(4) return [label]

```
YRET: BSR  CLRLOC   (4)186     vide les variables locales  A0' = (TMPROCA)
      MOVE -(A0),D0            k  indicateur du sous-progr
      BEQ  ERRGRT                 return sans appel
      MOVE -(A0),D1            n  nb d'arguments
      MOVE.L -(A0),A2             après l'appel
      ADD  D1,D1
      ADD  D1,D1
      SUB  D1,A0
      CMP  #1,D0
      BEQ  MA50                -> procédure
      CMP  #5,D0
      BEQ  MA50                -> SP 68000
  fonction (3 et 4)
      MOVE.L -(A0),D1          [α | ] -> {0 ; 1 option remember i}
      BTST #0,D1
x     BEQ  GAI30
  remember i                   d1 = α
      MOVEM.L D0/A0/A2,-(SP)
                               0 -> a -> ... -> c -> d -> ∞
                               devient 0 -> d -> a -> ... -> c -> ∞
      MOVEM.L THREM,A0/A1
      MOVE (A0),d2             d
      MOVE.L (A0),d0           a
      EXT.L d0
      MOVE (a0,d2.w),d3        c
      CLR  2(a0,d3.w)          c [x | 0]
      MOVE.L D0,(a0,d2.w)      d [0 | a]
      MOVE d2,(a0,d0.w)        a [d | x]
      MOVE.L d2,(a0)     d     } 0 [c | d]
      MOVE d3,(a0)       c
      MOVE.L d1,-4(a1,d2.w)    d [α]

      BSR  REMVAR              } n° de la variable
      MOVE D2,-(SP)
      MOVE GVALU+2,d2          } pousse value
      BSR  LB321
      MOVE (SP)+,d0
      BSR  WASGN2
      MOVEM.L (SP)+,d0/a0/a2
```

```
GAI30:MOVE.L  -(A0),GVALU              ⊗
      MOVE.L  TPILE,SP
      MOVE.L  -(A0),TPILE

MA50: MOVE.L  -(A0),D1          ancien TMFOR
      LEA     TMFOR,A1
      CMP.L   (A1),D1
      BCS     ERRTLP            le S-P a terminé des boucles
      MOVE.L  D1,(A1)
                                rectifie la pile
      MOVE    -(A0),TVARLS
      MOVE.L  -(A0),D1
      BRA     MA503
MA502: MOVEM.L D0/D1/A0-A2,-(SP)
MA502: BSR    POPN
      MOVEM.L (SP)+,D0/D1/A0-A2
MA503 : MOVEM.L D0/D1/A0-A2,-(SP)
MA503: CMP    TVARN,D1
      BNE     MA502

        {  MOVEM.L D0/D1/A0-A2,-(SP)
           ADDQ #1,D1         v(d1) contient v(5) mémorisé
           BRA  MA503
        MA502:MOVE D1,-(SP)
           BSR  POPN
           MOVE (SP)+,D1
        MA503:CMP TVARN,D1
           BNE  MA502

           MOVEQ #5,D0        remet v(5) pour
           BSR  WASGN2        boucles forw/forc
           MOVEM.L (SP)+,D0/D1/A0-A2  }

      MOVE.L  D1,TVARND <
      CMP     #5,D0
      BEQ     MA510             68000
      CMP     #1,D0            { MOVEM.L A0/A1,-(SP)
      BEQ     MA51               → proc
                           <-  { BSR DEPILE2     }   (4)16      dépile
      MOVE.L  A2,A5            { MOVE.L (SP)+,A0/A1 }
      BSR     BCKTMP
      MOVE.L  A5,TDEBA5         instruction en cours
      CMP.L   YRETC,A0          pour debug (D) ok si 2 appels de
      BCC     MA505               la m̂ fonction dans une expression
      MOVE.L  SP,YRETC
MA505:RTS
MA51: BSR     BCKTMP
      BSR     DECTMN
      BNE     YGOTO       → return label
      MOVE.L  A2,A5 <---- BSR DEPILE2  (4)16  ⊗  dépile
      BRA     LB93  RTS ok
MA510:BSR     BCKTMP            retour d'appel dans code 68000
      BSR     LB920
      BSR     MA512
      JMP     (A2)

DEPILE2:BSR   LB920     }  ⊗
      BRA     DEPILE1   }
```