

6Y-9 ベクトル数値計算向き Prolog の提案

辰口和保 村岡洋一

早稲田大学理工学部

1. はじめに

Prologでは、assert/retractによる以外に大域的に作業用データを保存する方法がない。そこで、このような副作用を持つ作業用データ構造として配列を導入する試みがなされている。[1]では、大域変数を導入し、その集合として配列を表わしている。[2]では、配列の値をグローバル・スタック上に取り、バックトラック時に配列用領域に書き込むことでメモリアクセス数を減らし、副作用を効率的に実現している。どちらの場合も、配列要素としていかなる形式の項も許している。本論文では、特に数値計算に注目して、数値データ向け配列をPrologに導入し、これをベクトル演算機構を用いて高速処理するための言語の拡張について述べる。

2. ベクトル数値計算向きProlog

人工知能向き言語と言われるPrologは、定理証明システムやエキスパートシステムの記述のほか、最近では汎用プログラミング言語としても使われている。これらの分野ではユニフィケーションによる強力なパターンマッチング操作が有効に利用されている。一方、人工知能分野でも音声認識や、回路シミュレーション・故障診断エキスパートシステムなど、数値計算の性能をも必要とするような分野がある。数値計算用述語もPrologには用意されているが、満足行くものではない。このような数値計算用述語を整備するだけでなく、ベクトル演算機構によって数値計算を高速処理できれば、Prologの応用範囲がさらに広がることになる。

3. 言語仕様と実装方針

本論文で述べるベクトル数値計算向きPrologは、IBM 3090上のVM/PROLOGをもとに、これに配列のベクトル処理述語を拡張したものである。配列は副作用を持ち、バックトラックによって値が元に戻されない。ベクトル処理述語は、変換プログラムによりFORTRANに変換され、VS-FORTRAN自動ベクトル化コンパイラによってコンパイルされる。述語1つずつが1つのループに対応するが、連続するベクトル処理述語は、1つのループに統合できる場合が多く、FORTRANに変換した段階でループの統合を行なう。

配列とベクトル処理述語については、現在次のようなものを考えている。なお、各述語において不必要な引数は省略可能なものとする。

1)配列の宣言

書式：dec_array(array_name,type,index)
array_name：配列名として与えられるアトム
type：配列の型
index：添字の最大値

たとえば、FORTRANの宣言文

```
INTEGER A(1000)
```

は、

```
dec_array(a,integer,1000)
```

Prolog with Vector Numerical Computation Capability
Kazuyasu TATSUGUCHI, Yoichi MURAOKA
Waseda University

となる。配列は1次元のみとし、添字は1から始まるものとする。

2)単機能ベクトル演算

書式：v_op(top,end,step,mask,10,
a1,11,a2,12,a3,13)
op：演算・関数名 mask：マスクベクトル
top：ループ開始値 a1~3：配列名(又は定数)
end：ループ終了値 10~3：リストベクトル
step：刻み幅

a2とa3の演算結果をa1に代入する。マスクベクトルは、あらかじめ作られた論理値の配列である。リストベクトルもあらかじめ作っておく。加算について

```
v_add(top,end,step,mask,10,d0,a1,11,a2,12,a3,13)
は、
DO 100 I=TOP,END,STEP
  IF (MASK(LO(I))) GOTO 100
  A1(L1(I))=A2(L2(I))+A3(F3(I))
100 CONTINUE
```

を表わす。

3)比較

書式：compare(func,a1,a2,mask)
func：'<','>','='...などの比較条件
a1,a2：配列名(又は定数)
mask：比較結果によって設定される論理値ベクトル

マスクベクトルを作るのに使用する。

4)複合ベクトル演算

書式：vector(top,end,step,mask,式)
top：ループ開始値 mask：マスクベクトル
end：ループ終了値 式：複合演算式
step：刻み幅

"A(I)=SIN(B(I))*COS(C(I))"などのような式のベクトル処理を行なう。

5)圧縮・拡張

書式：compress(top,end,step,mask,a1,a2,index)
expand(top,end,step,mask,a1,a2,index)

a1からa2へmaskにしたがって配列を圧縮・拡張する。indexにはa2の最後の要素の添字が入る。

```
compress(1,100,1,mask,a1,a2,ind)
は、
```

```
IND=0
DO 100 I=1,100
  IF (MASK(I)) GOTO 100
  IND=IND+1
  A2(IND)=A1(I)
100 CONTINUE
```

である。VS-FORTRANでは圧縮・拡張がベクトル化できないため、他の命令とは別にアセンブラ・サブルーチンを用意する。

6)要素の参照・代入

書式：v_ld_element(a1,index,element) (参照)
v_st_element(element,index,a1) (代入)

配列a1のindex位置の要素elementを取り出す。または、elementを配列a1のindex位置に書き込む。

7)マスクビットの位置検出

書式：v_mask_on(mask,pos)

マスクベクトルの最初のTRUE位置をposに与える。もしFALSEだけならposは0となる。

その他、FORTRANの数値計算ライブラリを簡単に呼び出せるような述語を設ける。

4. 応用例

[3]ではコンパイル法による論理シミュレーションをベクトル計算機で行なう方法について述べている。図1の論理回路に対して、この方法を適用してシミュレーションを行ない、与えられた故障（ゲート1）と入力パターンに対して、故障検出パターンを発見するFORTRANプログラムは次のようになる。

```
10 READ (B0(1),B1(1),B2(1),B3(1),I=1,LEN)
   DO 100 I=1,LEN
     A1(I)=IAND(B0(I),B1(I))
     A2(I)=IAND(B2(I),B3(I))
     B4(I)=IOR(A1(I),A2(I))
100 CONTINUE
   DO 200 I=1,LEN
     A1(I)=I
     A2(I)=IAND(B2(I),B3(I))
     B5(I)=IOR(A1(I),A2(I))
200 CONTINUE
   DO 300 I=1,LEN
     IF (B4(I).NE.B5(I)) GOTO 400
300 CONTINUE
   GOTO 10
400 故障検出パターンの発見
   STOP
```

これをベクトル数値計算向きPrologで書くと次のようになる。

```
sim:-
  read(LEN),
  dec_array(b0,logical,LEN),
  . . . ,
  dec_array(b4,logical,LEN),
  repeat,
  v_read(b0,LEN),
  . . .
  v_read(b4,LEN),
  logic(b0,b1,b2,b3,b4,LEN),
  error(b0,b1,b2,b3,b5.LEN),
  check(b4,b5,LEN).
```

```
logic(A,B,C,D,E,N):-
  dec_array(a1,logical,N),
  dec_array(a2,logical,N),
```

```
v_and(1,N,A,B,a1),
v_and(1,N,C,D,a2),
v_or(1,N,a1,a2,E).
```

```
erroc(_,_,C,D,E,N):-
  dec_array(a1,logical,N),
  v_and(1,N,C,D,a1),
  v_or(1,N,1,a1,E).
```

```
check(X,Y,LEN):-
  dec_array(m,logical,LEN),
  compare('=',X,Y,m),
  v_mask_on(m,N),
  check1(N,LEN).
```

```
check1(0,LEN):-
  retract(num(N)),
  N1 is N+LEN,
  assert(num(N1)).
check1(N,LEN):-
  num(M),
  N1 is N+M,
  write(N1).
```

このようにして回路シミュレーション・故障シミュレーションシステムを記述できる。

5. おわりに

Prologに数値計算向き配列を導入し、ベクトル処理を行なう拡張について述べた。これによって、数値計算能力をも必要とするような問題に対しても、Prologを用いることができるようになる。

6. 参考文献

- [1]中村：Prologにおける配列と大域変数の実現について、第31回情報処理学会全国大会7E-7,pp399-400,1985.
- [2]小松、濱：プロログにおける副作用を伴う配列とその効率的実現、第35回情報処理学会全国大会4Q-6,pp745-746,1987.
- [3]石浦、安浦、矢島：ベクトル計算機による高速論理シミュレーション、情報処理学会論文誌Vol.27, No.5, pp510-517,1986.

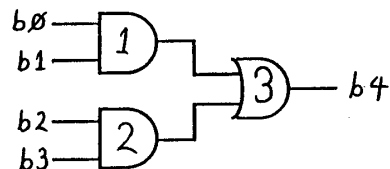


図1：サンプル論理回路