. おわりに

ESP プログラムを静特性、動特性の両面から解析し、それぞれに現われる特性及び両特性の相違点、共通点について検討した。

この解析から以下のことがわかった。

- ・組込述語の出現率は静特性、動特性とも高い。
- ・出現率の高い組込述語の上位4種類程度で全出現組込述語の約半分を占める。
- ・ユーザ定義述語の処理ではユニフィケーションの約半分は失敗する。
- ・全バックトラックのうちシャローバックトラックの比率がディープバックトラックよりも高い。

対象プログラム数が少なく、これらのデータだけで一般論を議論することはむずかしいが、およそのプログラム特性は解析出来たと思われる。今後さらに各種のプログラムからデータを収集することにより、さらにはっきりとしたプログラム特性がつかめるものと思われる。

これらの結果は、今後論理型言語マシンを設計する際の一助となるであろう。

最後に、日頃ご指導ご鞭撻いただく横井 俊夫第3研究室長、内田 俊一第3研究室長代理、評価対象プログラムを提供し、議論して下さいICOT研究員の方々に深謝する。

参考文献

- [尾内 84-3] 尾内、清水、益田、麻生
逐次型Prologプログラムの解析
Proc. of The Logic Programming Conference'84
- [山本 84-10] 山本、横田、西川、轟、内田
逐次型推論マシンのマイクロインタプリタ
情報処理学会 第29回記号処理研究会
- [Yokota 84-11] Yokota, Yamamoto, Taki,
Nishikawa, Uchida, Nakajima, Mitsui
A Microprogrammed Interpreter for The Personal Sequential Inference Machine
Proc. of The International Conference on Fifth Generation Computer Systems'84 Japan
- [Chikayama 84-11] Chikayama
Unique Features of ESP
Proc. of The International Conference on Fifth Generation Computer Systems'84 Japan

	8PUZZLE	8UP	WINDOW	HARMONIZER
total predicate	25	58	2392	62
total clause	97	110	2895	467
CLAUSE TYPE				
unit clause	72(74%)	6(6%)	154(5%)	296(63%)
pseudo unit clause	12(13%)	43(39%)	948(33%)	135(29%)
transitive clause	6(6%)	22(20%)	388(13%)	14(3%)
regular clause	7(7%)	39(35%)	1405(49%)	22(5%)
ave. of OR relation	3.9	2.0	1.2	7.5
inner OR	2(2%)	23(21%)	539(19%)	7(1%)
built-in	91(73%)	315(65%)	9116(70%)	269(79%)
extended head clause	17(18%)	80(73%)	1358(47%)	145(31%)
extended head built-in	48(53%)	166(53%)	3934(43%)	207(77%)
ave. of head arguments	2.3	3.9	3.2	4.2
DATA TYPE OF HEAD ARG.				
constant	142(64%)	14(3%)	557(6%)	1509(77%)
variable	73(33%)	284(67%)	7851(89%)	409(21%)
structure	7(3%)	128(30%)	387(5%)	48(2%)
DATA TYPE OF BODY ARG.				
constant	62(18%)	76(7%)	8469(27%)	108(16%)
variable	265(79%)	905(85%)	22243(72%)	544(80%)
structure	10(3%)	88(8%)	276(1%)	25(4%)

table 1. result of static analysis

8PUZZLE

D	A	L	C	C	O
A	R	O	O	O	B
T	I	G	M	N	J
A	T	I	P	T	E
	H	C			C
27	35	22	4	10	1

8UP

D	A	C	C	O
A	R	O	O	B
T	I	M	N	J
A	T	P	T	E
	H			C
39	12	18	31	10

WINDOW

D	A	L	C	C	O
A	R	O	O	O	B
T	I	G	M	N	J
A	T	I	P	T	E
	H	C			C
18	9	1	12	16	44

HARMONIZER

D	A	C	C	O
A	R	O	O	B
T	I	M	N	J
A	T	P	T	E
	H			C
10	31	5	51	3

table 2. static analysis of built-in

ARG.	8PUZZLE(%)	8UP(%)	WINDOW(%)	HARMONIZER(%)
0	1	0	0	0
1	2	4	16	3
2	83	48	41	11
3	3	7	14	26
4	6	2	11	2
5	3	2	6	50
6	2	4	5	1
7	0	2	1	0
8	0	12	1	6
9	0	4	4	0
10	0	0	0	1
11	0	0	0	0
12	0	0	1	0
13	0	0	0	0
14	0	0	0	0
15	0	0	0	0
16	0	0	0	0
17	0	0	0	0
18	0	0	0	0

table 3. distribution of static head argument count

	8PUZZLE	8UP	WINDOW	HARMONIZER
total clause	78732	527	2814	14635
CLAUSE TYPE				
unit clause	30326(39%)	5(1%)	8(0%)	2714(19%)
pseudo unit clause	84(0%)	63(12%)	853(30%)	2529(17%)
transitive clause	16(0%)	112(21%)	485(17%)	311(2%)
regular clause	48306(61%)	347(66%)	1468(53%)	9081(62%)
non determ. clause	59201(75%)	374(71%)	1374(49%)	9944(68%)
BODY CALL				
user	35423(7%)	231(30%)	1366(11%)	5639(26%)
built-in	479908(93%)	503(64%)	10569(82%)	15469(70%)
inner OR	10(0%)	44(8%)	961(7%)	890(4%)
extended head built-in	35698(7%)	376(75%)	4562(43%)	9830(64%)
ave. of user goal	4.0	3.2	5.0	4.5
DATA TYPE OF BODY ARG.				
constant	88461(62%)	127(17%)	6075(59%)	15756(62%)
variable	35423(25%)	367(49%)	1795(17%)	4561(18%)
structure	17681(13%)	255(34%)	2463(24%)	4985(20%)
DATA TYPE OF HEAD ARG.				
constant	88612(45%)	739(55%)	11736(88%)	31979(61%)
variable	91277(46%)	148(11%)	1401(10%)	15167(29%)
structure	17697(9%)	466(34%)	261(2%)	4967(10%)
clause indexing	50%	0%	12%	25%

table 4. result of dynamic analysis

	built-in		user		backtrack	
	success	fail	success	fail	shallow	deep
SPUZZLE	467211 (97%)	12696 (3%)	48111 (61%)	30621 (39%)	26470 (61%)	16847 (39%)
BUG	372 (75%)	123 (25%)	341 (65%)	186 (35%)	281 (91%)	28 (9%)
WINDOW	9094 (94%)	593 (6%)	2417 (86%)	397 (14%)	546 (55%)	444 (45%)
HARMONIZER	12360 (83%)	2592 (17%)	8172 (56%)	6475 (44%)	7160 (79%)	1907 (21%)

table 5. fail and backtrack

SPUZZLE			
caller	callee		
	variable	constant	structure
variable	17752(9%)	30328(15%)	30(0%)
constant	70845(36%)	60949(31%)	1(0%)
structure	15(0%)	0(0%)	17666(9%)
BUG			
caller	callee		
	variable	constant	structure
variable	454(34%)	4(0%)	114(9%)
constant	92(7%)	58(4%)	0(0%)
structure	193(14%)	86(6%)	352(26%)
WINDOW			
caller	callee		
	variable	constant	structure
variable	1412(10%)	433(3%)	39(0%)
constant	7502(56%)	888(7%)	2(0%)
structure	2822(21%)	80(1%)	220(2%)
HARMONIZER			
caller	callee		
	variable	constant	structure
variable	774(2%)	5431(11%)	139(0%)
constant	28733(55%)	8497(16%)	0(0%)
structure	2472(5%)	1239(2%)	4828(9%)

table 6. pattern of unification

8PUZZLE

D	A	R	L	C
A	R	I	O	O
T	I	T	G	M
A	T	H	I	P
			C	
19	44	31	6	

BUP

D	A	C	C	O
A	R	O	O	B
T	I	M	N	J
A	T	P	T	E
	H			C
14	17	41	26	2

WINDOW

D	A	L	C	C	O	H	O
A	R	O	O	O	S	A	B
T	I	G	M	N	I	R	J
A	T	Y	P	T	D	E	C
	H	C				C	
10	9	13	14	21	12	13	28

HARMONIZER

D	A	C	C	O
A	R	O	O	B
T	I	H	N	J
A	T	P	T	E
	H			C
13	66	18	10	13

table 7. dynamic analysis of built-in

ARG.	8PUZZLE(%)	BUP(%)	WINDOW(%)	HARMONIZER(%)
0	0	0	2	0
1	0	0	8	7
2	55	21	26	29
3	0	66	24	10
4	0	7	12	7
5	0	2	3	15
6	45	0	1	1
7	0	1	2	0
8	0	0	2	30
9	0	3	3	0
10	0	0	1	0
11	0	0	1	1
12	0	0	3	0
13	0	0	0	0
14	0	0	0	0
15	0	0	10	0
16	0	0	2	0
17	0	0	0	0
18	0	0	0	0

table 8. distribution of dynamic body argument count