TM-0011, 0014～0023

# 情報処理学第２７回全国大会

## 発表論文集（１１件）

October, 1983

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

# パーソナル逐次型推論マシンψ
## —— そのマイクロインタプリタ ——

山本 明　横田 実　西川 宏　萬 和男　内田 俊一
（（財）新世代コンピュータ技術開発機構）

## 1. はじめに

パーソナル逐次型推論マシンψ（ＰＳＩ）は、第５世代コンピュータシステム研究開発におけるソフトウェア開発用ツールとして使用されるパーソナルマシンである。ψ上で実行されるプログラムは、OSを含め全て論理型言語で記述されることが特徴で、機械語自身もProlog-like な論理型言語（KL0 と呼ぶ）に設定されている。機械語の記号表現は、コンパイラによって内部形式に変換された後ψのマイクロプログラムにより直接解釈実行される。本論文ではψのマイクロプログラム化されたインタプリタ（マイクロインタプリタと呼ぶ）についてその処理方式と特徴を報告する。

## 2. 内部形式

ψハードウエアのメモリアクセスの単位は、 8ビットのタグ部と32ビットのデータ部とからなるワードである。ψの機械語仕様は、DEC10 Prologを一部拡張した論理型言語であり、クローズは 1対 1の内部形式に変換される。すなわちクローズ表現はクローズヘッド部、ヘッドアーギュメント部，ボディ部からなる数ワードのワード列で表わされ、マイクロインタプリタは、各ワードに付けられたタグを解釈しながらこの内部形式を実行していく。同一述語名をもつクローズは１つの集合（プロシジャと呼ぶ）として表現され、プロシジャヘッド部、クローズインデキシングのためのテーブル、及びクローズの内部形式の集合で構成される。 （図－１）

## 3. スタック方式

内部形式を解釈実行する際にはクローズの実行順序制御と変数への値の代入が必要だが、これらの管理方式としてψではDEC10 Prologと同じようにスタックを用いる。使用するスタックは、ローカル、グローバル、トレイル及びコントロールの４種類であり、各スタックにはハードウェアのアドレス変換機構により論理的に独立した空間が割り当てられそれぞれ16M ワードまで伸ばすことが出来る。
各スタックの用途は次のようになっている。

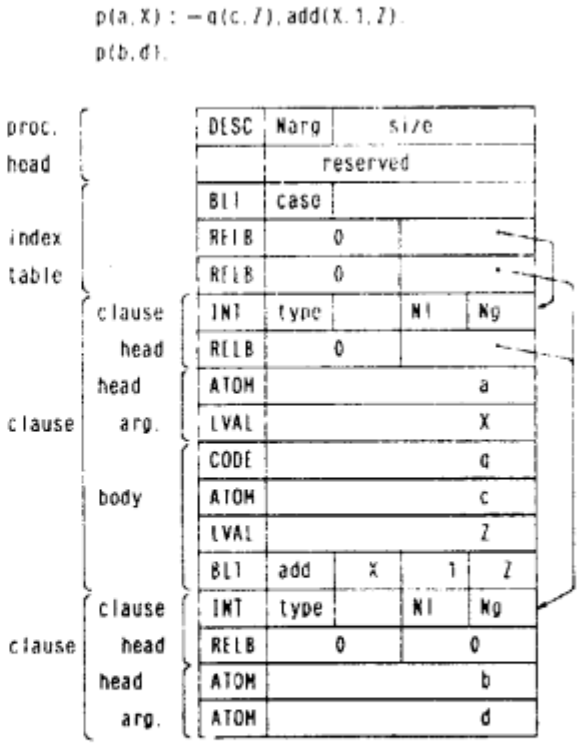
グローバルスタック・・・グローバル変数（構造体中に表われる変数）に対応するセル及び各種情報の格納

ローカルスタック・・・グローバル変数以外の変数に対応するセルの格納

トレイルスタック・・・変数セルにバインドされた値をbacktrack 発生時に元に戻すためのセルアドレスの格納

コントロールスタック・・・クローズの実行順序を制御する情報の格納

DEC10 Prologと異なりコントロール情報をローカルスタックから分離したのはKL0 にDEC10 より豊富な機能を導入したため、用途別にスタックを分けることにより、実行制御を行ないやすくするためである。その例としてクローズ内ORの制御がある。クローズ内OR中のゴール列は、自クローズと変数領域を共有している。しかし、バックトラックやカットに関する実行制御の情報は自クローズのそれとは異なるため、クローズ内OR用のコントロールフレームを別に作ることが必要で、ローカルスタックから分離することが望ましい。 （図－２）




図－１

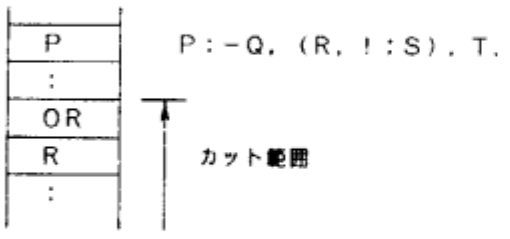
## 4. Structure Sharing 方式

構造体の表現形式にはstructure sharing とstructure copying とがあるが、仮想記憶をサポートしないψでは実行速度の点からsharing 方式を採用した。sharing 方式では、構造体はスケルトンアドレスとフレームベースアドレスで規定されるため、変数セルは 2つの情報を持つ必要がある。ψでは上記アドレスはψの全論理空間を指すことが出来るようにそれぞれ1ワードで表現される。したがって、これらはグローバルスタックに積まれ、変数セルからはポインタによって指される。

## 5. 実行順序制御

マイクロインタプリタの基本動作は、呼び出し側のゴールアーギュメントと呼ばれた側のヘッドアーギュメントとを取り出してそれらの間でユニフィケーションを行なうことであるが、呼ばれた側のクローズが更に別のクローズを呼び出すというような繰り返しを処理するため、呼び出し元へ戻るための情報の管理と変数領域の動的な生成が必要となる。さらにユニフィケーションに失敗した場合、途中まで戻って新たに別のクローズの呼び出しからやり直すという処理を行なうための情報の管理も必要である。これらの処理はスタックをもちいておこなわれるが、ψでは処理の高速化のためスタックの先頭1フレーム分をフレームバッファと呼ぶCPU 内のワークファイル上に置き、その上でユニフィケーションを行なう。又、実行中の環境に対する情報も全て制御用のレジスタ中に置くことにより、処理の高速化をはかっている。

この他、ψではGC(Garbage Collector) の対象外であるローカル、コントロールスタックの消費を極力押えるためTRO (Tail Recursion Optimization)を採用している。




図－2

## 6. Extended Head

組込述語処理では、クローズの呼び出し処理が不要であり、呼び出し元と同一のクローズの環境で処理が行なわれる。この処理は通常スタック上で行なわれるが、組込述語がクローズヘッドに続いて出現する場合、ψではユニフィケーションがフレームバッファ（ハードウェアのバッファ）上で行なわれるためこの組込述語もヘッドの一部と見なし（extended head と呼ぶ）、フレームバッファの内容を用いて処理することが可能になる。これにより組込述語処理の高速化が期待できる。

## 7. おわりに

本稿ではψのマイクロインタプリタについてその処理方式と特徴について述べた。

マイクロインタプリタの基本仕様の決定はほぼ完了し、現在マイクロプログラムのコーディングを行なっている。今後はシミュレータによる本方式の有効性の検証、OSのためのシステム制御機能の追加を行なっていく予定である。

最後に、日頃有益な助言をいただく近山氏はじめICOTメンバ諸氏に深く感謝する。

## 参考文献

1. Warren, D.H.D
「Implementing Prolog compiling predicate logic programs」
D.A.I Research Report no 39 ～40(May, 1977)

2. Warren, D.H.D
「An improved Prolog Implementation which optimises Tail Recursion」
Proc. of the Logic programming workshop, Hungary (July, 1980)

3. Chikayama,T.et al
「Fifth Generation Kernel language Version-0」
Proc. of the Logic programming conference,Japan (March,1983)