# ある型のDOループ・プログラムに対する配列限界オーバ検査システム

羽賀隆洋・杉野忠・福村晃夫
（名古屋大学）

## 1. まえがき

種々の仮定をされたDOループ・プログラムに対してではあるが，配列限界オーバの必要十分条件が先に導かれた[1]。本論文では，その必要十分条件に基づく検査システムの実現，検査例，実際のフォートラン・プログラムに対する適用性などを主体として述べる。

なお，検査対象プログラムに対する主な仮定は次の３つである：(i) DOループからの飛出しがないこと，(ii) 検査には線形な式のみが関係すること，(iii) 検査に関係する，DOループ内の変数には，DOループの制御変数及びDOループ内で値が変化しない変数のみが関係すること，すなわち，後向記号的実行[2]がDOループに対して１回で収束すること（厳密には2.参照）。なお，変数が扱かう値は整数のみとする。

このようにかなり強い仮定を置いているにもかかわらず，実際に用いられているかなりのフォートラン・プログラムに対して検査可能となることが5.において示される。

なお，本検査法における検査手順の妥当性の証明はすべて省略する（文献[1]参照）。

## 2. 検査システムのあらまし

### 2.1 検査対象プログラムに対する仮定

検査対象プログラムは，整数値のみを扱かうフォートランの１つのプログラム単位とし，次の仮定が置かれる。なお，これ以後，フォートラン用語がことわりなく用いられる。

〔仮定〕

(1) 使用命令は以下の文のみである：
代入文，算術IF文，GO TO文，DO文，CONTINUE文，STOP (RETURN)文，DIMENSION文，COMMON文，SUBROUTINE文。

(2) 飛出しなしの，DOループ・プログラムである。

(3) 添字変数，IF文判別変数，DO文の初期値，終値パラメータには，線形な式のみが代入され得る。

(4) 添字変数，IF文判別変数，DO文の初期値，終値パラメータに対する後向追跡（後向記号的実行[2]）は，どのDOループに対しても１回で収束する（２回目に収束したと判定される）。但し，追跡対象を内部に含まないDOループに対しては，追跡は全く変化しない。

以上が本質的仮定である。仮定(3),(4)のさらに厳密な意味は，後の検査手順１自身により与えられる。なお，仮定(3),(4)は，一般性を失うことなく，プログラムに現れる添字式，IF文判別式及びDO文の初期値，終値パラメータは定数，単純変数であるとしてよいことを考慮して述べられている。

さらに検査手順の記述及び検査システムの実現を容易にするため，検査対象プログラムは次の性質を満たすように，あらかじめ前処理がなされているものとする（常に可能）：(i) DO文は増分１（省略値）に限り，対応するCONTINUE文をか

ならず付ける，(ii) 各DOループの制御変数は，通常の変数とは異なる，一意の固有の名前をもつ，(iii) 代入文右辺の式は，高々1つの四則演算子を含む，など。

## 2.2 初期条件

検査システムは，2.1の仮定を満たすフォートラン・プログラムと初期条件（$\mathcal{L}_0$で表わす）を入力とする。先と同様に，検査システム作成の便宜上，$\mathcal{L}_0$は式(1)の基本条件のAND形のみとする。

$$C_\ell \leqq V \leqq C_u \quad (V\text{は単純変数}, C_\ell, C_u\text{は(整)定数}) \tag{1-1}$$

$$C_{\ell 1} \leqq \alpha(\beta \in D_1) \leqq C_{u1}, \cdots, C_{\ell t} \leqq \alpha(\beta \in D_t) \leqq C_{ut} \tag{1-2}$$

（$\alpha$は配列名，$\beta \equiv (\beta_1, \cdots, \beta_k)$は添字対
$D_1, \cdots, D_t$は互に*disjoint*，かつ，$D_1 \cup \cdots \cup D_t = \{\beta\text{の全集合}\}$）

ここに，$\beta \in D$は次の形とする：$V_{\ell 1} \leqq \beta_1 \leqq V_{u1}, \cdots, V_{\ell k} \leqq \beta_k \leqq V_{uk}$ (1-3)

（各$V_\ell, V_u$は定数，又は，単純変数）

## 2.3 検査法のあらまし

本論文で提案される配列限界オーバの検査システムは，2.1，2.2で述べられたフォートラン・プログラム及び初期条件$\mathcal{L}_0$を入力とし，「オーバなし」，「オーバあり」，「検査不能」の3種のいずれかの検査結果を出力する。

検査手順は簡単のため，指定された位置の添字$V$及び入口から添字$V$の出現命令までのパス$P$（正確には文献(1)参照）の組を任意に一組固定して述べられる。もちろん，実際には，$V$，$P$のあらゆる組に対して検査が行われる必要がある。

なお，添字$V$が(整)定数である場合は無条件に検査可能とする。

(例) 下図中の配列要素IA(K,3)の添字Kまでのパスは，以下の$P_1$，$P_2$の2つである：

```
   DO 10 ···
   ---
10 CONTINUE
   IF(N) 1, 1, 2
 1 DO 20 ···
   DO 30 ···
   ---
30 CONTINUE
   K=L+1
   J=IA(K,3)-2
   ---
20 CONTINUE
 2 ---
   ---
```

$P_1$ = (DO-BLOCK1, IF(N)···, DO 20 ···,
DO-BLOCK2, K=L+1)
IF(N)··· の推移条件は —

$P_2$ = (DO-BLOCK1, IF(N)···, DO 20 ···,
DO-BLOCK2, K=L+1)
IF(N)··· の推移条件は 0

さて，検査手順は次の3つの手順に大別される。但し，プログラムに対する仮定(1)，(2)はすでにチェックされているものとする。

【手順1】仮定(3)，(4)が満たされるかどうかの判定及び検査に関係する命令の決定。

【手順2】配列限界オーバの必要十分条件を表わす条件Q（いくつかの連立一次不等式系）の作成。

【手順3】条件Qに対する可能解（整数解）の存在性判定及びその結果に基づく検査結果の出力。

## 2.4 記号的実行

検査手順2において後向記号的実行(後向追跡)[2]が用いられる。命令系列$S = (A_1, \cdots, A_n)$に対する，式$Y$の後向追跡結果を$\mathcal{G}_S(Y) \equiv \mathcal{G}_{A_1}(\mathcal{G}_{A_2}(\cdots(\mathcal{G}_{A_n}(Y))\cdots))$で表わす。但し，追跡不能の場合は$\mathcal{G}_S(Y) \equiv \varepsilon$とする。又，一般に$\mathcal{G}_A(Y)$は，$A$が代入文でないとき$\mathcal{G}_A(Y) \equiv Y$とし，$A$が代入文のときは文献(2)参照。

## 3. 検査手順

### 3.1 手順1

【定義1】

(i) $A$ が代入文のとき: $l(A) \triangleq$ (左辺の単純変数, 又は, 配列要素)
$R(A) \triangleq$ {右辺の単純変数, 及び, 配列要素(添字となっている変数を含む)} (2)

(ii) $A$ がIF文のとき: $l(A) \triangleq$ (判別変数)
$R(A) \triangleq$ {判別変数} (3)

(iii) $A$ がDO文のとき: $l(A) \triangleq$ (制御変数)
$R(A) \triangleq$ {定数以外の, 初期値, 終値パラメータ} (4)

与えられた検査対象添字を$V$, 入口から添字$V$に至るパスを$P=(A_1,\cdots,A_{k-1})$($A_k$が添字$V$の出現命令)とする。

(手順1-1) $i=k-1$, $T_{i+1}=T_k=\{V\}$とする。但し, $i=0$(すなわち, $P$が空系列)ならば, 仮定(3), (4)が満たされているとして終了。

(手順1-2) $A_i$, $T_{i+1}$から$T_i$を次のように求める。

(i) $A_i$が代入文のとき:

$$T_i=\begin{cases}(T_{i+1}-\{l(A_i)\})\cup R(A_i) & (l(A_i)\text{が単純変数かつ}l(A_i)\in T_{i+1}\text{, 又は, }l(A_i)\text{が配列要素}\alpha(\beta)\text{かつ}T_{i+1}\text{中の}\alpha(\cdots)\text{はすべて}\alpha(\beta)\text{,のとき})\\ T_{i+1} & (l(A_i)\text{が単純変数かつ}l(A_i)\notin T_{i+1}\text{, 又は, }l(A_i)\text{が配列要素}\alpha(\beta)\text{かつ}T_{i+1}\text{中に}\alpha(\cdots)\text{は全く現れない, とき})\\ \varepsilon & (\text{その他のとき})\end{cases} \quad (5)$$

$T_i=\varepsilon$となるとき検査不能として終了。式(5)の第1番目の置き換えが行われたとき, $A_i$の右辺が線形でなければ仮定(3)が満たされないとして終了, 線形ならばこの$A_i$を関係命令とする。

(ii) $A_i$がGO TO文のとき: $T_i=T_{i+1}$とする。

(iii) $A_i$がIF文のとき: $T_i=T_{i+1}\cup R(A_i)$とする。なお, IF文$A_i$は無条件に関係命令とする。

(iv) $A_i$がDO文のとき: ある$x\in T_{i+1}$に対して, $x$, あるいは, $x$が配列名$\alpha$の配列要素のときその配列名$\alpha$の配列要素$\alpha(\cdots)$を左辺とする代入文が, このDOループ内に存在すれば, 仮定(4)が満たされないとして終了。その他のとき,

$$T_i=\begin{cases}(T_{i+1}-\{l(A_i)\})\cup R(A_i) & (l(A_i)\in T_{i+1}\text{のとき})\\ T_{i+1} & (l(A_i)\notin T_{i+1}\text{のとき})\end{cases} \quad (6)$$

とする。式(6)の前者の場合, $A_i$を関係命令とする。

(v) $A_i$がDO-BLOCK$j$のとき: まず, 上記(iv)の前半と同様に, DO-BLOCK$j$内で$T_{i+1}$中の変数が変化する可能性があるかどうか調べ, 変化する可能性があれば仮定(4)が満たされないとして終了。その他のとき$T_i=T_{i+1}$とする。

(手順1-3) $i=i-1$とする。$i=0$ならば(このとき, $T_1$は求められている), 仮定(3), (4)が満たされるとして終了。$i\geqq 1$ならば手順1-2へ。

なお, この手順1において配列要素が関係しない場合には, 各$T_i$をビット・ベクトルで表わすことにより高速に処理が行われる。

### 3.2 手順2

【定義2】 一般に, $A_i$に現われる1つの単純変数を$x$, 入口から$A_i$までのパスを$P=(A_1,\cdots,A_{i-1})$としたとき,

$$\tilde{\chi} \triangleq \varphi_{(A_1,\cdots,A_{i-1})}(\chi) \qquad (7)$$

と定める。

この$\tilde{\chi}$は，追跡不能でないとき，入口時点の変数及びDO文の制御変数のみを用いて表わされる。

(手順2－1) まず，条件Qを次式(8)により作成する。

$$Q = l_0 \wedge [(\tilde{v} \leqq 0) \vee (\tilde{v} \geqq \mathrm{Limit}+1)] \wedge \bigwedge_{i=1,m} (\tilde{u}_i \gtreqless 0)$$
$$\wedge \bigwedge_{i=1,n} [((w_{ci} = \tilde{w}_{li}) \wedge (\tilde{w}_{li} > \tilde{w}_{ui})) \vee ((\tilde{w}_{li} \leqq w_{ci} \leqq \tilde{w}_{ui}) \wedge (\tilde{w}_{li} \leqq \tilde{w}_{ui}))] \qquad (8)$$

ここに，Limitは添字vを出現させる配列要素の配列寸法(整合配列の場合は変数)，mはパスP上のIF文の個数，$u_1,\cdots,u_m$は m個のIF文の判別変数，$u_i \gtreqless 0$ は $u_i>0, u_i=0, u_i<0$ のいずれか (for each $i=1,\cdots,m$)，nは関係DO文の個数，$w_{ci}, w_{li}, w_{ui}$は各々第i関係DO文の制御変数，初期値，終値パラメータを表わす。

(手順2－2) 上記の手順2－1において作成された条件Qにおいて，初期条件$l_0$部分以外に配列要素が現われなければ条件Qが完成されたとして終了 (そのとき，配列要素に対する初期条件があれば，それは不要である)。その他のとき，以下のようにQに現れる配列要素を順次処理する：Qの$l_0$部分以外に現われる，他の添字となっていない配列要素 (最外配列要素と呼ぶ) の配列名を任意に1つ選んでαとする。そして，そのようなα(…)の出現の全種類を$\alpha(\beta^{(1)}),\cdots,\alpha(\beta^{(g)})$とし，

$$Q = \bigvee_{(i_1,\cdots,i_g)\in\{1,\cdots,t\}^g} Q^{(i_1,\cdots,i_g)} \qquad (9)$$

を次のように作る。すなわち，$Q^{(i_1,\cdots,i_g)}$を作るため，

(i) $l_0$部分中の各$C_{li} \leqq \alpha(\beta) \leqq C_{ui}$ ($\beta \in D_i$, $i=1,\cdots,t$) を $C_{li} \leqq \alpha_i \leqq C_{ui}$で置き換える (但し，すでにこの置き換えがなされている場合には，この操作は不要)。

(ii) 各条件$\beta^{(j)} \in D_{i_j}$をANDして付加する ($j=1,\cdots,g$)。

(iii) 各出現$\alpha(\beta^{(j)})$をそれぞれ$\alpha_{i_j}$に置き換える ($j=1,\cdots,g$)。

(手順2－3) 上記の各$Q^{(i_1,\cdots,i_g)}$をQとみなして，手順2－2を繰返す。そして，すべての配列要素が処理された段階の$Q^{(\cdots)}$をすべてORして，求める条件Qとする。なお，このとき$l_0$部分に，(i)の処理がなされていない，配列に対する初期条件が残されていれば，それを不要として取り去る。

### 3.3 手順3

上記の手順2－2で導入された，各配列名αに対する各$\alpha_i$を各々異なる単純変数とみなして，条件Qに可能解が存在するかどうか調べる。条件Qが可能解をもたないとき，「添字vにおいて，パスPに対応する実行経路(1)を通っては，配列限界オーバは生じない」として終了。条件Qが可能解をもつとき，「添字vにおいて，パスPに対応する実行経路を通って，配列限界オーバが生じる」(可能解がその初期値例である) として終了。

なお，(i) 仮定が満たされない，(ii) 後向追跡が不能となる，(iii) 条件Qにおいて可能解の存在性が判定不能である，のいずれかの場合には，「添字vに対して，パスPについては検査不能」とする。

## 4. 検査システム

以下では，3.の検査手順に基づく検査システムの実現について述べるが，それに関連して，次の2点についてまずふれる：(1) 配列要素が関係しないときの通過不能パスの判定手順追加，(2) 配列要素が関係するときの特別な初期条件の場

合の簡便法。

なお，条件Qに対する可能解の存在性判定法については，文献(1)参照。

### 4.1 通過不能パスの判定

検査に配列要素が関係しない場合には，変数の値範囲情報を作成することによって，パスpが通過不能（すなわち，パスpに対応する実行経路は，$I_0$のもとでは決して生じない）であると知られる場合がある。なお，この判定は手順1と手順2との間に行なうものとし，パスpが通過不能と判定された場合は，手順2以降を行うことなく，「p, pの組に対してオーバなし」として終了できる。又，パスpが通過不能と判定されない場合で，この段階でオーバなしと判定されることもあり得る。但し，上記のように判定されて，手順2以降が省略できる場合はそれほど多くなく，従って，その判定が高速に行われる場合にのみこの手順追加は有効である（4.3参照）。

### 4.2 配列要素が関係する場合の簡便法

各配列に対する初期条件が，通常よく生ずると考えられる次式(10)の特別な形をしている場合には，条件Qをもっと簡便な形で作ることができる。

$$\left.\begin{array}{c} C_\ell \leqq \alpha(\beta) \leqq C_u \quad (\beta \in D) \\ (\beta \triangleq (\beta_1, \cdots, \beta_k) \in D) = (1 \leqq \beta_1 \leqq Limit_{\alpha,1}) \wedge \cdots \wedge (1 \leqq \beta_k \leqq Limit_{\alpha,k}) \end{array}\right\} \quad (10)$$

ここに，各$Limit_{\alpha,i}$は，配列$\alpha$の第i添字の寸法とする（$i=1,\cdots,k$）。

【手順1】 3.1の手順1に同じ。

【手順2】

（手順2-1） 3.2の手順2-1に同じ。

（手順2-2） 上記で得られたQに現れる各最外配列要素$\alpha(\cdots)$を単に$\alpha$で置き換える。

【手順3】 プログラムに現れる添字ひ，入口からひまでのパスpのあらゆる組について手順1,2を行い，どの条件Qも可能解をもたないとき，$I_0$を満たす初期条件のもとではプログラムのどの添字においてもオーバしないとして終了（より詳細については，文献(1)の妥当性3参照）。

### 4.3 検査システムの実現

以上の方法に基づく検査システムがFACOM 230-38上に実現され，いくつかの実際のフォートラン・プログラムに対して適用された。システム記述言語はフォートランであり，次の処理部に大別される：

(1) 木操作部：式の表現及び部分式の他の式への置き換え操作などのための，一般的木操作を行う。

(2) 記号的実行部：木操作部をもとに2.4の記号的実行を行う。若干の数学的知識（式の簡略化など）を含む。

(3) プログラム解析部：プログラムの各種基本的情報を内部表現に変換する。

(4) IP（整数計画法）に対する可能解存在性判定部：切除平面法，実行不可能量の最小化等からなる。

(5) 検査部：本方法に基づく，配列限界オーバの検査本体（すなわち，手順1~3，その他）。

システムの規模は，約4000ステップである。検査時間は添字出現位置，パス上のIF文数，関係DO文数によりかなり変化するが，1つの添字ひ当り高々20~30秒程度であった。次に，検査時間に関する若干のデータ値を挙げる（いずれも，検査に配列要素が関係しない場合である）。

| サブルーチン名 | 添字位置 | 検査時間 (秒) | | | | |
|---|---|---|---|---|---|---|
| | | $t_1$ | $t_2$ | $t_3$ | $t_4$ | 計 |
| FILTIC[3] | 1 | 2.9 | 0.5 | 0.2 | 0.8 | 4.4 |
| | 2 | 2.9 | 0.6 | 0.2 | 0.8 | 4.5 |
| | 3 | 2.9 | 0.6 | 0.3 | 0.8 | 4.6 |
| | 4 | 2.8 | 0.6 | 0.3 | 0.7 | 4.6 |
| | 5 | 3.2 | 0.7 | 0.3 | 0.7 | 4.9 |
| MAXF3I[3] | 1 | 1.3 | 0.2 | 1.8 | 11.1 | 14.4 |
| | 2 | 1.3 | 0.2 | 1.7 | 5.5 | 8.7 |

ここに，$t_1$：プログラムの基本情報取出し
$t_2$：手順1
$t_3$：通過不能パスの判定
$t_4$：手順2

但し，各サブルーチンの各添字位置は，次の表で与えられるような位置である。

| サブルーチン名 | 添字位置 | N | K | NF | ND | NC |
|---|---|---|---|---|---|---|
| FILTIC | 1 | 7 | 2 | 0 | 1 | 83 |
| | 2 | 34 | 2 | 0 | 1 | 83 |
| | 3 | 49 | 7 | 0 | 1 | 83 |
| | 4 | 63 | 3 | 0 | 0 | 83 |
| | 5 | 77 | 3 | 0 | 0 | 83 |
| MAXF3I | 1 | 13 | 11 | 2 | 2 | 36 |
| | 2 | 13 | 10 | 2 | 1 | 36 |

ここに，N：添字位置までのプログラムテキスト上の命令数
K：添字位置までの関係命令数
NF：添字位置までのIF文数
ND：添字位置までの関係DO文数
NC：プログラム全体の命令数

## 5. 検査例

ここでは，本検査法の具体的適用例，及び，本検査法の実際のフォートラン・プログラムに対する適用可能性の実験結果を述べる。

### 5.1 具体的適用例

この検査システムの木操作部分で用いられているサブルーチンTRANSTに対する検査例：

```
(Δi)
      SUBROUTINE TRANST(ITREE,NTREE,NO,ITW,NTW)
      DIMENSION ITREE(10,100),ITW(10,100)
      SYOKI ZYOKEN NTREE.LE.100
      SYOKI ZYOKEN ITREE(3,1:NTREE).LE.7
01    NTW=NTREE
02    IF(NO) 1,1,2
03  2 IF(NTREE) 1,1,3
04  3 DO 1000 I1=1,NTREE
05    N=ITREE(3,I1)+3
06    DO 2000 I2=1,N
07    ITW(I2,I1)=ITREE(I2,I1)
08 2000 CONTINUE
09    ITW(2,I1)=ITW(2,I1)+NO
10    ITW(2,I1)=ITW(2,I1)-1
11    IV=N-4
12    IF(IV) 1000,4,4
13  4 DO 3000 I3=4,N
14    ITW(I3,I1)=ITW(I3,I1)+NO
15    ITW(I3,I1)=ITW(I3,I1)-1
16 3000 CONTINUE
17 1000 CONTINUE
18    1 RETURN
      END
```

添字ひとして第7命令の右辺の添字I2とする。パスは $p=(\lambda_1,\cdots,\lambda_6)$ のみである。検査手順2-1で得られる条件Qは，

$$
\begin{aligned}
Q=&(\mathrm{NTREE}\leqq 100)\wedge(\mathrm{ITREE}(3,1{:}\mathrm{NTREE})\leqq 7)\\
&\wedge((\mathrm{I2}\leqq 0)\vee(\mathrm{I2}\geqq 11))\wedge(\mathrm{NO}\geqq 1)\wedge(\mathrm{NTREE}\geqq 1)\\
&\wedge[(\mathrm{NTREE}\leqq 0)\wedge(\mathrm{I1}=1))\vee((1\leqq\mathrm{NTREE})\wedge(1\leqq\mathrm{I1}\leqq\mathrm{NTREE}))]\\
&\wedge[((\mathrm{ITREE}(3,\mathrm{I1})+3\leqq 0)\wedge(\mathrm{I2}=1))\vee((1\leqq\mathrm{ITREE}(3,\mathrm{I1})+3)\wedge(1\leqq\mathrm{I2}\leqq\mathrm{ITREE}(3,\mathrm{I1})+3))]
\end{aligned}
$$

次に，検査手順2-2を行うが，その際，初期条件 $\mathrm{ITREE}(3,1{:}\mathrm{NTREE})\leqq 7$ は

$$
\left.\begin{aligned}
&-\infty\leqq \mathrm{ITREE}(\beta)\leqq 7 \quad (\beta\in D_1)\\
&-\infty\leqq \mathrm{ITREE}(\beta)\leqq +\infty \quad (\beta\in D_2)
\end{aligned}\right\}
$$

（但し，$D_1=\{(i,j)\mid i=3,\ 1\leqq j\leqq \mathrm{NTREE}\}$，$D_2=\{(i,j)\text{の全集合}\}-D_1$）

を意味することに注意する。さて，配列名αとしてITREEとすれば，$\alpha(\beta^{(1)}),\cdots,\alpha(\beta^{(q)})$ は $\alpha(3,\mathrm{I1}),\ q=1$ となる。従って，式(10)は $Q=\bigvee_{(i,j)\in\{1,2\}^1} Q^{(i,j)}=Q^{(1)}\vee Q^{(2)}$ となり，

$$
\begin{aligned}
Q^{(1)}=&(\mathrm{NTREE}\leqq 100)\wedge(\underline{\mathrm{ITREE1}\leqq 7})\wedge((3=3)\wedge(1\leqq\mathrm{I1}\leqq\mathrm{NTREE}))\\
&\wedge((\mathrm{I2}\leqq 0)\vee(\mathrm{I2}\geqq 11))\wedge(\mathrm{NO}\geqq 1)\wedge(\mathrm{NTREE}\geqq 1)\\
&\wedge[(\mathrm{NTREE}\leqq 0)\wedge(\mathrm{I1}=1))\vee((1\leqq\mathrm{NTREE})\wedge(1\leqq\mathrm{I1}\leqq\mathrm{NTREE}))]\\
&\wedge[((\underline{\mathrm{ITREE1}}+3\leqq 0)\wedge(\mathrm{I2}=1))\vee((1\leqq\underline{\mathrm{ITREE1}}+3)\wedge(1\leqq\mathrm{I2}\leqq\underline{\mathrm{ITREE1}}+3))]
\end{aligned}
$$

$$
\begin{aligned}
Q^{(2)}=&(\mathrm{NTREE}\leqq 100)\wedge((3\neq 3)\vee(\mathrm{I1}\leqq 0)\vee(\mathrm{I1}\geqq\mathrm{NTREE}+1))\\
&\wedge((\mathrm{I2}\leqq 0)\vee(\mathrm{I2}\geqq 11))\wedge(\mathrm{NO}\geqq 1)\wedge(\mathrm{NTREE}\geqq 1)\\
&\wedge[(\mathrm{NTREE}\leqq 0)\wedge(\mathrm{I1}=1))\vee((1\leqq\mathrm{NTREE})\wedge(1\leqq\mathrm{I1}\leqq\mathrm{NTREE}))]\\
&\wedge[((\underline{\mathrm{ITREE2}}+3\leqq 0)\wedge(\mathrm{I2}=1))\vee((1\leqq\underline{\mathrm{ITREE2}}+3)\wedge(1\leqq\mathrm{I2}\leqq\underline{\mathrm{ITREE2}}+3))]
\end{aligned}
$$

これで検査手順2は終了する。次に，$(3=3)=t$，$(3\neq 3)=f$ に注意して，このQを積和形に展開すれば，

$$Q=Q_1\vee\cdots\vee Q_{24}$$

$$
\begin{aligned}
Q_1=&(\mathrm{NTREE}\leqq 100)\wedge(\mathrm{ITREE1}\leqq 7)\wedge(1\leqq\mathrm{I1}\leqq\mathrm{NTREE})\\
&\wedge(\mathrm{I2}\leqq 0)\wedge(\mathrm{NO}\geqq 1)\wedge(\mathrm{NTREE}\geqq 1)\\
&\wedge(\mathrm{NTREE}\leqq 0)\wedge(\mathrm{I1}=1)\wedge(\mathrm{ITREE1}+3\leqq 0)\wedge(\mathrm{I2}=1)\\
Q_2=&(\mathrm{NTREE}\leqq 100)\wedge(\mathrm{ITREE1}\leqq 7)\wedge(1\leqq\mathrm{I1}\leqq\mathrm{NTREE})\\
&\wedge(\mathrm{I2}\leqq 0)\wedge(\mathrm{NO}\geqq 1)\wedge(\mathrm{NTREE}\geqq 1)\\
&\wedge(\mathrm{NTREE}\leqq 0)\wedge(\mathrm{I1}=1)\wedge(1\leqq\mathrm{ITREE1}+3)\wedge(1\leqq\mathrm{I2}\leqq\mathrm{ITREE1}+3)\\
&\vdots\\
Q_4=&(\mathrm{NTREE}\leqq 100)\wedge(\mathrm{I1}\leqq 0)\wedge(\mathrm{I2}\leqq 0)\wedge(\mathrm{NO}\geqq 1)\wedge(\mathrm{NTREE}\geqq 1)\\
&\wedge(\mathrm{NTREE}\leqq 0)\wedge(\mathrm{I1}=1)\wedge(\mathrm{ITREE2}+3\leqq 0)\wedge(\mathrm{I2}=1)\\
&\vdots
\end{aligned}
$$

これらの $Q_1,\cdots,Q_{24}$ はいずれも可能解をもたず，従って，Qに可能解なし(オーバなし)。

## 5.2 検査可能性の実験結果

ここでは，実際のフォートラン・プログラムに対して，検査可能となる添字の割合[+]を示す(オーバあり，なしは $l_0$ に依存することに注意)。1番から5番までは SLIP 中の画像フィルター[(3)]，6番から15番まではこの検査システム自身で用いられたサブルーチン(主として，式などを表わすための木操作)である。但し，明らかに本論文の仮定を満たさないと知られるプログラムは除外した。

+ 大略，仮定(3)，(4)を満たす添字の割合といえる。

| No. | サブルーチン名 | AN | TN | TN' | TN/AN (%) | TN'/AN (%) |
|---|---|---|---|---|---|---|
| 1 | FILTI7 | 20 | 20 | 20 | 100 | 100 |
| 2 | FILTIC | 37 | 37 | 37 | 100 | 100 |
| 3 | FILT3I | 52 | 52 | 52 | 100 | 100 |
| 4 | MAXF2I | 34 | 12 | 12 | 35 | 35 |
| 5 | MAXF3I | 10 | 2 | 4 | 20 | < 40 |
| 6 | EQT | 6 | 1 | 1 | 17 | 17 |
| 7 | COPYT | 6 | 2 | 6 | 33 | < 100 |
| 8 | CNVAST | 13 | 13 | 13 | 100 | 100 |
| 9 | INALEF | 5 | 4 | 4 | 80 | 80 |
| 10 | INATOM | 3 | 2 | 2 | 67 | 67 |
| 11 | TAKASA | 6 | 4 | 5 | 67 | < 83 |
| 12 | TRANST | 14 | 4 | 14 | 29 | < 100 |
| 13 | GETT | 8 | 3 | 3 | 38 | 38 |
| 14 | PUTT | 9 | 4 | 9 | 44 | < 100 |
| 15 | CVNORD | 14 | 10 | 11 | 77 | < 79 |

AN: プログラムに現れる添字の総数
TN: 配列要素が検査に関係しないとき，検査可能となる添字の個数
TN': 配列要素が関係する場合を考慮した手順で，検査可能となる添字の個数

なお，TN'/ANが100%にならない場合の，その主な原因は次のごとくであった：(i) DOループ中で変化する変数が添字の値に関係する，(ii) DOループからの飛出しがある，(iii) 一般のループを含む，(iv) サブルーチンのCALLがある，(v) 初期条件$J_0$として式(1)の範囲以外の形のものを必要とする，など。

## 6. あとがき

本論文で提案された配列限界オーバ検査システムは種々の仮定（特に，仮定(4)）のもとではあるが，配列限界オーバの必要十分条件に基づくものである。但し，そのような仮定を緩められるかどうかについては，原理的にはともかく，事実上不可能に近いといえよう。

従って，今後の方向としては，本格的検証能力を取り入れた検査システムを考えるとともに，本検査法のような簡便な検査を前段階で行なうこと及び通常の具体的数値によるテストを併用すること，などを考慮したより大局的観点からの検査システムの見直し，再構成が重要となろう。

謝辞：おわりに，種々の有益な御指摘をいただいた本学吉田雄二助教授，ならびに，御討論いただく研究室の皆さんに感謝する。

【文献】

(1) 羽賀，杉野，福村："ある型のDOループ・プログラムに対する配列限界オーバの必要十分条件とそれに基づく検査システム"，信学技報 AL82-43 (1982).

(2) 羽賀，福村："プログラムの記号的実行における代入関係追跡に関する基礎的一考察"，電子通信学会論文誌，63-D [1]，p.101 (1980).

(3) 鳥脇，福村："画像処理サブルーチンライブラリSLIPについて —— 画像処理機能の一覧 ——"，情報処理学会研資コンピュータビジョン 1-2 (1979).