4B-11 並列Ｐｒｏｌｏｇマシン ＰＡＲＫ上の Ｐｒｏｌｏｇ処理系の実現について

松田秀雄（神戸大・自然科学），増尾剛（神戸大・工），
小畑正貴（岡山理大・工），金田悠紀夫，前川禎男（神戸大・工）

１．はじめに

現在，筆者らの研究室では，並列Prologマシン ＰＡＲＫ (Parallel Processing System of Kobe University) を製作中である[1]．本稿では，ＰＡＲＫ上に実現する並列Prolog処理系について，その概要と実現の方式について述べる．

２．ハードウェア

ＰＡＲＫのハードウェアの特徴をあげると次のようになる．

◇16ビットマイクロプロセッサMC68000（モトローラ社製）によるマルチマイクロプロセッサシステム
◇プロセッサを一本の共通バスにより結ぶ密結合構成
◇１台の入出力処理用プロセッサ（ホスト）と複数台の並列実行用プロセッサ（スレーブ）の２種類のプロセッサ構成
◇ブロードキャスト機能を持つ共有メモリ

共有メモリは，各スレーブプロセッサに512KBづつ付いており，それぞれ別々のアドレスに割り当てられている．ブロードキャスト機能により，全プロセッサの共有メモリへ同一のデータを一斉書き込みすることができる．この機能はプロセッサ間で共有されるデータの更新に有効である．

３．並列Prolog

本稿で述べる並列Prolog（以下PARK-Prologと呼ぶ）の主な機能をまとめると次のようになる．

◇ＡＮＤ並列，ＯＲ並列を組み合わせて行える
◇逐次実行と並列実行の両方を記述できる
◇並列実行部分はユーザが明示的にプロセスとして指定
◇プロセス間通信は通信チャネルを介して行う

３．１ 並列実行部分の指定

並列に動作する部分は次にあげるようなコンストラクタ（ゴールリテラル間のオペレータ）およびモード宣言により指定できる．これらによって指定された部分はそれぞれ別プロセスにより実行される．DEC-10 Prologと同様 ‘,’ または ‘;’ で結ばれたゴールや，並列ＯＲモードで宣言されていない定義節の呼び出しは単一のプロセスにより逐次的に実行される．

◇並列ＡＮＤコンストラクタ…… ‘//’
◇並列ＯＲモード宣言…… ‘%mode par’

変数の束縛環境は，プロセスごとに独立である．プロセスの生成が行われる時点で，呼び出し元プロセスの環境がコピーされる．ＡＮＤで結ばれたゴール間での共有変数の環境は，プロセス生成の時点で別々となるが，これらの値の照合はプロセス間通信機能を使ってユーザが明示的に行う．

３．２ プロセス間通信

プロセス間通信はチャネルを介して行う．チャネルとは共有メモリ上に設けられた一種の変数領域であり，sendとreceiveの２つの操作が行える．１つのチャネルに対して同時には１つのプロセスだけしか操作は行えない．sendとreceiveは対になって使われ，あるプロセスがどちらか一方の操作を行うと別のプロセスがもう一方の操作を行うまで実行は中断される．この機能を利用してプロセス間の同期を取ることができる．

通信の際のデータの受渡しは，sendプロセスとreceiveプロセスとがそれぞれパターンと呼ばれるデータをチャネルに書き込み，それらを統一化することにより行う．統一化に失敗すると，receiveの実行が失敗する．パターンには，Prologで取り扱える任意の項が使える．

チャネルの生成，チャネルに対するsend，receiveは全て組込み述語の形で記述される．以下にそれらの例を示す．

```
:- mkchan([C]), (p(C) // q(C,X)).
p(C) :- C!a.
q(C,X) :- C?X.
```

Implementing Prolog System on Parallel Prolog Machine PARK
Hideo MATSUDA[1],Tsuyoshi MASUO[1],Masaki KOHATA[2],Yukio KANEDA[1],et al.
1 Kobe University, 2 Okayama University of Science

ゴールp(C)とq(C,X)とは並列に実行され，pはaをsendし，qはそれをreceiveし変数Xに代入する．

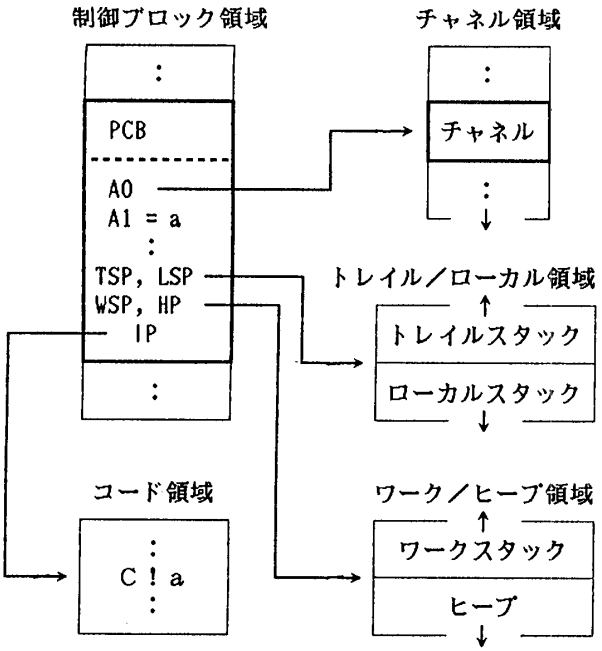
チャネルはプログラムの先頭で宣言して，以下その名前で使うこともできる．また，入出力実行用のチャネルがあらかじめ宣言されており，そのチャネルに対して入出力命令をsendすることによって，入出力を行うことができる．

```
:- io!read(X), io!write(X).
```

4．処理系の実現

4．1　プロセスの構成

プロセスは下図に示す各種の領域を持つ．図でチャネル領域は全プロセッサで共有される．制御ブロック領域とコード領域はプロセス間で共有される．ワーク／ヒープ領域とトレイル／ローカル領域はプロセスごとに独立に領域が取られる．プロセス制御ブロックには，プロセスの実行に必要な情報がまとまっており，他の領域は全てブロック中のポインタによって参照される．



PCB: ﾌﾟﾛｾｽ制御ﾌﾞﾛｯｸ　AO, A1: 引数ﾚｼﾞｽﾀ

TSP: ﾄﾚｲﾙｽﾀｯｸﾎﾟｲﾝﾀ LSP: ﾛｰｶﾙｽﾀｯｸﾎﾟｲﾝﾀ HP: ﾋｰﾌﾟﾎﾟｲﾝﾀ

WSP: ﾜｰｸｽﾀｯｸﾎﾟｲﾝﾀ　IP: ｲﾝｽﾄﾗｸｼｮﾝﾎﾟｲﾝﾀ

トレイルスタック，ローカルスタック，ヒープの持つ意味はWarrenの方式[2]と同じである．

4．2　処理系の構成

PARK-Prologの処理系は，モニタおよびコンパイラからなっている．モニタは５つの部分から構成され，ホストプロセッサ側には実行管理部および入出力管理部が，スレーブプロセッサ側にはプロセッサ管理部，プロセス管理部，メモリ管理部が置かれる．スレーブプロセッサ側の部分は，各スレーブにそれぞれ同じものがおかれる．

実行管理部，入出力管理部は，それぞれ並列実行の監視および入出力処理を行う．プロセッサ管理部はスレーブプロセッサの実行の中断・再開およびプログラムのロードなどを行う．ホストの実行管理部から割り込みにより呼び出される．プロセス管理部は，既にそのスレーブプロセッサに割り当てられたプロセスのスケジューリング，これから生成されるプロセスのプロセッサへの割当てなどを行う．

Prologの実行はインタプリタにより行われるのではなく，コンパイラによりいったんオブジェクトコード（MC68000の機械語）にコンパイルされてから行われる．全ての定義節が実行に先だってコンパイルされている必要がある．

5．おわりに

MC68000を要素プロセッサとするマルチマイクロプロセッサシステムＰＡＲＫ上で実現する並列Prolog処理系の概要とその実現の方式について述べた．現在この処理系を開発中であるが，それとは別にＧＨＣ[3]処理系の開発も並行して行い，両者の比較を行うことによって性能評価をする予定である．

[参考文献]

[1]松田秀雄，小畑正貴，増尾剛，金田悠紀夫，前川禎男：並列PrologマシンＰＡＲＫについて，Proc. of Logic Programming Conference '85, 2.4, pp.39-48(1985).

[2]Warren,D.H.D.:An Abstract Prolog Instruction Set, SRI Technical Note 309(1983).

[3]Ueda,K.: Guarded Horn Clauses, Proc. of Logic Programming Conference '85, 9.3, pp.225-236 (1985).