2Q-5

並列処理言語Pへの実行順序制御機能の導入

鈴木　秀仁　有田　隆也　曽和　将容
(名古屋工業大学)

1. はじめに

我々は，コントロールフロー原理に基づく並列計算機[1]を提案し，研究を行なっている．ソフトウェアに関しても並列処理計算機におけるオペレーティングシステムの研究が行なわれており[2]，P言語はそのような並列計算機でのシステムプログラムの記述を目的として開発されたものである[3]．

P言語における文の実行順序は，データ依存性と制御構文（繰り返し，条件選択）のコントロールにより決まるが，入出力命令について見ると，命令間に依存関係が存在しない場合が多く，並列に実行される可能性がある．このため，P言語では同一インスタンス内の入出力命令間に，直接コントロールによる順序付けを行っている．しかし，異なるインスタンスに存在する入出力命令においては，完全な順序付けを行うことはできない．

そこで本稿では，命令の実行順序付けを制御する機能のP言語への導入を試みる．

2. P言語の概要

P言語は，並列計算機のシステムプログラムの記述を目的として設計された．P言語の特徴は次の通りである．

(1) プログラマが並列性を明示的に記述することなくプログラミングをおこなうために，単一代入規則を採用している．
(2) プログラムは，関数定義の集合から構成され，プログラムの実行はmain関数から始められる．
(3) 関数は，文またはブロックの集まりからなる．また，関数の戻り値として複数の値を許す．
(4) 条件選択文としてcase文を用意し，複数の条件を並列に評価する．
(5) 繰り返し文としてiter文を用意し，繰り返しの各インスタンス内の命令を並列に処理する．
(6) システム記述のために，低レベル処理（ビット演算，メモリの直接アクセス）を導入している．
(7) ノイマン処理との互換性を守る逐次処理を導入している．

P言語の動作環境は，並列コントロールフローマシンを想定している．現在，P言語はプロトタイプの言語設計が終了し，コンパイラとエミュレータが完成しており，プログラムの記述性の評価・検討を行なっている．

3. 実行順序の明示

現状のP言語では，同一関数内の入出力命令の順序関係しか記述できない．そこで，任意の命令の順序付けを完全に行うために，コントロール型の変数を導入する．

3－1. 異なる命令間の実行順序の明示

命令の実行順序は，データの依存性と制御構文のコントロールにより決定される．しかし，ファイルや端末などへの入出力を考えると，必ずしもこれだけでは十分ではない．たとえば，画面に出力するデータ間にデータ依存性がない場合，同時に複数の出力命令が実行可能になることも考えられる．そこで，命令の実行順序をプログラマの希望通りに行う方法が必要になる．

ここで，コントロール型という新しい変数の型を導入することにより，任意の異なる命令間の順序付けを可能にする．このコントロール型変数へ代入されるのは，「文またはブロックの実行の終了を表す“コントロール”」である（図1(a)文(1)）．この例で't'は変数cnt1がコントロール型の変数であることを示している．つぎに，コントロール型変数を参照する文またはブロックの実行は，「文またはブロック内のデータ依存性が満たされ，かつ，参照しているコントロール型変数に“コントロール”が代入されたとき実行可能になる」と定めることにより，代入する文（ブロック）と参照する文（ブロック）の間に順序関係がつけられたことになる（図1(a)文(2)）．

プログラマはこのようにコントロール型変数の代入・参照関係を用いて，明示的に任意の命令間に順序関係をつけることができる．

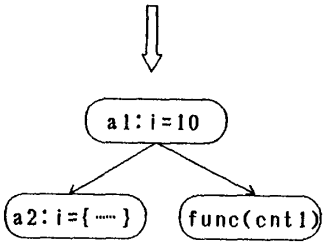
このコントロール型変数は，関数の引数としても関数からの戻り値としても使用可能であるため，異なる関数間にまたがる順序付けも記述できる（図1(a)文(3)）．

コントロール変数による順序付けのため図1(a)の文(1),(2),(3)からなるプログラムは，図1(b)のような順序で実行される．

An Introduction of the Sequence Control Function to the P Language
Hidehito SUZUKI, Takaya ARITA, Masahiro SOWA
Nagoya Institute of Technology

```
cnt1:t = (a1:i=10);  ..... (1)
(a2:i) = {
           |
         },cnt1;      ..... (2)
cnt2:t = func(cnt1); ..... (3)
```

(a) プログラム例



(b) (a)の実行順序

図1 コントロール変数

3－2. 同一命令の異なるインスタンス間の実行順序の明示

ループ本体に入出力命令が存在する場合や，入出力命令を含む関数が並列に呼び出される場合における，同一入出力命令の異なるインスタンス間の順序付けにもコントロール変数は有効である．

P言語においてループの各インスタンスはデータの依存性を守りながら，並列に実行される．このとき，ループ本体にファイルや端末の入出力命令が存在する場合に，入出力命令が同時に実行される可能性がある．この場合は，ループ本体内の各インスタンスの入出力命令間に順序付けを行う．コントロール型変数は，普通の変数と同じ性質をもつため，ループ内ではインスタンスにまたがる代入を行うことができる．このことを用いて記述したプログラムの例を示す（図2）．

```
iter {
          |
  new cnt1:t = (keyin(),cnt1);
}
```

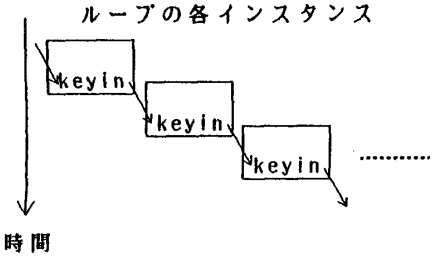


図2 ループにおける順序付け

関数の並列呼び出しの場合には，関数内の入出力命令の実行開始を指示するコントロールを引数として関数内に渡し，実行終了のコントロールを戻り値として関数の呼び出し側に返す．このような開始と終了のコントロールにより実行順序関係をつける（図3）．

```
{
        |
c2:t=func(j1,c1);
c3:t=func(j2,c2);
        |
}

cnt2:t=func(j:i,cnt1:t)
{
    cnt2:t=(crtout(j),cnt1);
}
```

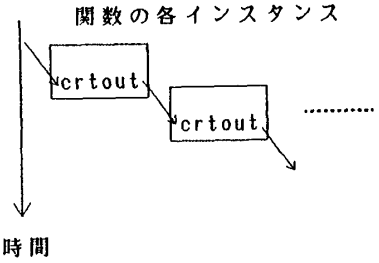


図3 関数における順序付け

4. むすび

入出力命令などの順序付けを行うために，コントロール型変数を提案をした．これにより，単一代入規則のもとで，容易に，いろいろなレベル（文，ブロック，関数）の順序関係を任意に記述することができるようになった．

参考文献

[1] Sowa,M., “Control flow parallel computer architecture”，情報処理学会，計算機アーキテクチャ研資，45-1，1982.

[2] 曽和，林，“並列オペレーティングシステム（ＳＳＳ）の基礎的考察”，情処学会第32回全国大会，1986.

[3] 奥平，曽和，“データフロー用言語P”，信学会データフローワークショップ，1987-10.