ALGORITHM VERIFICATION APPLIED TO THE TODD-COXETER ALGORITHM

by

Carolyn T. Williamson and W. D. Maurer

Memorandum No. ERL-M317

2 December 1971

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

Research sponsored by the National Science Foundation, Grant GJ-821.

1. Introduction

Recent work by Maurer has provided a framework and tools for analysis of computer programs and rigorous proof that a program works correctly. The purpose of this report is to give a detailed and nontrivial example of the application of these methods. The basic concepts, definitions and theorems are summarized below. For proofs and further details see (Maurer 3 and 4). The idea is to show that if a program is started with variables satisfying a given set of conditions the result is well defined (the program terminates) and the final values of the variables satisfy a second given set of conditions.

Definition: A p-set is any set of the form $\mathcal{S} = \prod_{x \in M} V_x$, where M is a set such that there is a set, V_x, of values for each x in M. Π is the cartesian product.

Definition: A condition is a subset $\mathcal{J} \subset \mathcal{S}$. If $S \in \mathcal{S}$, then S satisfies $\mathcal{J}$ if $S \in \mathcal{J}$. Denote by S(x) the value of the x component of S $(S(x) \in V_x)$.

Example: Let $M = \langle I, N, A(20) \rangle$, $V_I = V_N = \langle 1 <= I,\ N <= 20 \rangle$, $V_{A(i)} = \langle$ floating point numbers that can be represented in a computer word $\rangle$ for $1 <= i <= 20$. Then form the p-set on M, $\mathcal{S} = \prod_{x \in M} V_x$. Further let $Q = \langle F_i;\ 1 <= i <= n,$ F_i an executable instruction $\rangle$ and define λ to be the program counter. $V_\lambda = Q$ and λ is the next instruction to be executed. Form a new p-set on $M \cup \lambda$, $\mathcal{T} = \mathcal{S} \times Q = \prod_{x \in M \cup \lambda} V_x$. Such p-sets play an important role in the theory to be used.

Example: Let $Q = \langle F_i;\ 1 <= i <= 7 \rangle$ with F_i as follows

```
F1   I = 1
F2   N = 20
F3   A(1) = 0.0
F4   A(I+1) = A(I) + 0.999
F5   I = I + 1
F6   If (I.LT.N) GO TO F4
F7   STOP
```

Each F_i is a function from $\mathcal{S}$ to $\mathcal{T} = S \times Q$ and Q is a program on $\mathcal{S}$, ($\mathcal{S}$ and $\mathcal{T}$ are as defined above). F_i can be expressed in terms of its components $P_i : \mathcal{S} \longrightarrow \mathcal{S}$ and $N_i : \mathcal{S} \longrightarrow Q$. N_i is called the next statement function. Thus for $S \in \mathcal{S}$, $P_1(S) = S'$ where $S'(x) = S(x)$ if $x \neq I$ and $S'(I) = 1$; $N_1(S) = F_2$; $N_6(S) = F_4$ if I $(= S(I)) < N$ $(= S(N))$ and $N_6(S) = F_7$ if $I >= N$. F_7 is interpreted as P_7 = the identity on $\mathcal{S}$ and N_7 is nowhere defined. A program reaching F_y where N_y is undefined for $S \in \mathcal{J} \subset \mathcal{S}$ will terminate at F_y if $S \in \mathcal{J}$.

Definition: If F is a function from $\mathcal{S}$ to $\mathcal{S}$, F has effective domain $\Delta(F) \sim \langle x \in M$ such that a change in $S_1(x)$ may change $F(S_1) = S_2 \rangle$ and F has effective range $\rho(F) \sim \langle x \in M$ such that F may change the value of x, i.e, $S_1(x) \neq S_2(x) \rangle$
For a more precise definition see Maurer (3).
If $\mathcal{S}' \subset \mathcal{S}$ is a condition $\mathcal{S}'$ has an effective domain $\Delta(\mathcal{S}')$ defined by considering $\mathcal{S}'$ as a function from $\mathcal{S}$ to $\langle T,F \rangle$ such that $\mathcal{S}'(S) = T$ if $S \in \mathcal{S}'$, $\mathcal{S}'(S) = F$ otherwise.

Example continued: $\Delta(P_1) = \Phi$ (the null set), $\rho(P_1) = I$, $\Delta(N_6) = \langle I,N \rangle$.

Definition: A program P on $\mathcal{J} = \mathcal{S} \times P$ is a finite set $\langle F_i \rangle$, $F_i = (P_i, N_i)$ functions on $\mathcal{J}$. It is assumed that P contains a function F_x with P_x = identity and N_x nowhere defined and that N_i is everywhere defined for $i \neq x$. This can easily be accomplished without changing the effect of P on $\mathcal{S}$ by setting $N_i(S) = F_x$ if $N_i(S)$ is undefined. The computation sequence in P of (S, F_1) is a sequence (S_i, F_i') where $S_1 = S$, $F_1' = F_1$, $S_i = P_{i-1}'(S_{i-1})$, $F_i' = N_{i-1}'(S_{i-1})$.

Now it is possible to say what is meant by correctness. Let $F_1 \in P$, $\mathcal{S}_{IN}$ and $\mathcal{S}_{OUT}$ be conditions on $\mathcal{S}$, then P is correct with respect to $(F_1, \mathcal{S}_{IN}, \mathcal{S}_{OUT})$ if whenever $S \in \mathcal{S}_{IN}$ the computation sequence of S, F_1 terminates (say at F_n')

and $S_{n+1} \in \mathcal{S}_{OUT}$. P is partially correct with respect to $(F_1, \mathcal{S}_{IN}, \mathcal{S}_{OUT})$ if whenever the computation sequence of (S, F_1) terminates then $S_{n+1} \in \mathcal{S}_{OUT}$.

First we describe how to prove that a program is partially correct. Let P be a program on $\mathcal{T} = \mathcal{S} \times P$. A <u>precondition</u> <u>structure</u> for P is a collection of conditions $\mathcal{S}_i \subset \mathcal{S}$ with one $\mathcal{S}_i$ associated with each F_i in P. The precondition structure induces a directed graph on P having as nodes the F_i and a link from F_i to F_j if and only if there is an $S \in \mathcal{S}_i$ such that $N_i(S) = F_j$. A <u>substructure</u> is any subset U of the precondition structure such that $\mathcal{S}_x \in U$. A <u>control</u> <u>path</u> of U is a sequence of $F_{i_j} \in P$, $1 <= j <= n$, (distinct except possibly $i_1 = i_n$) such that $(\mathcal{S}_{i_1}, \mathcal{S}_{i_n}) \in U$ but no other $\mathcal{S}_{i_j}$ are in U, and $N_{i_{j-1}}(S) = F_{i_j}$ for some $S \in \mathcal{S}'_{i_{j-1}}$. The $\mathcal{S}'_{i_j}$ are defined inductively with $\mathcal{S}'_{i_1} = \mathcal{S}_{i_1}$ and $\mathcal{S}'_{i_j} = \langle P_{i_{j-1}}(S);\ S \in \mathcal{S}'_{i_{j-1}} \text{ and } N_{i_{j-1}}(S) = F_{i_j} \rangle$. A control path is <u>consistent</u> if $\mathcal{S}'_{i_n} \subset \mathcal{S}_{i_n}$ and $P_{i_j}(S)$ is defined for all $S \in \mathcal{S}'_{i_j}$ such that $N_{i_j}(S)$ is defined. A substructure of P is consistent if all of its control paths are consistent.

Theorem: Let P be a program on $\mathcal{J} = \mathcal{S} \times P$ with $F_1 \in P$, $\mathcal{S}_{IN}$, $\mathcal{S}_{OUT}$ conditions on $\mathcal{S}$. Then P is partially correct with respect to $(F_1, \mathcal{S}_{IN}, \mathcal{S}_{OUT})$ if and only if P has a consistent substructure U with $\mathcal{S}_1 \in U$, $\mathcal{S}_{IN} \subset \mathcal{S}_1$ and $\mathcal{S}_x \subset \mathcal{S}_{OUT}$.

Theorem: Let P be a program on $\mathcal{S} \times P$. A substructure U for P will have a finite number of control paths if the graph on P obtained by removing from the induced graph all nodes F_i and associated links for $\mathcal{S}_i \in U$ is acyclic.

Thus to prove that a program is partially correct it is sufficient to find a suitable substructure and check each control path for consistency.

To show that a program terminates we introduce the notion of expressions. Let P be a program on $\mathcal{J} = \mathcal{S} \times P$ with precondition structure $\mathcal{S}_i$. A subset $Q \subset P$ is a <u>set of sufficiency</u> if the graph on P obtained by removing from the induced graph all nodes F_i and associated links for $F_i \in Q$ is acyclic. $UQ = \langle \mathcal{S}_i (F_i \in Q) \cup \mathcal{S}_x \rangle$ is a substructure for P with a finite number of control paths. An <u>expression</u> is a function e from $\mathcal{S}$ to $\mathbb{Z}$ (the integers). <u>A controled set of expressions is</u> a set $\langle e_i, i = 1,n,\ e_i$ an expression$\rangle$ such that for a fixed set of sufficiency Q there is at least one e_j that is increasing along any control path

$F_{i_1} \ldots F_{i_n} \neq F_x$ in UQ with e_j bounded at F_{i_n} and $\langle e_k, k < j \rangle$ are nondecreasing along the control path.

Theorem: A program P on $\mathcal{S} \times P$ terminates if P has a controlled set of expressions.

When subroutine calls occur in a program for purposes of proof the call is replaced by a function duplicating the effect of the subroutine on $\mathcal{S}$. Then each subroutine and main program is treated independently. Maurer discusses this method in (4).

The program COSET uses the Todd-Coxeter algorithm to enumerate the cosets of a subgroup $\mathcal{H}$ in a free group $\mathcal{F}$. The result is the index of $\mathcal{H}$ in $\mathcal{F}$ if suitably small, otherwise the program terminates with an overflow signal. The groups $\mathcal{H}$ and $\mathcal{F}$ are defined below.

$\mathcal{F}$ is the free group on NJ generators: $\mathcal{F} = \langle 1,2,\ldots \mathrm{NJ} \rangle$. A word in $\mathcal{F}$ is a product, $J_1 \, J_2 \, \ldots \, J_K$ of generators and their inverses. Let $\mathcal{S} = \langle U(K) \mid 1 \Leftarrow K \Leftarrow NU \rangle$ and $\mathcal{R} = \langle R(K) \mid 1 \Leftarrow K \Leftarrow NR \rangle$ be subgroups of $\mathcal{F}$ generated by the words U(K) and R(K) respectively, then $\mathcal{H}$ is the subgroup of $\mathcal{F}$ generated by $\mathcal{S}$ and the normalizer of $\mathcal{R}$ in $\mathcal{F}$.

The version of COSET for which the correctness proof was carried out was written by H. F. Trotter. For further details on the theory and other programs see Trotter [1] and Leech [2].

2. P-Sets.

In the following sections lower case letters will be used for numbers, upper case for variables (or their values as determined by context).

Sets will be enclosed by $\langle\ \rangle$. For $x \in M$, M a set of variables, the set of allowed values of x is assumed to be that allowed by the type of x (e.g., integer, logical, etc.). For x an input variable the allowed values are those determined by the corresponding format statement.

The set M on which Coset is defined is the disjoint union of the following sets:

M(INPUT) = ⟨ INPUT(i,j) = ith field of jth input record
FORMAT(I1, I3, 34I2) ⟩

M(COMMON) = ⟨ KL(n), KV(n), KF(n,k), KR(n,k),
1 <= n <= NMAX, 1 <= k <= NJX ⟩
∪ ⟨ LS(n), LF(n), LR(k,n), 1 <= n <= NRX,
1 <= k <= NSX ⟩
∪ ⟨ NMAX, N, NJ, NR, NVAC, ID, KN ⟩

M(MAIN) = ⟨ NRX, NSX, NJX, F, SW, I, J, NCD, ICD, MULT, LWD, NCOS ⟩

M(CONSOL) = ⟨ M, J, I, KT, K ⟩

M(APPLY) = ⟨ IT, LA, L, LEN, LNS, I, IB, IX, M, MX, MY, J, JT, IA ⟩

M(NOTE) = ⟨ IX, IY, IT, IA, IB ⟩

Variables with the same name appearing in different subsets are distinct as they in fact occur in different subprograms. It is convenient to define subarrays of $A' \subset A$ of those occurring in M(COMMON) since in general program statements and preconditions refer only to elements of these subarrays.

$KV' = \langle KV(n),\ 1 <= n <= N \rangle$

$KL' = \langle KL(n),\ 1 <= n <= N \rangle$

$KF' = \langle KF(n,k),\ 1 <= n <= N,\ 1 <= k <= NJX \rangle$

$KR' = \langle KR(n,k),\ 1 <= n <= N,\ 1 <= k <= NJX \rangle$

$LS' = \langle LS(n),\ 1 <= n <= NR \rangle$

$LF' = \langle LF(n),\ 1 <= n <= NR \rangle$

$LR' = \langle LR(k,n),\ 1 <= k <= LS(n),\ 1 <= n <= NR \rangle$

3. Special Conditions.

Conditions will have the form:

COND identifier index = ⟨* ... *⟩

The index is optional, occurring when more than one condition has the same identifier. Reference to an indexed identifier without specific index will be reference to the intersection over all values of the index of conditions with that identifier. If A and B are identifiers then the intersection of A and B will be written ⟨*A, B*⟩.

3.1. INPUT

Let iz, ig, ir, be non-negative integers such that the following inequalities are satisfied:

(i) 0 <= iz <= ig <= ir and iz < ir

(ii) ir - ig < NRX

COND INPUT A = ⟨* INPUT(1,icd) = 0, 1 <= icd <= iz;
INPUT(1,icd) ≠ 0, 2, iz < icd <=ig;
INPUT(1,icd) = 2, ig < icd <= ir;
INPUT(1,icd) = 0, icd = ir + 1 *⟩

COND INPUT B = ⟨* (|INPUT(J,icd)| <= NJX, 3 <= j <= NSX+2,
INPUT(2,icd) > 0, INPUT(3,icd) ≠ 0),
iz < icd <= ir *⟩

$$\Delta(\text{INPUT}) = \text{M(INPUT)}$$

3.2. GC(j)

Let U(ℓ), R(ℓ) belong to the sets defined in section 1 and let IY(j) be the identity element of $\mathcal{F}$ expressed as the product $j\bar{j}\bar{j}j$ where j is a generator of $\mathcal{F}$ and $\bar{j}$ its inverse. W(ℓ) represents the word in $\mathcal{F}$ given by $W(\ell)=\Pi_{i=1}^{LF(\ell)} LR(ii,\ell)$ where ii ≡ i mod LS(ℓ) and 1 <= ii <= LS(ℓ) and Π is the group operation written multiplicatively

COND $\quad GC(0) = \langle * \; 1 <= N <= NMAX, \; 1 <= NR <= NRX, \; 1 <= NJ <= NJX, \; 0 <= KN <= N \; * \rangle$

COND $\quad GCL(j) = \langle * \; 1 <= LS(i) <= NSX, \; 1 <= LF(i) <= 1000 \; NSX, \; LS(i) | LF(i), \; (1 <= |LR(k,i)| <= NJX, \; 1 <= k <= LS(i)), \; 1 <= i <= j \; * \rangle \qquad \text{for all } j > 0$

COND $\quad GCW = \langle * \; W(1) = U(icd-ig) \; * \rangle, iz < icd <= ig;$

$= \langle * \; W(\ell) = R(\ell), \; 1 <= \ell <= icd-ig \; * \rangle, \; ig < icd <= ir;$

$= \langle * \; W(\ell) = R(\ell), \; 1 <= \ell <= ir-ig, \; W(ir-ig+1) = \Pi_{k=1}^{NJ} \; IY(k) \; * \rangle, icd = ir + 1 \; .$

COND $\quad GC(j) = \langle * \; GC(o), \; GCL(j), \; GCW \; * \rangle \qquad j > 0$

$\Delta(GC(j)) = \langle N, \; NR, \; NJ, \; KN, \; ICD, \; ((LR(k,i), \; 1 <= k <= LS(i)), \; LS(i), \; LF(i), \; 1 <= i <= j) \rangle$

3.3. K

COND $\quad K0 = \langle * \; 0 <= ID <= N; \; 0 <= KL(i) <= N, \; 1 <= i <= N \; * \rangle$

Let GKL be the directed graph defined by KL' as follows. GKL has N+1 vertices labelled 0,1,...,N. GKL has an arc from i to j = KL(i), 1 <= i <= N, an arc from 0 to ID, and no other arcs. Define MCOS, C(k), P(k), I(k) as follows: MCOS + 1 is the number of components of GKL, $P(k)$ is the number of vertices in the kth component, $C(k) = \langle i \mid \text{vertex } i \text{ belongs to component } k \rangle$, $I(k) = \min C(k)$, and assume that the components are numbered such that $k < k' \Rightarrow I(k) < I(k')$, $0 \le k <$ MCOS

COND K1 = ⟨* The components of GKL are cyclic and the vertices 0, 1 belong to different components *⟩

Note that K1 implies the following:

$1 \le \text{MCOS} \le N$, $I(0) = 0$, $I(1) = 1$, $0 \le I(k) \le N$, $1 \le P(k) \le N$.

If KL(0) is defined to be ID then KL can be viewed as a function on the set $\langle 0,1,\ldots,N \rangle$ and composition of KL with itself p times is also a function on $\langle 0,1,\ldots N \rangle$ with $KL^{p}(i) = KL(KL^{p-1}(i))$ and $KL^{o}(i) = i$.

COND K2 = ⟨* $KV(i) = I(j)$ for $i \in C(j)$, $1 \le j \le$ MOCS; $-i < KV(i) < 0$ and $(-KV(i) \in C(0)$ or $-KV(i) = I(j)$ for some j, $1 \le j \le$ MCOS$)$, $i \in \langle C(0) - 0 \rangle$ *⟩

K0-2 imply that the following sequence is well defined and terminates for each $K \in \langle C(0) - 0 \rangle$: $K_0 = K$, $K_i = -KV(K_{i-1})$ if $K_{i-1} \in C(0)$, K_i terminates the sequence if $K_i \notin C(0)$. Thus the sequence will satisfy $K = K_0 > K_1 \cdots > K_n = I(j)$ for some j, $1 \le j \le$ MCOS.

Now define the function EC on $\langle 0,1,\ldots,N \rangle$

$$EC(0) = 0$$

$$EC(k) = KU(K) \qquad k \notin C(0)$$

$$EC(k) = K_n \qquad k \in \langle C(0) - 0 \rangle$$

EC partitions the set $\langle 0,\ldots,N \rangle$ into equivalence classes, one class for each j, $0 <= j <= MCOS$. For $k \in \langle C(0) - 0 \rangle$ the sequence k_i above will be called the EC chain with head k, tail EC(k); for $k \notin C(0)$ the pair (K,KV(k)) will be called the EC Chain with head k, tail EC(k).

Links and Chains

For k, ℓ, j such that $1 <= k, \ell <= N$, $1 <= |j| <= NJ$ we say that $\langle KF' + KR' \rangle$ contains a link from k to ℓ covering j, written $(k,j,\ell) \in \langle KF' + KR' \rangle$, iff (a) $j > 0$ and there is some $k', \ell' \ni EC(k) = EC(k'), EC(\ell) = EC(\ell')$ and $KF(k',j) = \ell'$ or (b) $j < 0$, k', ℓ' as in (a) and $KR(k',-j) = \ell'$.

Similarly, for $1 <= k_i <= N$, $1 <= |j_i| <= NJ$ we say that $\langle KF' + KR' \rangle$ contain a chain from k_0 to k_n covering $W = j_1 j_2 \cdots j_n$, written $(k_0,w,k_n) \in \langle KF' + KR' \rangle$, iff $(k_{i-1},j_i k_i) \in \langle KF' + KR' \rangle$, $1 <= i <= n$. $\langle W(i,j) \rangle$ denotes the set of words in $\mathfrak{J}$ covered by a chain from i to j, W(i,j) an element of this set.

Let the integer 0 represent the identity in $\mathfrak{F}$ and define $KF(k,0) = KR(k,0) = k,\ 1 \leq k \leq N$

Then the trivial link $(k,0,k)$ is contained in $KF' + KR'$ for $1 \leq k \leq N$. The statement $(k,j,0) \in KF'+KR'$ means that (a) $j > 0$ and $KF(k',j) = 0$ for all k' such that $EC(k') = EC(k)$ or (b) $j < 0$ and $KR(k',-j) = 0$ for all k such that $EC(k') = EC(k)$.

COND $\quad K\ 3A = \langle *\ KF(i,j) = KR(i,j) = 0$ if $N < i \leq NMAX$
or $NJ < j \leq NJX$;
$0 \leq KF(i,j),\ KR(i,j) \leq N$ if $1 \leq i \leq N$
and $1 \leq j \leq NJ. * \rangle$

$K\ 3B = \langle *\ EC(KF(I(\ell),j)) = I(k) \Rightarrow EC(KR(k',j)) = I(\ell)$
and $EC(KR(I(\ell),j)) = I(k) \Rightarrow EC(KF(k',j)) = I(\ell)$
for $1 \leq k,\ell \leq MCOS$ and some k' such that
row k' of KF, KR is labeled $I(k)\ * \rangle$

$K\ 3C = \langle *\ (i,j,k) \in (KF' + KR')$ for $j \leq i,k \leq N$
$j \leq j \leq NJ,\ EC(i) = I(\ell) \Rightarrow$
$EC(KF(i',j)) = EC(k)$ for some i' such that
row i' is labeled $I(\ell)\ * \rangle$

$K\ 3D = \langle * \langle W(1,I(k)) \rangle \neq \emptyset\ ,\ 1 \leq k \leq MCOS\ * \rangle$

A row i' is said to be labeled I(i) if i' = I(i) or if i' $\in \langle$ C(0) - 0 $\rangle$ and EC(i') = I(i).

COND K 4 = $\langle$* 0 <= NVAC, NVAC + MCOS + P(0) = N+1 *$\rangle$

3.4. Chain

The word W(ℓ) as defined in section 3.2 can be expressed as a product as follows.

Define

$$W(j,\ell) = \prod_{i=1}^{j-1} LR(ii,\ell), \quad 2 <= j <= LF(\ell)+1$$

$$W(j,\ell) = 0, \ j = 0 \text{ or } 1$$

$$W(-j,\ell) = \prod_{i=LF(\ell)-j+1}^{LF(\ell)} LR(ii,\ell) \qquad 1 <= j <= LF(\ell)$$

where as before ii $\equiv$i mod (LS(ℓ)) and i <= ii <= LS(ℓ).

Note that W(LF(ℓ+1),ℓ) = W(-LF(ℓ),ℓ) = W(j,ℓ) W(-(LF(ℓ) - j + 1), ℓ) = W(ℓ).

COND CHAIN (ix, iy, $\pm$ j, ℓ) =
$\langle$* (I(x), W($\pm$j,ℓ), I(y)) $\in \langle$ KF' + KR' $\rangle$ for
I(x) = EC(ix), I(y) = E(iy) *$\rangle$

4. Array AK

4.1. Definition and interpretation

Let $\mathcal{A}$ be the set of all arrays having non-negative integer entries. AK is the function from $\mathcal{S}' = \langle$* K, GC(NR)*$\rangle$ into $\mathcal{A}$ defined as follows:

For S $\in \mathcal{S}'$, AK(S) is an array with NJ+1 columns indexed by the integers 0, 1,... , NJ, and rows determined by KL', KV', KF' KR'. For each k, 1 <= k <= MCOS and each row k' labeled I(k), AK(S) has a row with I(k) in column 0, and EC(KF(k',j)) in column j, 1 <= j <= NJ. For each pair (k,j), 1 <= k <= MCOS, 1 <= j <= NJ, such that KR(I(k),j) = 0, AK(S) has a 'single entry' row with I(k) in column j and zeros elsewhere. Δ(AK) = $\langle$ KV', KL', KF', KR', N, NJ, ID$\rangle$.

We say that i $\in$ AK(S) iff i occurs as an entry in some row and column of the array. (Zeros are considered to be blanks.) In particular, i $\in$ AK(S) iff i = I(k) for some k, 1 <= k <= MCOS. AK(S) is complete if for every row x $\in$ AK(S) there is a nonzero entry in column j, 0 <= j <= NJ. In the following sections AK or AK(S) will be used to denote the array described above.

A link in AK is a triple (i,j,k), i, k $\in$ AK, |j| <= NJ such that either j >= 0 and AK contains a row with i in column 0, k in column j or j < 0 and (k,-j,i) $\in$ AK. Chains in AK are defined in the obvious way. $\langle$* K3 *$\rangle$ insures that (i,j,k) $\in$ AK iff (k,j,k) $\in$ $\langle$KF' + KR'$\rangle$.

Proposition. If $S \in \mathscr{S}' = \langle * \text{ K, GC(NR) } * \rangle$ then AK(S) satisfies the following properties:

(i) Every row of AK with more than one (non-zero) entry has an entry in column 0.

(ii) $1 \in AK$

(iii) $i \in AK \Rightarrow i$ occurs at least once in each column of AK

(iv) For each i in AK, AK contains a chain from 1 to i covering some word in $\mathfrak{F}$.

Proof:

(i) clear from definition of array

(ii) $I(1) = 1$ since $S \in \langle * \text{ K1 } * \rangle$

(iii) i occurs in column 0 by definition, for each $j > 0$, either $KR(i,j) = 0$ or $EC(KR(i,j)) = k$, $k \in AK$. If the former is true then i occurs in column j of a single entry row, if the latter then i occurs in column j of the row defined by $\langle KF(k,j) \rangle$.

(iv) clear since $S \in \langle * \text{ K3 } * \rangle$

Proposition. If $S \in \mathscr{S}'' = \langle * \text{ K, ID} = 0, \text{ GC(NR) } * \rangle$ then AK(S) satisfies the additional properties:

(v) $i \in AK \Rightarrow i$ occurs at most once in any column of AK.

(vi) No proper subset of rows of AK satisfy (i-iii).

Proof.

(v) $ID = 0 \Rightarrow \langle C(0) - 0 \rangle = \emptyset$ thus i occurs only once in column 0.

Suppose i occurs in rows x and y column j with k_x, k_y in column 0 of rows x and y. From the definition not both k_x and k_y are 0, suppose $k_x \neq 0$. If $k_y = 0$, then $KR(i,j) = 0$, but since $S \in \langle * K3, ID = 0 * \rangle$

$$EC(KF(k_x,j)) = i \Rightarrow KR(i,j) = k_x \neq 0$$

Thus $k_y \neq 0$. If $k_y \neq k_x$ then $EC(KR(i,j)) = k_y$ contradicting $EC(KR(i,j)) = k_x$.

(vi) Let $AK' \subset AK$ be a subset of rows of AK such that AK' satisfies (i-iii) then we have

(a) $i \in AK - AK' \Rightarrow i \notin AK'$ since AK satisfies (v) and AK' satisfies (iii).

(b) $1 \notin AK - AK'$.

Thus it is sufficient to show $AK - AK' \neq \emptyset \Rightarrow 1 \in AK - AK'$. For each $i \in AK$ there is a chain from 1 to i given by $\langle (i_{k-1}, j_k, i_k) \mid 1 \leq k \leq mi \rangle$ with $i_0 = 1$, $i_{mi} = i$. If $i \in AK - AK'$ then $i_{mi-1} \in AK - AK'$ since i_{mi-1} is contained in some row also containing i. Thus by induction on mi it is clear that $i \in AK - AK' \Rightarrow 1 \in AK - AK'$.

For each row x of AK define $\mathscr{L}(x) = \langle (i,j,k),(k,-j,i) \mid$ row x column 0 contains i and (j,k) such that row x column j contains $k \neq 0\rangle$ and $\mathscr{L}(AK) = U_x\mathscr{L}(x)$, all $x \in AK$. If AK contains two rows x, xa such that $\mathscr{L}(x) \subset \mathscr{L}(xa)$ then deleting row $\mathscr{L}(x)$ leaves $\mathscr{L}(AK)$ unchanged. Let $H(AK) = \langle W(1,1)\rangle$ be the set of words in $\mathfrak{F}$ covered by a chain from 1 to 1 in AK. H(AK) is a subgroup of $\mathfrak{F}$. Clearly any function $P : S \longrightarrow S'$ that leaves $\mathscr{L}(AK)$ fixed also leaves H(AK) fixed.

If $S \in \mathscr{S}'' = \langle *$ K, ID = 0, GC(NR) $*\rangle$ then the sets $\langle W(1,i)\rangle$ belong to different cosets of H(AK) in $\mathfrak{F}$ for different i. If AK is complete AK affords a transitive permutation representation of $\mathfrak{F}$ on the cosets of H(AK), and the set $\langle i \mid i \in AK\rangle$ is in 1-1 correspondence with cosets of H(AK) in $\mathfrak{F}$. For $H0 \in H(AK)$ and k such that 0 <= k <= N define H(K,AK) = $\langle$HO+$\langle$ W(1,kk) W(ℓ) W(kk,1); 1 <= kk <= k, 1 <= ℓ <= NR$\rangle\rangle$ For H0 = $\mathscr{H}$ of introduction and W(ℓ) = R(ℓ), H(N,AK) is contained in H(AK) implies H(AK) = $\mathscr{K}$(See Trotter [1].)

4.2. Lemmas

Lemma 1: Let P be a function on $\mathscr{S}' = \langle *$ K, GC(NR) $*\rangle$ (P(S) = S' $\in \mathscr{S}'$ for S $\in \mathscr{S}'$) such that S'(NJ) >= S(NJ) and S'(x) = S(x), $x \in \langle \Delta(S')$-NJ$\rangle$. Then H(AK') = H(AK).

Proof: S ϵ K3 => KF(i,j) = KR(i,j) = 0, 1 <= i <= N, N < NJ therefore $\mathcal{L}$(AK') = $\mathcal{L}$(AK) and H(AK') = H(AK).

Lemma 2: Let $\mathcal{S}$'' = $\langle$* GC(NR), K, KF(i,j) = 0 for some (i,j) such that 1 <= i <=N, 1 <= j <= NJ *$\rangle$

Let P be a function on $\mathcal{S}$'' (P(S) = S' for S ϵ $\mathcal{S}$'') such that N' = S'(N) = S(N)+1 <= NMAX, S'(K (N')) = S'(KL(N')) = N', S'(KF(i,j)) = N', S'(KR(i,j)) = 1 and S'(x) = S(x) elsewhere. Then S' ϵ $\langle$* GC(NR), K *$\rangle$, and H(AK') = H(AK).

Proof: S' ϵ $\langle$* GC(NR), K *$\rangle$ clear since P effectively adds one new single vertex component GKL and one new pair of links (i,j,N'), (N',-j,i) to KF' + KR'. $\mathcal{L}$(AK') = $\mathcal{L}$(AK) + (i,j,N') + (N',-j,i). Any chain (1,w,1) is AK' not in AK must contain one of the two new links. However the only link that can be adjacent on the N' side is the other new link, thus they must always occur pairwise and cover the identity in $\mathcal{G}$, giving no new words in H(AK').

Lemma 2': Lemma 2 with KF, KR interchanged.

Lemma 3: Let $S'' = \langle *\ K,\ GC(NR),\ KF(ia,j) = k,\ KF(i,j) = 0$ for some i, ia, j, k such that $EC(i) = EC(ia)$, $1 <= j <= NJ$, $1 <= i, ia, k <= N\ * \rangle$. Then the function P on $\mathscr{S}''$ ($P(S) = S'$, $S \in \mathscr{S}''$, $S' \in \mathscr{S}'$ as defined above), $S'(KF(i,j)) = k$, $S'(x) = S(x)$, $x \in \langle \Delta(\mathscr{S}') - KF(i,j) \rangle$ leaves $H(AK)$ unchanged.

Lemma 3': Lemma 3 with KF replaced by KR. Clear since $L(AK') = L(AK)$.

Lemma 4: Let $ia = I(a) < ib = I(b)$ for some a, b, $1 <= a < b <= MCOS$ and let P be a function on $\mathscr{S}' = \langle *\ GC(NR), K\ * \rangle$ such that $S'(KF,KR) = S(KF,KR)$, $S'(KV(i)) = S(KV(i))$, $i \notin C(b)$, $S'(KV(ib)) = -ia$, $S'(KV(i)) = ia$, $i \in \langle C(b) - ib \rangle$, $S'(KL)$ such that $C'(i)=C(i)$, $i \neq 0,a,b$, $C'(0) = \langle C(0) \cup ib \rangle$, $C'(a) = C(a) \cup \langle C(b)-ib \rangle$; $S''(x) = S(x)$ elsewhere in $\Delta(\mathscr{S}')$. Then (i) $S' \in \mathscr{S}'$; (ii) AK' is equal to AK with ib replaced by ia throughout and duplicate single entry rows deleted; and (iii) $H(AK') = H(AK) \cup \langle W(1,ia)W(ib,1) \rangle$.

Proof.(i): $S' \in GC(NR)$, K 3A since $\rho(P) \cap \Delta(GC(NR), K3A) = \varphi$ $S' \in K0, K1, K2$ since KV is changed to be consistent with changes in KL. $S' \in K4$ because MCOS is decreased by 1 and $P(0)$ is increased by 1 while NVAC and N are unchanged.

From definition of EC and row labels it is clear that the final set of rows labeled I(a) is the union of the initial sets of rows labeled I(a) and I(b). Thus $S' \in K3C$.
Let KF#, KR# represent initial values and KF, KR represent final values. A chain covering W in KF#' + KR#' is also a chain covering W in KF' + KR'--possibly with different head and tail if the original head or tail belonged to the equivalence class I(b), thus $S' \in K3D$. For K3B it is sufficient to show the implications are true for k = a since the others are not affected. $EC(KF(I(\ell),j) = I(a) \Rightarrow EC(KF\#(I(\ell),j) = I(a)$ or $I(b) \Rightarrow EC(KR\#(k',j)) = I(\ell)$ for some k' in the set of rows labeled initially I(a) or I(b), respectively $\Rightarrow EC(KR(k',j)) = I(\ell)$ for some k' in the set of rows labeled I(a). Similarly for $EC(KR(I(\ell),j)) = I(a) \Rightarrow EC(KR(k',j)) = I(\ell)$. Thus $S' \in K3B$ and part (i) is proved.

Proof (ii): Note that each k such that EC(K) = ib in S EC(k) = ia in S' since $S \in \langle *K* \rangle$ and EC(K) unchanged if not ib in S. Result then follows from definition of array.
(iii) see Trotter [1].

Corollary 1: If AK contains a chain (ia, o, ib) and P as in Lemma 4 then H(AK') = H(AK).

Corollary 2: Let $\mathscr{S}'' = \langle *\ K,\ GC(NR),\ Chain(it,ix,i,\ell),$ $Chain(ig,it,-(LF(\ell) - i + 1),\ell), it, ix, iy \in AK\ * \rangle$ and set $ib = \max(ix,iy)$, $ia = \min(ix,iy)$. Then for P as above ($S \in \mathscr{S}''$, $S' = P(S)$), $H(AK') = H(AK)$ $\cup \langle w(1,it),\ W(\ell)\ w(it,1) \rangle$.

Recall that $W(\ell) = W(i)\ W(-(LF(\ell)-i+1))$.

Corollary 3: Let $\mathscr{S}'' = \langle *\ K,\ GC(NR),\ Chain(it,ix,i,\ell),$ $Chain(iy,it,-(LF(\ell)-i),\ell),\ (iy,-j,0) \in \langle\ KF' + KR' \rangle$ where $it,\ ix,\ iy \in AK$, $j = LR(ii,\ell)$, $ii \equiv i\ MOD(LS(\ell))$, $1 \Leftarrow ii \Leftarrow LS(\ell)\ * \rangle$. Consider two cases (i) $(ix,j,0) \in \langle KF' + KR' \rangle$ or (ii) $(ix,j,iz) \in \langle\ KF' + KR' \rangle$, $iz \in AK$.

(i) Let Q be the function from $\mathscr{S}''$ to $\mathscr{S}'$ such that $Q(S) = S'$ and $S'(KF(ix,j)) = iy$, $S'(KR(iy,j)) = ix$ if $j > 0$ (or $S'(KR(ix,-j)) = iy$, $S(KF(iy,-j)) = ix$ if $j < 0$), $S'(x) = S(x)$, $x \in \langle\ \Delta(\mathscr{S}') - KF(ix,j),\ KR(iy,j) \rangle$ or $(x \in \langle \Delta(\mathscr{S}') - KF(iy,-j),\ KR(ix,-j) \rangle)$.
Then $H(AK') = H(AK) \cup \langle\ w(1,it)\ W(\ell)\ w(it,1) \rangle$.

Proof: Q is equivalent to application of P of Lemma 2 followed by P of Lemma 4 with $ib=N+1$, $ia = iy$.

(ii) Let Q be the function such that S'(KR(iy,j)) = ix $j > 0$ or S'(KF(ix,-j)) = iy, $j < 0$, S'(x) = S(x) elsewhere, followed by P of Lemma 4 with ib = max(iz,iy), ia = min(iz,iy). The A(AK') = H(AK) $\cup \langle$ w(1,it) W(ℓ) w(it,1) $\rangle$.

Proof: Q is equivalent to P of Lemma 4 followd by P of Lemma 3.

Lemma 5 (CONSOL): Let $\mathscr{S}'$ = $\langle$* K, ID = 0, GC(NR) *$\rangle$ and suppose H(I(g),AK) ϵ H(AK) for some I(g), 0 <= I(g) <= KN Let Im be the function from $\langle$ik = I(k); 0 <= k <= MCOS$\rangle$ onto $\langle$ k; 0 <= k <= MCOS$\rangle$ such that 0 <= IM(ik) < IM(iℓ) <= MCOS for 0 <= k < ℓ <= MCOS. Im is 1-1, onto, order preserving. In fact IM(I(k)) = k = |$\langle$ kk; 1 <= kk <= k, KV(KK) = KK$\rangle$|. Denote by IMI the inverse of I .

For S ϵ $\mathscr{S}'$ let KV#(i) = S(KV(i)) and similarly for other variables. Let P be the following function on $\mathscr{S}'$. (S' = P(S), KV(i) = S'(KV(i)) and similarly for other variables.)

KF(k,j) = IM(KU#(KF#(IMI(k),j))), 1 <= k <= MCOS
= 0, MCOS < k <= N; 1 <= j <= NJ.

$$KR(k,j) = IM(KV\#(KR\#(IMI(k),j))),\ 1 \Leftarrow k \Leftarrow MCOS$$
$$= 0,\ MCOS < k \Leftarrow N;\ 1 \Leftarrow j \Leftarrow NJ.$$

$KV(k) = KL(k) = k; \qquad 1 \Leftarrow k \Leftarrow MCOS$

$N = MCOS$

$NVAC = 0$

$KN = IM(KV\#(KN\#))$

and $S'(x) = S(x),\ x \in \langle \Delta GC(NR)-N,\ NVAC,\ KN \rangle$. Then $S' \in \langle * \ K,\ ID = 0,\ GC(NR)\ * \rangle$, $H(AK) = H(AK\#)$, $H(g,AK) \in H(AK)$.

Proof: $S' \in G(NR)$ is clear since $\rho(P) \cap \Delta(GC(NR)) = (N,NVAC,KN)$ and conditions of GC are satisfied by the final values of these variables. $S' \in K0, K4$ is obvious. GKL is the graph with MCOS+1 components, each with a single vertex. $KV(k) = KL(k) = K$ thus $S' \in K1,K2$. From the definitions of P, IM, and $S \in K$, $ID = 0$ the following is true:

$$KF(k,u) = \ell \iff KV\#(KF\#(IMI(k),j)) = IMI(\ell) \iff KV\#(KR\#(IMI(\ell),j))$$
$$= IMI(k) \iff KR(\ell,j) = k$$

K3C is trivial thus $S' \in$ K3A,B,C.

AK contains a chain $\Pi_{k=0}^{p}$ $(I(a_k), j_k, I(a_{k+1}))$ covering $w = \Pi_{k=0}^{p} j_k$ $(1 <= a_k <= MCOS, 1 <= |j_k| <= NJ; 0 <= k <= p)$, iff AK contains a chain $\Pi_{k=0}^{p}$ (a_k, j_k, a_{k+1}) covering $w = \Pi_{k=0}^{p} j_k$. Since I(1) = 1 chains from 1 to 1 are preserved thus S' ϵ K3D, H(AK) = H(AK) and H(g,AK) ϵ H(AK).

5. Verification of Coset

Verification of the program COSET and of the subroutines called by COSET is carried out in two steps (For details of algorithm verification see Maurer [3].) First proof that the given precondition substructure is pathwise consistent, thus proving partial correctness. Proof of consistency will consist of listing each control path with its initial and final preconditions and brief indication of reasoning necessary to prove consistency. The effective range of the path is contained in the set defined by RHO (NAME) Path $= \langle \ \rangle$. PC(NAME) PATH$\#$ $= \langle * \ \ * \rangle$ gives the conditions satisfied by the variables at branch points along the path. Reasoning common to many of the control paths is discussed below. The proofs are brief since the preconditions are set up to make consistency obvious in mose cases. Lemmas have been proved to simplify presentation and are referred to where appropriate. Full listing of each (sub) program with preconditions can be found in Appendix A.

The most common reason for consistency of a final precondition or subcondition, $\mathcal{A}''_x$, with an initial condition, $\mathcal{A}'$, is indicated in the proofs as NULL INTERSECTION. That is, the effective range of the path $\rho(P)$ has the property that $\rho(P) \cap \Delta(\mathcal{A}''_x) = \emptyset$ and $\mathcal{A}''_x \supset \mathcal{A}'$. A condition is labeled vacuous to indicate the set $\mathcal{A}''_x$ is the compliment of $\emptyset$. This occurs with the domain of the condition is empty. For example: $\langle *\ KV(I) = IM(I),\ 1 <= I < K\ * \rangle$ where K is known to be less than or equal to 1. A condition said to be consistent by PATH CONDITION assumes also properties of the initial conditions and possibly assignments. For example suppose $\mathcal{A}'_x = \langle *\ 1 <= I <= N\ * \rangle$ $\mathcal{A}''_y = \langle *\ 1 <= I <= N\ * \rangle$, PC $= \langle *\ I <= N\ * \rangle$ and the assignment I = I+1 is only statement along the path affecting the value of I. Then PATH CONDITION will be written for proof of consistency of $\mathcal{A}''_y$.

A precondition is labeled global if it is a valid precondition for all statements with preconditions that can be reached via a computation sequence beginning at the statement before which the condition is inserted. Proof of consistency for such preconditions will be included in the introduction to the correctness proof for the (sub) program.

The second step is the proof of termination which in each case is proof that a given set of expressions is a controlled set of expressions with respect to a given set of sufficiency, Q.

The proof of termination assumes that the preconditions are consistent. The variable λ occurs in the set of expressions for each (sub) program. The value of λ is determined as follows. The graph of the (sub) program is divided into disjoint subgraphs Di such that the induced graph on $\langle$Di$\rangle$ is acyclic. The value of λ at any statement in the (sub) program is i where the statement is in subgraph Di. The graphs are shown in Figs. 1-4.

5.1. Proof of correctness for subroutine NOTE

This section contains proof that if S ϵ COND-IN(NOTE) and P = NOTE then P(S) ϵ COND-OUT(NOTE). Let X = EC(IX), Y = EC(IY) upon entering.

$$\text{COND-IN(NOTE)} = \langle *\ \text{GC(NR), K, } 0 \Leftarrow \text{IX, IY} \Leftarrow \text{N} * \rangle$$

$$\text{COND-OUT(NOTE)} = \langle *\ \text{K, GC(NR), one of the set } \langle \text{CI, CII, CIII, CIV} \rangle * \rangle$$

$$\text{CI} = \langle *\ \text{IX} = 0 * \rangle$$

$$\text{CII} = \langle *\ \text{IX} > 0,\ \ \text{Y} = \text{IX, X} = 0 * \rangle$$

$$\text{CIII} = \langle *\ \text{X} = \text{Y} * \rangle$$

$$\text{CIV} = \langle *\ \text{IX, iY} > 0\text{, IA} = \text{MIN(X,Y), IB} = \text{MAX(X,Y),}\ \text{EC(IB)} = \text{EC(IX)} = \text{EC(IY)} = \text{IA} * \rangle$$

GC(NR) is global by null intersection.

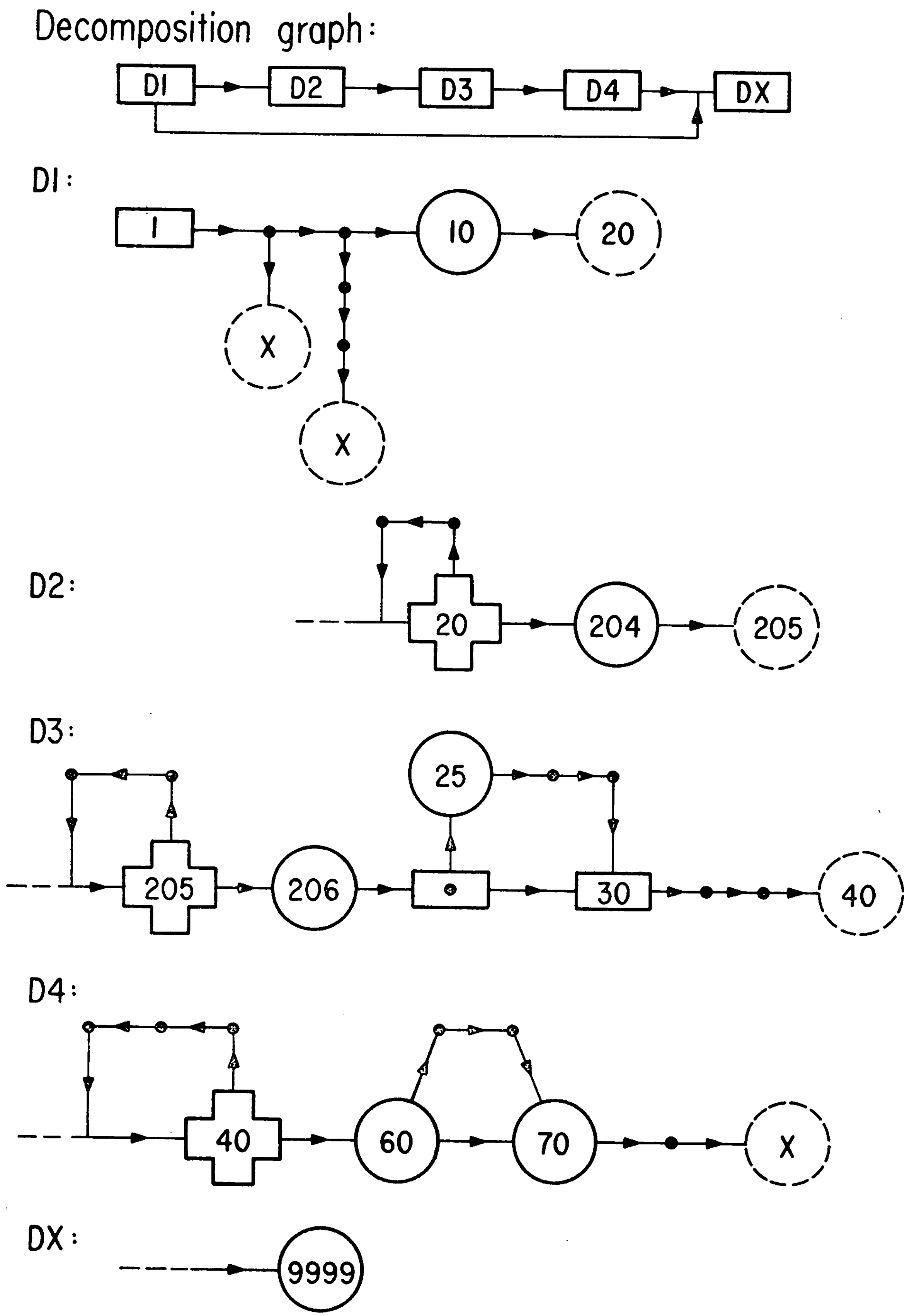


Fig. 1. Graph of Subroutine Note*

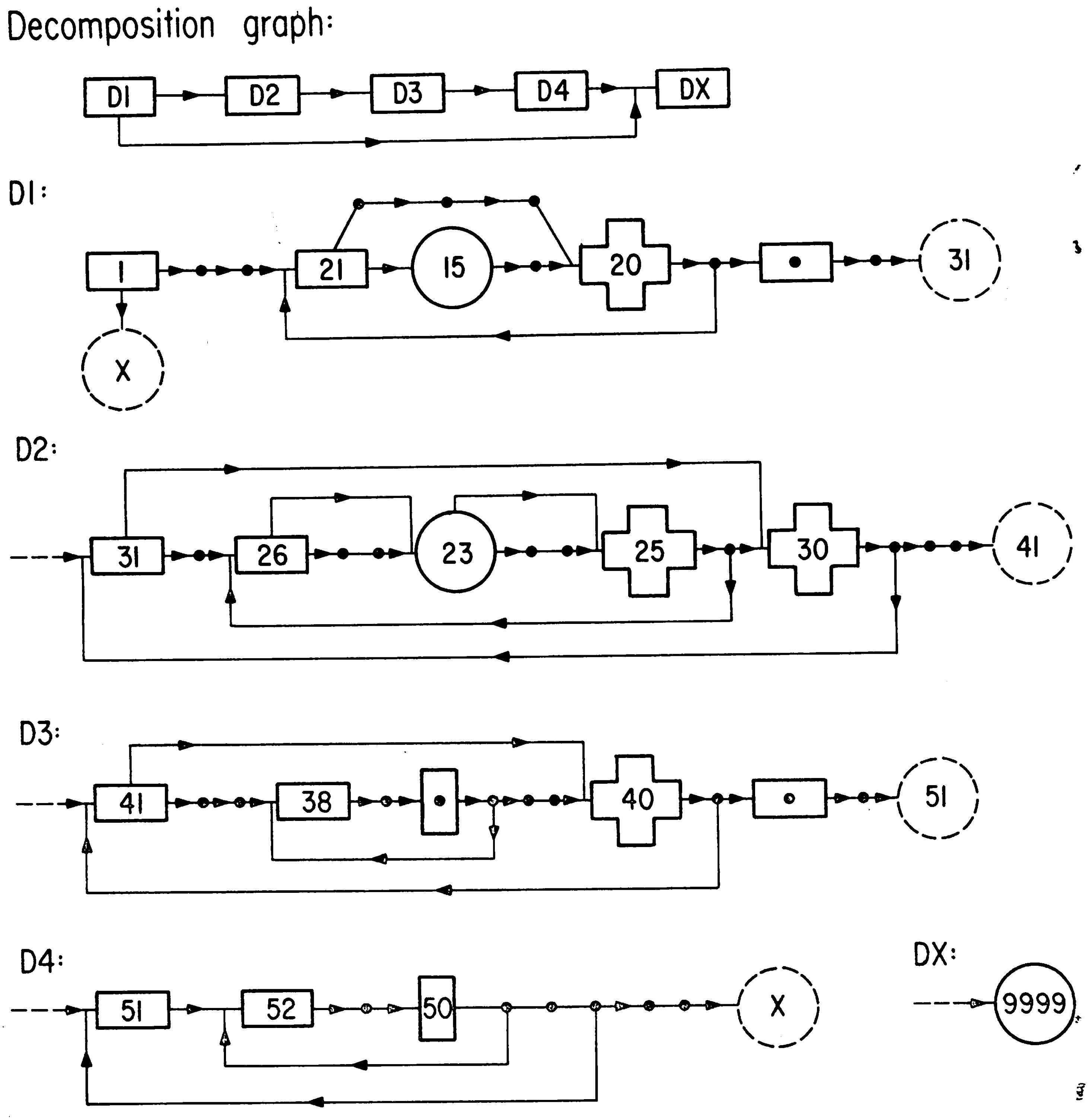


Fig. 2. Graph of Subroutine Consol*

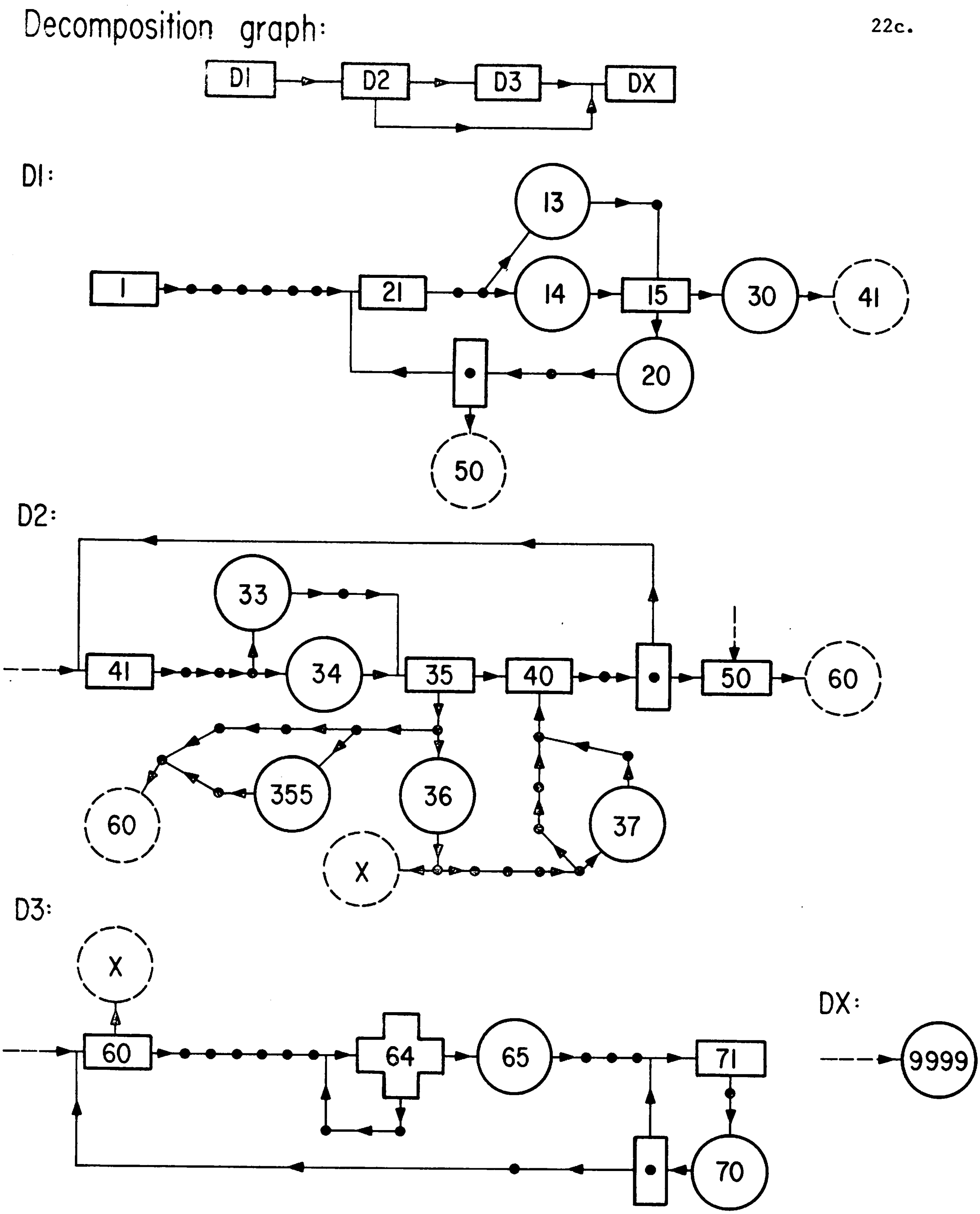


Fig. 3. Graph of Subroutine Apply*

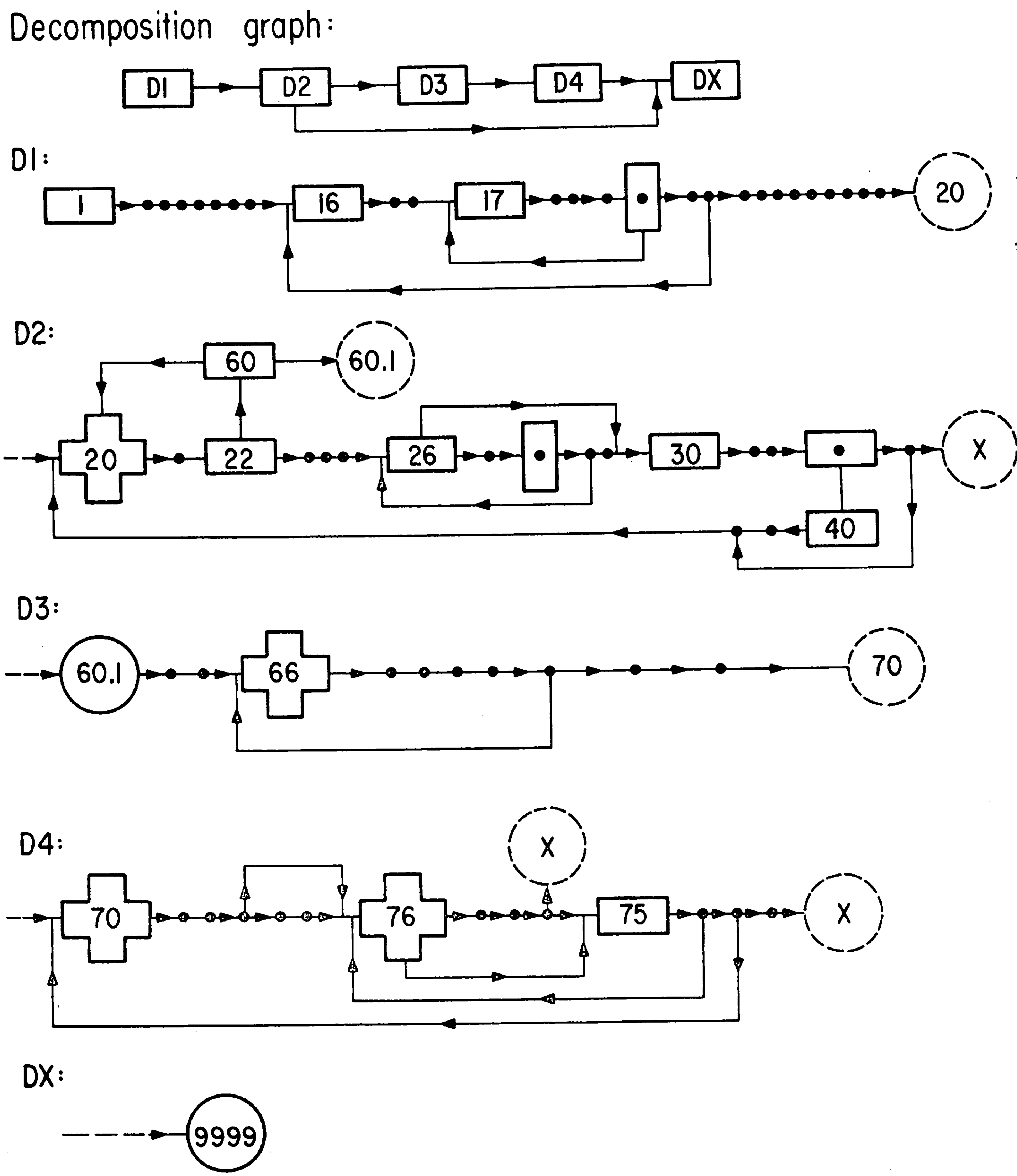


Fig. 4. Graph of Coset*

```
COND          <*  GC(NE)   *>    GLOBAL
COND  N 1:  <*  K, 0<=IX,IY<=N   *>
COMMENT   LET X,Y BE THE INITIAL VALUES OF EC(IX),EC(IY)
  1      IA = IX
         IF ( IA .EQ. 0 )  GO TO 9999
         IB = IY
 10      IF ( IB .NE. 0 ) GO TO 20
         IY = IA
         GO TO 9999
COND  N 9999 A:  <*  K   *>
COND  N 9999 B:  <*  GC(NB)   *>
COND  N 9999 C:  <*  ONE OF THE SET CI,CII,CIII,CIV   *>
C               CI:  <*  IX=0   *>
C              CII:  <*  IX>0, IY=IX, IB=0)   *>
C             CIII:  <*  IA=EC(IX)=IB=EC(IY)   *>
C              CIV:  <*  IX,IY>0,  IA=MIN(X,Y), IB=MAX(X,Y),
C                        EC(IB)=EC(IX)=EC(IY)=IA   *>
```

RHO NOTE 1 = ⟨ IA, IB, IY ⟩

PC NOTE 1 = ⟨* IA ¬= 0, IB = 0 *⟩

N 9999 A: null intersection

N 9999 B: Global

N 9999 C = CII: by assignments and path conditions

```
NOTE   CONTROL PATH   NUMBER   2

COND          <*  GC(NP)  *>    GLOBAL
COND   N 1:  <*   K,  0<=IX,IY<=N   *>
COMMENT   LET X,Y BE THE INITIAL VALUES OF EC(IX),EC(IY)
 1        IA = IX
          IF ( IA .EQ. 0 )   GO TO 9999
          IB = IY
 10       IF ( IB .NE. 0 ) GO TO 20
COND   N 20 A:  <*   K   *>
COND   N 20 B:  <*   IB=IY,   0<IX,IY<=N   *>
COND   N 20 C:  <*   EC(IA)=X,   IA BELONGS TO EC CHAIN WITH
C                     HEAD IX AND TAIL X.   *>
```

RHO NOTE 2 = ⟨IA, IB⟩

PC NOTE 2 = ⟨* IA ¬ = 0, IB ¬ = 0 *⟩

N 20 A: null intersection

N 20 B: assignment, null intersection

N 20 C: IA = IX by assignment

```
COND            <*  GC(NR)   *>     GLOBAL
COND  N 1:  <*  K,  0<=IX,IY<=N   *>
COMMENT   LET X,Y BE THE INITIAL VALUES OF EC(IX),EC(IY)
 1     IA = IX
       IF ( IA .EQ. 0 )  GO TO 9999
COND  N 9999 A:  <*  K   *>
COND  N 9999 B:  <*  GC(NR)   *>
COND  N 9999 C:  <*  ONE OF THE SET CI,CII,CIII,CIV   *>
C              CI:  <*  IY=0   *>
C             CII:  <*  IX>0, IY=IY, IB=0)   *>
C            CIII:  <*  IA=EC(IX)=IB=EC(IY)   *>
C             CIV:  <*  IX,IY>0,  IA=MIN(X,Y), IB=MAX(X,Y),
C                       EC(IB)=EC(IX)=EC(IY)=IA   *>
```

RHO NOTE 3 = ⟨ IA ⟩

PC NOTE 3 = ⟨* IA = 0 *⟩

N 9999 A: null intersection

N 9999 B: global

N 9999 C = CI: path condition

```
        NOTE   CONTROL PATH   NUMBER   4

     COND   N 20 A:  <*   K   *>
     COND   N 20 B:  <*  IB=IY,   0<IX,IY<=N   *>
     COND   N 20 C:  <*  EC(IA)=Y,  IA BELONGS TO EC CHAIN WITH
     C                          HEAD IX AND TAIL Y.   *>
      20    IF ( KV(IA) .GE. 0 )  GO TO 204
            IA = - KV(IA)
            GO TO 20
     COND   N 20 A:  <*   K   *>
     COND   N 20 B:  <*  IB=IY,   0<IX,IY<=N   *>
     COND   N 20 C:  <*  EC(IA)=Y,  IA BELONGS TO EC CHAIN WITH
     C                          HEAD IX AND TAIL X.   *>
```

RHO NOTE 4 = < IA >

PC NOTE 4 = <* KV(IA) < 0 *>

N 20 A;B: null intersection

N 20 C: definition of EC, path condition says -KV(IA)
is next lower member of EC chain from IX to X.

NOTE CONTROL PATH NUMBER 5

```
      COND  N 20 A:  <*  K  *>
      COND  N 20 B:  <*  IB=IY,  0<IX,IY<=N  *>
      COND  N 20 C:  <*  EC(IA)=X,  IA BELONGS TO EC CHAIN WITH
      C                   HEAD IX AND TAIL X.  *>
       20    IF ( KV(IA) .GE. 0 )  GO TO 204
       204   IA = KV(IA)
      COND  N 205 A:  <*  K  *>
      COND  N 205 B:  <*  IA=X,  0<IX,IY<=N  *>
      COND  N 205 C:  <*  EC(IB)=Y,  IB BELONGS TO EC CHAIN WITH
      C                   HEAD IY AND TAIL Y.  *>
```

RHO NOTE 5:= <IA>

PC NOTE 5 = <* KV(IA) >= 0 *>

N 205 A,C: null intersection

N 205 B: definition of EC and path condition say

KV(IA) = EC(IA) = X, null intersection

```
NOTE   CONTROL PATH   NUMBER   6

COND   N 205 A:   <*   K   *>
COND   N 205 B:   <*   IA=X,   0<IX,IY<=N   *>
COND   N 205 C:   <*   EC(IB)=Y,   IB BELONGS TO EC CHAIN WITH
C                      HEAD IY AND TAIL Y.   *>
 205    IF  ( KV(IB)  .GT.  0 )  GO TO 206
        IB  =  -  KV(IB)
        GO TO 205
COND   N 205 A:   <*   K   *>
COND   N 205 B:   <*   IA=X,   0<IX,IY<=N   *>
COND   N 205 C:   <*   EC(IB)=Y,   IB BELONGS TO EC CHAIN WITH
C                      HEAD IY AND TAIL Y.   *>
```

Path 6 Proof analogous to path 4

```
      COND  N 205 A:  <*  K  *>
      COND  N 205 B:  <*  IA=X,   0<IX,IY<=N   *>
      COND  N 205 C:  <*  EC(IB)=Y,  IB BELONGS TO EC CHAIN WITH
      C                   HEAD IY AND TAIL Y.  *>
       205   IF ( KV(IB) .GE. 0 ) GO TO 206
       206   IB = KV(IB)
      COND  N 206.1 A:  <*  K  *>
      COND  N 206.1 B:  <*  0<IA=X<=N,   0<IB=Y<=N   *>
```

Path 7 Proof analogous to path 5

```
NOTE    CONTROL PATH   NUMBER    8

COND   N 206.1 A:   <*   K   *>
COND   N 206.1 B:   <*   0<IA=X<=N,   0<IB=Y<=N   *>
          IF   (IB.EQ.IA)   GO TO 9999
          IF   (IB.GT.IA)   GO TO 30
 25       IT = IA
          IA = IB
          IB = IT
COND   N 30 A:   <*   K   *>
COND   N 30 B:   <*   0<IA<IB,   IA=MIN(X,Y),   IB=MAX(X,Y)   *>
COMMENT    LET  A BE SUCH THAT I(A)=IA, SIMILARLY I(B)=IB
```

RHO NOTE 8 = ⟨ IA, IB, IT ⟩

PC NOTE 8 = ⟨* IA > IB *⟩

N 30 A: null intersection

N 30 B: path condition assignments result in exchange
of values of IA and IB.

```
COND N 206.1 A: <* K *>
COND N 206.1 B: <* 0<IA=X<=N, 0<IB=Y<=N *>
      IF (IB.EQ.IA) GO TO 9999
      IF (IB.GT.IA) GO TO 30
COND N 30 A: <* K *>
COND N 30 B: <* 0<IA<IB, IA=MIN(X,Y), IB=MAX(X,Y) *>
COMMENT LET A BE SUCH THAT I(A)=IA, SIMILARLY I(B)=IB
```

RHO NOTE 9 = $\langle \emptyset \rangle$

PC NOTE 9 = $\langle IA < IB \rangle$

N 30 A: null intersection

N 30 B: path condition

```
COND  N 206.1 A:  <*  K  *>
COND  N 206.1 B:  <*  0<IA=X<=N,  0<IB=Y<=N  *>
        IF  (IB.EQ.IA)  GO TO 9999
COND  N 9999 A:  <*  K  *>
COND  N 9999 B:  <*  GC(NP)  *>
COND  N 9999 C:  <*  ONE OF THE SET CI,CII,CIII,CIV  *>
C               CI:  <*  IX=0  *>
C              CII:  <*  IX>0, IY=IX, IB=0)  *>
C             CIII:  <*  IA=EC(IY)=IB=EC(IY)  *>
C              CIV:  <*  IX,IY>0,  IA=MIN(X,Y), IB=MAX(X,Y),
C                        EC(IB)=EC(IX)=EC(IY)=IA  *>
```

RHO NOTE 10 = ⟨∅⟩

PC NOTE 10 = ⟨* IB = IA *⟩

N 9999 A: null intersection

N 9999 B: global

N 9999 C = CIII: path condition

```
NOTE    CONTROL PATH  NUMBER  11

COND   N 30 A:   <*   K   *>
COND   N 30 B:   <*   0<IA<IB,   IA=MIN(X,Y),  IB=MAX(X,Y)   *>
COMMENT   LET  A BE SUCH THAT I(A)=IA, SIMILARLY I(B)=IB
 30     KV(IB) = -IA
        IT = IB
        M=0
COND   N 40 A:   <*   K FOR INITIAL VALUES OF KV,KL,KR,KF:
C                     KL,KR,KF UNCHANGED, KV(IT) UNCHANGED
C                     FOR IT ¬< C(B)   *>
COND   N 40 B:   <*   KV(IB)=-IA   *>
COND   N 40 C:   <*   0<IA<IB,   IA=MIN(X,Y),  IB=MAX(X,Y)   *>
COND   N 40 D:   <*   0<=M<P(B),   IT=KL(IB)**M   *>
COND   N 40 E:   <*   KV(I)=IA FOR I=KL(IB)**Q, 1<=Q<=M   *>
```

RHO NOTE 11 = ⟨KV(IB),IT,M⟩

PC NOTE 11 = ∅

N 40 A;C: null intersection

N 40 B;D: assignment

N 40 E: vacuous

NOTE CONTROL PATH NUMBER 12

```
COND   N 40 A:  <*  K FOR INITIAL VALUES OF KV,KL,KR,KF:
C                    KL,KR,KF UNCHANGED, KV(IT) UNCHANGED
C                    FOR IT ¬< C(B)   *>
COND   N 40 B:  <*  KV(IB)=-IA   *>
COND   N 40 C:  <*  0<IA<IB,  IA=MIN(X,Y), IB=MAX(X,Y)   *>
COND   N 40 D:  <*  0<=M<P(B),  IT=KL(IB)**M   *>
COND   N 40 E:  <*  KV(I)=IA FOR I=KL(IB)**O, 1<=O<=M   *>
 40     IF ( KL(IT) .EQ. IB ) GO TO 60
        IT = KL(IT)
        KV(IT) = IA
        M=M+1
        GO TO 40
COND   N 40 A:  <*  K FOR INITIAL VALUES OF KV,KL,KR,KF:
C                    KL,KR,KF UNCHANGED, KV(IT) UNCHANGED
C                    FOR IT ¬< C(B)   *>
COND   N 40 B:  <*  KV(IB)=-IA   *>
COND   N 40 C:  <*  0<IA<IB,  IA=MIN(X,Y), IB=MAX(X,Y)   *>
COND   N 40 D:  <*  0<=M<P(B),  IT=KL(IB)**M   *>
COND   N 40 E:  <*  KV(I)=IA FOR I=KL(IB)**O, 1<=O<=M   *>
```

RHO NOTE 12 = 〈IT, KV(IT), M〉

PC NOTE 12 = 〈* KL(IT) ¬ = IB *〉

N 40 A,B,C: null intersection

N 40 D: path condition implies initial value of M < P(B)-1 since M < P(B) and KL(IB) ** (M+1) ¬ = IB. Remainder follows from assignments

N 40 E: initial condition and assignment

```
COND  N 40 A:  <*  K FOR INITIAL VALUES OF KV,KL,KR,KF:
C                  KL,KR,KF UNCHANGED, KV(IT) UNCHANGED
C                  FOR IT ¬< C(B)  *>
COND  N 40 B:  <*  KV(IB)=-IA  *>
COND  N 40 C:  <*  0<IA<IB,  IA=MIN(X,Y), IB=MAX(X,Y)  *>
COND  N 40 D:  <*  0<=M<P(B),  IT=KL(IB)**M  *>
COND  N 40 E:  <*  KV(I)=IA FOR I=KL(IB)**Q,  1<=Q<=M  *>
 40    IF ( KL(IT) .EQ. IB ) GO TO 60
 60    IF ( IT .EQ. IB )  GO TO 70
       KL(IT) = KL(IA)
       KL(IA) = KL(IB)
 70    KL(IB) = ID
       ID = IB
COND  N 9999 A:  <*  K  *>
COND  N 9999 B:  <*  GC(NP)  *>
COND  N 9999 C:  <*  ONE OF THE SET CI,CII,CIII,CIV  *>
C            CI:  <*  IX=0  *>
C           CII:  <*  IX>0, IY=IX, IB=0)  *>
C          CIII:  <*  IA=EC(IX)=IB=EC(IY)  *>
C           CIV:  <*  IX,IY>0,  IA=MIN(X,Y), IB=MAX(X,Y),
C                     EC(IB)=EC(IX)=EC(IY)=IA  *>
```

RHO NOTE 13: <KL(IT),KL(IA),KL(IB),ID>

PC NOTE 13: <* IT = KL(IB)**(P(B)-1) ¬ = IB *>

N 9999 A: Lemma 4 Sect 4 N 40 and path give function of Lemma 4

N 9999 B: global

N 9999 C = CIV: null intersection for first part Lemma 4 for last.

```
      NOTE   CONTROL PATH   NUMBER  14

COND   N 40 A:  <*  K FOR INITIAL VALUES OF KV,KL,KR,KF:
C                   KL,KR,KF UNCHANGED, KV(IT) UNCHANGED
C                   FOR IT ¬< C(B)   *>
COND   N 40 B:  <*  KV(IB)=-IA   *>
COND   N 40 C:  <*  0<IA<IB,  IA=MIN(X,Y), IB=MAX(X,Y)   *>
COND   N 40 D:  <*  0<=M<P(B),  IT=KL(IB)**M   *>
COND   N 40 E:  <*  KV(I)=IA FOR I=KL(IB)**Q, 1<=Q<=M   *>
 40     IF ( KL(IT) .EQ. IB ) GO TO 60
 60     IF ( IT .EQ. IB )  GO TO 70
 70     KL(IB)  = ID
        ID = IB
COND   N 9999 A:  <*  K   *>
COND   N 9999 B:  <*  GC(NB)   *>
COND   N 9999 C:  <*  ONE OF THE SET CI,CII,CIII,CIV   *>
C               CI:  <*  IX=0   *>
C              CII:  <*  IX>0, IY=IX, IB=0)   *>
C             CIII:  <*  IA=EC(IX)=IB=EC(IY)   *>
C              CIV:  <*  IX,IY>0,  IA=MIN(X,Y), IB=MAX(X,Y),
C                        EC(IB)=EC(IX)=EC(IY)=IA   *>
```

Path 14 is alternate case of Lemma 4.

Proof of termination for NOTE

Q(NOTE) = ⟨ F 20, F 205, F 40⟩

Expressions: e1 = λ, e2 = -IA, e3 = -IB, e4 = M.

Proof that ⟨e1,e2,e3,e4⟩ is a controlled set of expressions:

Path 1 : 20 —> 20

PC 1 = ⟨* KV(IA) < 0 *⟩

λ is constant along path, -iA < 0 at F 20 and increasing along path by definition of EC chain (see control path H).

Path 2 : 20 —> 205

PC 2 = ⟨* KV(IA) < 0 *⟩

λ is increasing along path, constant at F 205.

Path 3 : 205 —> 205

PC 3 = ⟨* KV(IB) < 0 *⟩

λ, -IA constant thus non-decreasing along path. -IB < 0 at F 205 and increasing along path by definition of EC chain (see Control path 6)

Path 4a,b : 205 —> 40

PC 4a ⟨* IB > IA *⟩

PC 4b ⟨* IB < IA *⟩

λ is increasing along path and constant at F 40.

Path 5 : 40 ⟶ 40

PC 5 = $\langle$* KL(IT) $\neq$ IB *$\rangle$

λ, -IA, -IB are constant along path thus non-decreasing.

M is increasing along path by assignment M = M+1.

M < P(B) at F 40 (COND 40 D).

Q.E.D.

5.2. Proof of consistency for subroutine CONSOL.

This section contains proof that if $S \in$ COND-IN(CONSOL) and P = CONSOL then $P(S) \in$ COND-OUT(CONSOL)

$$\text{COND-IN(CONSOL)} = \langle * \ K,\ ID = 0,\ GC(NR) \ * \rangle$$

$$\text{COND-OUT(CONSOL)} = \langle * \ K,\ ID = 0,\ GC(NR),\ NVAC = 0,\ KN = IM(KV\#(KN)) \ * \rangle$$

ρ(CONSOL) $\cap \Delta$ GC(NR) = N, KN

N occurs only in ρ(PATH 26),

KN occurs only in ρ(PATH 7), in each case GC(NR) is consistent by assignment. The remainder of GC(NR) is valid by null intersection. The global condition on KN is also true by null intersection.

```
COND  C  1:  <* K,ID=0,GC(NR) *>
COND  GLOBAL  <* ID=0,GC(NR) *>
C  X# DENOTES THE INITIAL VALUE OF THE VARIABLE X,
C  X DENOTES THE CURRENT VALUE
 1     IF (NVAC.EQ.0)  GO TO 9999
        M = 0
       I=1
COND  C  21 A:  <*  1<=I<=N,  M=|<II: II<I, KV#(II)=II>|  *>
COND  C  21 B:  <* 1<=KV(K)<KV(KK)<=M FOR K,KK SUCH THAT
C                    1<=K<KK<I, KV#(K)=K, KV#(KK)=KK;
C                     KV(K)=-KV(KV#(K))  FOR K SUCH THAT
C                     1<=K<I, KV#(K)¬=K;  FOR EACH MM SUCH THAT
C                     1<=MM<=M  KV(K)=MM FOR SOME K,
C                    1<=K<I,KV#(K)=K *>
COND  C  21 C:  <* KV(K)=KV#(K), I<=K<=N *>
COND  C  21 D:  <* KL=KL#, KF=KF#, KR=KR# *>
```

RHO CONSOL 1 = ⟨M, I⟩

PC CONSOL 1 = ⟨* NVAC > 0 *⟩

C 21 A: Assignment

C 21 B: Vacuous

C 21 C;D: Null intersection

CONSOL CONTROL PATH NUMBER 2

```
COND  C  1:  <* K,ID=0,GC(NR) *>
COND  GLOBAL  <* ID=0,GC(NR) *>
C  X# DENOTES THE INITIAL VALUE OF THE VARIABLE X,
C  X DENOTES THE CURRENT VALUE
 1    IF (NVAC.EQ.0)  GO TO 9999
COND  C  9999:  <* K,ID=0,NVAC=0,H(AK)=H(AK#),
C                         KN=IM(KV#(KN#))  *>
```

RHO CONSOL 2 = $\langle \emptyset \rangle$

PC CONSOL 2 = $\langle *\ \mathrm{NVAC} = 0\ * \rangle$

C 9999: Null intersection and path condition

```
COND  C  21 A:  <*  1<=I<=N,  M=|<II; II<I, KV#(II)=II>|   *>
COND  C  21 B:  <* 1<=KV(K)<KV(KK)<=M FOR K,KK SUCH THAT
C                  1<=K<KK<I, KV#(K)=K, KV#(KK)=KK:
C                   KV(K)=-KV(KV#(K)) FOR K SUCH THAT
C                   1<=K<I, KV#(K)¬=K;  FOR EACH MM SUCH THAT
C                   1<=MM<=M  KV(K)=MM FOR SOME K,
C                  1<=K<I,KV#(K)=K *>
COND  C  21 C:  <* KV(K)=KV#(K), I<=K<=N *>
COND  C  21 D:  <* KL=KL#, KP=KP#, KR=KR# *>
 21    IF ( KV(I) .NE. I ) GO TO 15
       M = M + 1
       KV(I) = M
       GO TO 20
COND  C  20 A:  <*  1<=I<=N, M=|<II; II<=I, KV#(II)=II>|   *>
COND  C  20 B:  <*  C  21 B FOR M=M-1,(KV#(I)=I, KV(I)=M)
C               OR C 21 B, (KV#(I)¬=I, KV(I)=-KV(KV#(I)) ) *>
COND  C  20 C:  <* KV(K)=KV#(K), I<K<=N *>
COND  C  20 D:  <* C  21 D *>
```

RHO CONSOL 3 = ⟨M, KV(I)⟩

PC CONSOL 3 = ⟨* KV(I) = I *⟩

C 20 A: Assignment and path condition. The new element of set ⟨II⟩ is counted by increase of M.

C 20 B(FIRST OPTION): Path condition and assignment

C 20 C,D: Null intersection

```
      COND  C  21 A:  <*  1<=I<=N,  M=|<II: II<I, KV#(II)=II>|   *>
      COND  C  21 B:  <* 1<=KV(K)<KV(KK)<=M FOR K,KK SUCH THAT
      C                  1<=K<KK<I, KV#(K)=K, KV#(KK)=KK:
      C                   KV(K)=-KV(KV#(K)) FOR K SUCH THAT
      C                   1<=K<I, KV#(K)¬=K:  FOR EACH MM SUCH THAT
      C                   1<=MM<=M  KV(K)=MM FOR SOME K,
      C                  1<=K<I,KV#(K)=K *>
      COND  C  21 C:  <* KV(K)=KV#(K), I<=K<=N *>
      COND  C  21 D:  <* KL=KL#, KF=KF#, KR=KR# *>
 21     IF ( KV(I) .NE. I ) GO TO 15
 15     J = KV(I)
        KV(I) = -KV(J)
      COND  C  20 A:  <*  1<=I<=N, M=|<II; II<=I, KV#(II)=II>|   *>
      COND  C  20 B:  <*  C  21 B FOR M=M-1,(KV#(I)=I, KV(I)=M)
      C                OR C 21 B, (KV#(I)¬=I, KV(I)=-KV(KV#(I)) ) *>
      COND  C  20 C:  <* KV(K)=KV#(K), I<K<=N *>
      COND  C  20 D:  <* C  21 D *>
```

RHO CONSOL 4: ⟨J, KV(I)⟩

PC CONSOL 4: ⟨* KV(I) ¬ = I *⟩

C 20 A: Path condition, no new element added to set ⟨II⟩
by including I in bounds

C 20 B(SECOND OPTION): Path condition and assignment
since J = KV (I)

C 20 C,D: Null intersection

```
      COND  C  20 A:  <*  1<=I<=N, M=|<II: II<=I, KV#(II)=II>|  *>
      COND  C  20 B:  <*  C  21 B FOR M=M-1,(KV#(I)=I,  KV(I)=M)
      C               OR C 21 B, (KV#(I)¬=I, KV(I)=-KV(KV#(I)) ) *>
      COND  C  20 C:  <* KV(K)=KV#(K), I<K<=N *>
      COND  C  20 D:  <* C  21 D *>
       20   I=I+1
            IF (I.LE.N)  GO TO 21
      COND  KVIM:  <* M=MCOS#,KV(I)=IM(I)  IF KV#(I)=I,
      C               KV(I)=-IM(KV#(I))  IF KV#(I)¬=I, 1<=I<=N  *>
      COND  C  20.2:  <* C  21 D, KVIM *>
```

RHO CONSOL 5 = ⟨ I ⟩

PC CONSOL 5 = ⟨* I > N *⟩

C 20.2: null intersection for C 21 D, C 20 A,B and
path condition for KVIM

```
      COND  C  20 A:  <*  1<=I<=N,  M=|<II: II<=I, KV#(II)=II>|  *>
      COND  C  20 B:  <*  C  21 B FOR M=M-1,(KV#(I)=I, KV(I)=M)
      C                    OR C 21 B, (KV#(I)¬=I, KV(I)=-KV(KV#(I)) ) *>
      COND  C  20 C:  <* KV(K)=KV#(K), I<K<=N *>
      COND  C  20 D:  <* C  21 D *>
       20   I=I+1
            IF (I.LE.N)  GO TO 21
      COND  C  21 A:  <*  1<=I<=N,  M=|<II: II<I, KV#(II)=II>|  *>
      COND  C  21 B:  <* 1<=KV(K)<KV(KK)<=M FOR K,KK SUCH THAT
      C                   1<=K<KK<I, KV#(K)=K, KV#(KK)=KK;
      C                    KV(K)=-KV(KV#(K)) FOR K SUCH THAT
      C                    1<=K<I, KV#(K)¬=K;  FOR EACH MM SUCH THAT
      C                    1<=MM<=M  KV(K)=MM FOR SOME K,
      C                   1<=K<I,KV#(K)=K *>
      COND  C  21 C:  <* KV(K)=KV#(K), I<=K<=N *>
      COND  C  21 D:  <* KL=KL#, KF=KF#, KB=KB# *>
```

RHO CONSOL 6 = ⟨I⟩

PC CONSOL 6 = ⟨* I <= N *⟩

C 21 A: C 20 with path condition and increase of I

C 21 B,C: C 20 B,C and increase of I

C 21 D: null intersection

```
COND  KVIM:  <* M=MCOS#,KV(I)=IM(I) IF KV#(I)=I,
C                KV(I)=-IM(KV#(I)) IF KV#(I)¬=I, 1<=I<=N  *>
COND  C  20.2:  <* C  21 D, KVIM *>
        KN = IABS( KV(KN) )
        I=1
COND  GLOBAL  <*  KN=|IM(KV#(KN#))|  *>
COND  C  31 A:  <* KVIM, KL=KL# *>
COND  C  31 B:  <* 1<=I<=N *>
COND  C  31 C:  <* KP(II,J)=IM(KV#(KP#(II,J))),
C                   KR(II,J)=IM(KV#(KR#(II,J))),
C                   1<=J<=NJ, 1<=II<I,KV#(II)=II *>
COND  C  31 D:  <* KP(II,J)=KP#(II,J), KR(II,J)=KR#(II,J)
C                   FOR I<=II<=N, 1<=J<=NJ  *>
```

RHO CONSOL 7 = ⟨KN, I⟩

PC CONSOL 7 = ⟨* Ø *⟩

COND 31 A,D: Null intersection

COND 31 B: Assignment

COND 31 C: Vacuous

```
      COND   GLOBAL  <*  KN=|IM(KV#(KN#))|   *>
      COND   C  31 A:  <* KVIM, KL=KL# *>
      COND   C  31 B:  <* 1<=I<=N *>
      COND   C  31 C:  <* KF(II,J)=IM(KV#(KF#(II,J))),
      C                    KR(II,J)=IM(KV#(KR#(II,J))),
      C                   1<=J<=NJ,  1<=II<I,KV#(II)=II  *>
      COND   C  31 D:  <* KF(II,J)=KF#(II,J), KR(II,J)=KR#(II,J)
      C                    FOR I<=II<=N,  1<=J<=NJ   *>
       31     IF ( KV(I) .LT. 0 )  GO TO 30
              J=1
      COND   C  26 A:  <* C   31 A,B,C,  KV(I)>0   *>
      COND   C  26 B:  <* 1<=J<=NJ *>
      COND   C  26 C:  <* KF(I,JJ)=IM(KV#(KF#(I,JJ))),
      C                   KR(I,JJ)=IM(KV#(KR#(I,JJ))), 1<=JJ<J   *>
      COND   C  26 D:  <* KF(II,JJ)=KF#(II,JJ), KR(II,JJ)=KR#(II,JJ)
      C                   IF  I<II OR I=II, J<=JJ *>
```

RHO CONSOL 8 = $\langle J \rangle$

PC CONSOL 8 = $\langle * \ KV(I) > 0 \ * \rangle$

C 26 A,D: Null intersection

C 26 B: Assignment

C 26 C: Vacuous

CONSOL CONTROL PATH NUMBER 9

```
COND  GLOBAL  <*  KN=|IM(KV#(KN#))|  *>
COND  C  31 A:  <* KVIM, KL=KL# *>
COND  C  31 B:  <* 1<=I<=N *>
COND  C  31 C:  <* KF(II,J)=IM(KV#(KF#(II,J))),
C                    KR(II,J)=IM(KV#(KR#(II,J))),
C                   1<=J<=NJ, 1<=II<I,KV#(II)=II *>
COND  C  31 D:  <* KF(II,J)=KF#(II,J) , KR(II,J)=KR#(II,J)
C                    FOR I<=II<=N, 1<=J<=NJ  *>
 31    IF ( KV(I) .LT. 0 )  GO TO 30
COND  C  30 A:  <* C  31 A,B,C *>
COND  C  30 B:  <* KV(I)>0 AND KF(I,J)=IM(KV#(KF#(I,J))),
C                   KR(I,J)=IM(KV#(KR#(I,J))), 1<=J<=NJ  *>
C               OR  <*  KV(I)<0  *>
COND  C  30 C:  <* KF(II,J)=KF#(II,J) , I<II<=N, 1<=J<=NJ *>
```

RHO CONSOL 9 = Ø

PC CONSOL 9 = <* KV(I) < 0 *>

C 30 A,C: Null intersection

C 30 B(SECOND CHOICE): Path condition

```
      COND  C  26 A:  <* C  31 A,B,C,  KV(I)>0  *>
      COND  C  26 B:  <* 1<=J<=NJ *>
      COND  C  26 C:  <* KF(I,JJ)=IM(KV#(KF#(I,JJ))),
      C                  KR(I,JJ)=IM(KV#(KR#(I,JJ))),  1<=JJ<J  *>
      COND  C  26 D:  <* KF(II,JJ)=KF#(II,JJ), KR(II,JJ)=KR#(II,JJ)
      C                  IF  I<II OR I=II,  J<=JJ *>
 26   IF ( KF(I,J) .EQ. 0 )  GO TO 23
      KT = KF(I,J)
      KF(I,J) = IABS( KV(KT) )
 23   IF ( KR(I,J) .EQ. 0 )  GO TO 25
      KT = KR(I,J)
      KR(I,J) = IABS( KV(KT) )
      COND  C  25 A:  <* C  26 A,B,C,D FOR J<JJ  *>
      COND  C  25 B:  <* KF(I,J)=IM(KV#(KF#(I,J))),
      C                  KR(I,J)=IM(KV#(KR#(I,J)))  *>
```

RHO CONSOL 10:= ⟨KT, KF(I,J), KR(I,J)⟩

PC CONSOL 10 = ⟨KF(I,J) > 0, KR(I,J) > 0⟩

COND C 25A: Null intersection

COND C 25B: Path conditions, definition of IM and assignments

```
COND   C  26 A:  <* C  31 A,B,C,  KV(I)>0  *>
COND   C  26 B:  <* 1<=J<=NJ *>
COND   C  26 C:  <* KF(I,JJ)=IM(KV#(KF#(I,JJ))),
C                   KR(I,JJ)=IM(KV#(KR#(I,JJ))), 1<=JJ<J  *>
COND   C  26 D:  <* KF(II,JJ)=KF#(II,JJ), KR(II,JJ)=KR#(II,JJ)
C                   IF  I<II OR I=II, J<=JJ *>
 26     IF ( KF(I,J) .EQ. 0 )  GO TO 23
        KT = KF(I,J)
        KF(I,J) = IABS( KV(KT) )
 23     IF ( KR(I,J) .EQ. 0 )  GO TO 25
COND   C  25 A:  <* C  26 A,B,C,D FOR J<JJ  *>
COND   C  25 B:  <* KF(I,J)=IM(KV#(KF#(I,J))),
C                   KR(I,J)=IM(KV#(KR#(I,J)))  *>
```

(PATH 11) Proof of consistency analogous to that for path 10.

```
      COND  C  26 A:  <* C  31 A,B,C,  KV(I)>0  *>
      COND  C  26 B:  <* 1<=J<=NJ *>
      COND  C  26 C:  <* KF(I,JJ)=IM(KV#(KF#(I,JJ))),
      C                  KR(I,JJ)=IM(KV#(KR#(I,JJ))),  1<=JJ<J  *>
      COND  C  26 D:  <* KF(II,JJ)=KF#(II,JJ), KR(II,JJ)=KR#(II,JJ)
      C                  IF  I<II OR I=II, J<=JJ *>
   26    IF ( KF(I,J) .EQ. 0 )  GO TO 23
   23    IF ( KR(I,J) .EQ. 0 )  GO TO 25
         KT = KR(I,J)
         KR(I,J) = IABS( KV(KT) )
      COND  C  25 A:  <* C  26 A,B,C,D FOR J<JJ  *>
      COND  C  25 B:  <* KF(I,J)=IM(KV#(KF#(I,J))),
      C                  KR(I,J)=IM(KV#(KR#(I,J)))  *>
```

(PATH 12) Proof of consistency analogous to that for Path 10.

CONSOL CONTROL PATH NUMBER 13

```
COND   C   26 A:  <* C   31 A,B,C,  KV(I)>0   *>
COND   C   26 B:  <* 1<=J<=NJ *>
COND   C   26 C:  <* KF(I,JJ)=IM(KV#(KF#(I,JJ))),
C                    KR(I,JJ)=IM(KV#(KR#(I,JJ))),  1<=JJ<J   *>
COND   C   26 D:  <* KF(II,JJ)=KF#(II,JJ) , KR(II,JJ)=KR#(II,JJ)
C                    IF  I<II OR I=II,  J<=JJ *>
 26      IF ( KF(I,J)  .EQ. 0 )   GO TO 23
 23      IF ( KR(I,J)  .EQ. 0 )   GO TO 25
COND   C  25 A:  <* C   26 A,B,C,D FOR J<JJ   *>
COND   C  25 B:  <* KF(I,J)=IM(KV#(KF#(I,J))),
C                   KR(I,J)=IM(KV#(KR#(I,J)))   *>
```

(PATH 13) Proof of consistency analogous to that for Path 10.

```
COND  C  25 A:  <* C  26 A,B,C,D FOR J<JJ  *>
COND  C  25 B:  <* KF(I,J)=IM(KV#(KF#(I,J))),
C                  KR(I,J)=IM(KV#(KR#(I,J)))  *>
 25     J=J+1
        IF (J.LE.NJ)  GO TO 26
COND  C  30 A:  <* C  31 A,B,C *>
COND  C  30 B:  <* KV(I)>0 AND KF(I,J)=IM(KV#(KF#(I,J))),
C                  KR(I,J)=IM(KV#(KR#(I,J))), 1<=J<=NJ  *>
C               OR  <*  KV(I)<0  *>
COND  C  30 C:  <* KF(II,J)=KF#(II,J), I<II<=N, 1<=J<=NJ *>
```

RHO CONSOL 14 = ⟨ J⟩

PC CONSOL 14 = ⟨ J > NJ⟩

C 30 A,C : Null intersection

C 30 B (FIRST CHOICE): C 25 and path condition

```
COND  C  25 A:  <* C  26 A,B,C,D FOR J<JJ  *>
COND  C  25 B:  <* KF(I,J)=IM(KV#(KF#(I,J))),
C                  KR(I,J)=IM(KV#(KR#(I,J)))  *>
 25     J=J+1
        IF (J.LE.NJ)  GO TO 26
COND  C  26 A:  <* C  31 A,B,C,  KV(I)>0  *>
COND  C  26 B:  <* 1<=J<=NJ *>
COND  C  26 C:  <* KF(I,JJ)=IM(KV#(KF#(I,JJ))),
C                  KR(I,JJ)=IM(KV#(KR#(I,JJ))),  1<=JJ<J  *>
COND  C  26 D:  <* KF(II,JJ)=KF#(II,JJ),  KR(II,JJ)=KR#(II,JJ)
C                  IF  I<II OR I=II,  J<=JJ *>
```

RHO CONSOL 15 = $\langle J \rangle$

PC CONSOL 15 = $\langle J \Leftarrow NJ \rangle$

C 26 A,D: Null intersection

C 26 B: Path condition

C 26 C: C 25 and assignment

```
COND   C  30 A:  <* C  31 A,B,C *>
COND   C  30 B:  <* KV(I)>0 AND KF(I,J)=IM(KV#(KF#(I,J))),
C                   KR(I,J)=IM(KV#(KR#(I,J))),  1<=J<=NJ   *>
C                OR  <*  KV(I)<0   *>
COND   C  30 C:  <* KF(II,J)=KF#(II,J), I<II<=N,  1<=J<=NJ  *>
 30     I=I+1
        IF  (I.LE.N)  GO TO 31
        J=0
        I=1
COND   C  41 A:  <*  1<=I<=N,  J=|<II; II<I,KV#(II)=II>|   *>
COND   C  41 B:  <* KL(JJ)=KV(JJ)=JJ,
C                   KF(JJ,K)=IM(KV#(KF#(IMI(JJ),K))),
C                  KR(JJ,K)=IM(KV#(KR#(IMI(JJ),K))),
C                   1<=JJ<=J,  1<=K<=NJ   *>
COND   C  41 C:  <*  KF(II,K)=IM(KV#(KF#(II,K))),
C                   KR(II,K)=IM(KV#(KR#(II,K))),  1<=K<=NJ,
C                   KV(II)=IM(II),  I<II<=N AND KV#(II)=II;
C                KV(II)=-IM(KV#(II)), KV#(II)¬=II,  I<II<=N   *>
```

RHO CONSOL 16 = $\langle I,J \rangle$

PC CONSOL 16 = $\langle *\ I > N\ * \rangle$

C 41 A: Assignment

C 41 B: Vacuous

C 41 C: C 30 with path condition

```
COND  C  30 A:  <* C  31 A,B,C *>
COND  C  30 B:  <* KV(I)>0 AND KF(I,J)=IM(KV#(KF#(I,J))),
C                  KR(I,J)=IM(KV#(KR#(I,J))), 1<=J<=NJ   *>
C               OR  <*  KV(I)<0  *>
COND  C  30 C:  <* KF(II,J)=KF#(II,J), I<II<=N, 1<=J<=NJ *>
 30     I=I+1
        IF  (I.LE.N)  GO TO 31
COND  GLOBAL  <*  KN=|IM(KV#(KN#))|  *>
COND  C  31 A:  <* KVIM, KL=KL# *>
COND  C  31 B:  <* 1<=I<=N *>
COND  C  31 C:  <* KF(II,J)=IM(KV#(KF#(II,J))),
C                   KR(II,J)=IM(KV#(KR#(II,J))),
C                  1<=J<=NJ, 1<=II<I,KV#(II)=II *>
COND  C  31 D:  <* KF(II,J)=KF#(II,J), KR(II,J)=KR#(II,J)
C                   FOR I<=II<=N, 1<=J<=NJ  *>
```

RHO CONSOL 17 = ⟨I⟩

PC CONSOL 17 = ⟨* I ≤ N *⟩

C 31 A,D: Null intersection

C 31 B: Path condition

C 31 C: C 30 and assignment

```
COND  C  41 A:  <*  1<=I<=N,  J=|<II; II<I,KV#(II)=II>|  *>
COND  C  41 B:  <* KL(JJ)=KV(JJ)=JJ,
C                   KF(JJ,K)=IM(KV#(KF#(IMI(JJ),K))),
C                  KR(JJ,K)=IM(KV#(KR#(IMI(JJ),K))),
C                   1<=JJ<=J,   1<=K<=NJ   *>
COND  C  41 C:  <*  KF(II,K)=IM(KV#(KF#(II,K))),
C                   KR(II,K)=IM(KV#(KR#(II,K))),  1<=K<=NJ,
C                   KV(II)=IM(II),  I<II<=N AND KV#(II)=II;
C               KV(II)=-IM(KV#(II)), KV#(II)¬=II, I<II<=N   *>
 41    IF  (KV(I).LT.0)  GO TO 40
       J=J+1
       K=1
COND  C  38 A:  <* 1<=I<=N, 1<=K<=NJ, KV(I)>0, J=IMI(I)  *>
COND  C  38 B:  <* C  41 B WITH J REPLACED BY J-1,
C                  KF(J,KK)=IM(KV#(KF#(IMI(J),KK))),
C                  KR(J,KK)=IM(KV#(KR#(IMI(J),KK))),
C                   1<=KK<K   *>
COND  C  38 C:  <* C  41 C *>
```

RHO CONSOL 18 = $\langle J,K \rangle$

PC CONSOL 18 = $\langle *\ KV(I) > 0\ * \rangle$

C 38 A: Assignment, IMI(I) = I < II; 1 <= II <= I,

KV (II) = II > 1. If KV (I) = I.

C 38 B: First part by increase in J and null intersection,

second part vacuous since K = 1.

C 38 C: Null intersection

```
COND  C  41 A:  <*  1<=I<=N,  J=|<II; II<I,KV#(II)=II>|  *>
COND  C  41 B:  <* KL(JJ)=KV(JJ)=JJ,
C                  KF(JJ,K)=IM(KV#(KF#(IMI(JJ),K))),
C                 KR(JJ,K)=IM(KV#(KR#(IMI(JJ),K))),
C                  1<=JJ<=J,  1<=K<=NJ  *>
COND  C  41 C:  <*  KF(II,K)=IM(KV#(KF#(II,K))),
C                  KR(II,K)=IM(KV#(KR#(II,K))), 1<=K<=NJ,
C                  KV(II)=IM(II),  I<II<=N AND KV#(II)=II;
C               KV(II)=-IM(KV#(II)), KV#(II)¬=II, I<II<=N  *>
 41    IF  (KV(I).LT.0)  GO TO 40
COND  C  40 A:  <*  1<=I<=N, J=|<II; II<=I,KV#(II)=II>|  *>
COND  C  40 B:  <* C  41 B *>
COND  C  40 C:  <* C  41 C *>
```

RHO CONSOL 19 = $\langle \emptyset \rangle$

PC CONSOL 19 = $\langle *\ KV(I) < 0\ * \rangle$

C 40 A: Path condition and null intersection

C 40 B,C: Null intersection

```
COND  C  38 A:  <* 1<=I<=N, 1<=K<=NJ, KV(I)>0, J=IMI(I)  *>
COND  C  38 B:  <* C  41 B WITH J REPLACED BY J-1,
C                  KF(J,KK)=IM(KV#(KF#(IMI(J),KK))),
C                  KR(J,KK)=IM(KV#(KR#(IMI(J),KK))),
C                   1<=KK<K   *>
COND  C  38 C:  <* C  41 C *>
 38     KF(J,K)  =  KF(I,K)
 37     KR(J,K)  =  KR(I,K)
        K=K+1
        IF  (K.LE.NJ)  GO TO 38
        KL(J)  =  J
        KV(J)=J
COND  C  40 A:  <*  1<=I<=N, J=|<II; II<=I,KV#(II)=II>|   *>
COND  C  40 B:  <* C  41 B *>
COND  C  40 C:  <* C  41 C *>
```

RHO CONSOL 20 = ⟨ KF(J,K), KR(J,K), K, KV(J), KL(J) ⟩

PC CONSOL 20 = ⟨* K > KJ *⟩

C 40 A,C: Null intersection

C 40 B: Assignment path condition and 38 B

```
COND   C   38 A:   <* 1<=I<=N, 1<=K<=NJ, KV(I)>0, J=TMI(I) *>
COND   C   38 B:   <* C  41 B WITH J REPLACED BY J-1,
C                         KF(J,KK)=TM(KV#(KF#(IMI(J),KK))),
C                         KR(J,KK)=TM(KV#(KR#(IMI(J),KK))),
C                          1<=KK<K   *>
COND   C   38 C:   <* C  41 C *>
 38     KF(J,K)  =  KF(I,K)
 37     KR(J,K)  =  KR(I,K)
        K=K+1
        IF  (K.LE.NJ)   GO TO 38
COND   C   38 A:   <* 1<=I<=N, 1<=K<=NJ, KV(I)>0, J=TMI(I) *>
COND   C   38 B:   <* C  41 B WITH J REPLACED BY J-1,
C                         KF(J,KK)=TM(KV#(KF#(IMI(J),KK))),
C                         KR(J,KK)=TM(KV#(KR#(IMI(J),KK))),
C                          1<=KK<K   *>
COND   C   38 C:   <* C  41 C *>
```

RHO CONSOL 21 = ⟨ KF(K,J), KR(K,J), K ⟩

PC CONSOL 21 = ⟨* K <= NJ *⟩

C 38 A: Path condition on K, null intersection for remainder

C 38 B: 38 B and assignment

C 38 C: Null intersection

```
COND  C  40 A:  <*  1<=I<=N,  J=|<II;  II<=I,KV#(II)=II>|   *>
COND  C  40 B:  <* C  41 B *>
COND  C  40 C:  <* C  41 C *>
 40     I=I+1
        IF  (I.LE.N)  GO TO 41
COND  C  40.2:  <*  KF(J,K)=IM(KV#(KF#(IMI(J),K))),
C                   KR(J,K)=IM(KV#(KR#(IMI(J),K))),
C                   KV(J)=KL(J)=J,  1<=K<=NJ,  1<=J<=M   *>
```

RHO CONSOL 22 = ⟨ I ⟩

PC CONSOL 22 = ⟨* I > N *⟩

C 40.2: C 40 and path condition

```
COND  C  40 A:  <*  1<=I<=N, J=|<II; II<=I,KV#(II)=II>|  *>
COND  C  40 B:  <* C  41 B *>
COND  C  40 C:  <* C  41 C *>
 40    I=I+1
       IF  (I.LE.N)  GO TO 41
COND  C  41 A:  <*  1<=I<=N,  J=|<II; II<I,KV#(II)=II>|  *>
COND  C  41 B:  <* KL(JJ)=KV(JJ)=JJ,
C                    KF(JJ,K)=IM(KV#(KF#(IMI(JJ),K))),
C                    KB(JJ,K)=IM(KV#(KB#(IMI(JJ),K))),
C                    1<=JJ<=J,  1<=K<=NJ  *>
COND  C  41 C:  <*  KF(II,K)=IM(KV#(KF#(II,K))),
C                    KB(II,K)=IM(KV#(KB#(II,K))), 1<=K<=NJ,
C                    KV(II)=IM(II),  I<II<=N AND KV#(II)=II;
C                  KV(II)=-IM(KV#(II)), KV#(II)¬=II, I<II<=N  *>
```

RHO CONSOL 23: ⟨ I ⟩

PC CONSOL 23: ⟨* I <= N *⟩

C 41 A: Path condition and increase in I

C 41 B,C: Null intersection

CONSOL CONTROL PATH NUMBER 24

```
COND  C  40.2:  <*  KF(J,K)=IM(KV#(KF#(IMI(J),K))),
C                   KR(J,K)=IM(KV#(KR#(IMI(J),K))),
C                   KV(J)=KL(J)=J,  1<=K<=NJ,  1<=J<=M  *>
          KT = M + 1
          I=KT
COND  C  51 A:  <* C  40.2 *>
COND  C  51 B:  <* M+1<=I<=N *>
COND  C  51 C:  <* KF(II,J)=KR(II,J)=0,  1<=J<=NJ,  KT<=II<I  *>
```

RHO CONSOL 24 = $\langle KT, I \rangle$

PC CONSOL 24 = Ø

C 51 A: Null intersection

C 51 B: Assignment

C 51 C: Vacuous

```
COND  C  51 A:  <* C  40.2 *>
COND  C  51 B:  <* M+1<=I<=N *>
COND  C  51 C:  <* KP(II,J)=KR(II,J)=0, 1<=J<=NJ, KT<=II<I *>
 51   J=1
COND  C  52:  <*  C  51; 1<=J<=NJ;
C                    KP(I,JJ)=KR(I,JJ)=0, 1<=JJ<J   *>
```

RHO CONSOL 25 = ⟨J⟩

PC CONSOL 25 = ∅

C 52: Null intersection, vacuous, assignment, respectively.

```
COND  C  52:  <*  C  51; 1<=J<=NJ;
C                       KF(I,JJ)=KR(I,JJ)=0, 1<=JJ<J  *>
 52     KF(I,J) = 0
        KR(I,J) = 0
 50    J=J+1
       IF (J.LE.NJ)  GO TO 52
       I=I+1
       IF (I.LE.N)  GO TO 51
        N = M
        NVAC = 0
COND  C  9999:  <* K,ID=0,NVAC=0,H(AK)=H(AK#),
C                        KN=IM(KV#(KN#))  *>
```

RHO CONSOL 26 = ⟨ KF(I,J), KR(I,J), I, J, N, NVAC⟩

C 9999: C 52 and path conditions show conditions are
those necessary for Lemma 5 to apply.

```
COND  C  52:  <*  C  51;  1<=J<=NJ;
C                      KF(I,JJ)=KR(I,JJ)=0,  1<=JJ<J   *>
 52      KF(I,J)  =  0
         KR(I,J)  =  0
 50      J=J+1
         IF  (J.LE.NJ)   GO TO 52
         I=I+1
         IF  (I.LE.N)   GO TO 51
COND  C  51 A:  <* C  40.2 *>
COND  C  51 B:  <*  M+1<=I<=N *>
COND  C  51 C:  <*  KF(II,J)=KR(II,J)=0,  1<=J<=NJ,  KT<=II<I  *>
```

RHO CONSOL 27 = ⟨KR(I,J), KF(I,J), I, J⟩

PC CONSOL 27 = ⟨I <= N, J > NJ *⟩

C 51 A: Null intersection

C 51 B: By path condition

C 51 C; Path condition and assignment

```
COND C 52: <* C 51; 1<=J<=NJ;
C                   KF(I,JJ)=KR(I,JJ)=0, 1<=JJ<J  *>
 52   KF(I,J) = 0
      KR(I,J) = 0
 50   J=J+1
      IF (J.LE.NJ) GO TO 52
COND C 52: <* C 51; 1<=J<=NJ;
C                   KF(I,JJ)=KR(I,JJ)=0, 1<=JJ<J  *>
```

RHO CONSOL 28 = ⟨ KF(I,J), KR(I,J), J ⟩

PC CONSOL 28 = ⟨* J <= NJ *⟩

C 52: Assignment and path condition for last part.

Null intersection for first part.

Proof of termination for CONSOL

Q(CONSOL) · ⟨ F 20, F 25, F 30, F 38.2, F 40, F 50⟩

Expressions: e1 = λ, e2 = I, e3 = J, e4 = K.

Proof that ⟨ei; i = 1,4⟩ is a set of controlled expressions:

PATH 1 (i,iii) 20 —> 21 ==> 20

λ is constant, I is increasing and I <= N at 20 by COND C 20.

PATH 2 20 —> 31 —> 30

λ increases, λ is constant at 30.

PATH 3 (i,iv) 20 —> 31 —> 26 ==> 23 ==> 25

λ increases, λ is constant at 25

PATH 4 25 —> 26 ==> 23 ==> 25

λ is constant, I is constant, J increases and J <= NJ at 25 by COND C 25.

PATH 5 25 —> 30

λ, I are constant, J increases and J <= NJ+1 at 30 by COND C 25 and J = J+1

PATH 6 $30 \longrightarrow 31 \longrightarrow 30$

λ is constant, I increases and I <= N at 30 by COND C 30.

PATH 7 (i,iv) $30 \longrightarrow 31 \longrightarrow 26 \Longrightarrow 23 \Longrightarrow 25$

λ is constant, I increases and I <= N at 25 by COND C 25.

PATH 8 $30 \longrightarrow 41 \longrightarrow 40$

λ increases and λ is constant at 40

PATH 9 $30 \longrightarrow 41 \longrightarrow 38.2$

λ increases and λ is constant at 38.2

PATH 10 $38.2 \longrightarrow 38 \longrightarrow 38.2$

λ, I, and J are constant, K increases and K <= NJ at 38.2 by COND C 38.

PATH 11 $38.2 \longrightarrow 40$

λ, I, J are constant, K increases and K <= NJ+1 at 40 by COND C 38 and K = K+1

PATH 12 $40 \longrightarrow 41 \longrightarrow 40$

λ is constant, I increases and I <= N at 40 by COND C 40.

PATH 13   40 — 41 — 38 — 38.2

λ is constant, I increases and I <= N at 38.2 by COND C 38.

PATH 14   40 —> 51 —> 50

λ increases and λ is constant at 50.

PATH 15   50 —> 52 —> 50

λ and I are constant, J increases and J <= NJ at 50 by COND C 52.

PATH 16   50 —> 51 —> 52 —> 50

λ is constant, I increases and I <= N at 50 by COND C 51.

Q.E.D.

5.3. Proof of correctness for subroutine APPLY(IT,IA)

This solution contains the proof that if $S \in$ COND-IN(APPLY) and P = APPLY the $P(S) \in$ COND-OUT(APPLY). Let HH = H(AK) at entry.

$$\text{COND-IN(APPLY} = \langle * \text{ GC(NR); K, ID} = 0,\ 1 <= \text{LA} <= \text{NR}$$
$$1 <= \text{IT} <= \text{N},\ \text{KV(IT)} = \text{IT} * \rangle$$

$$\text{COND-OUT(APPLY)} = \langle * \text{ GC(NR); K, ID} = 0;$$
$$\text{H(AK)} = \langle \text{HH} + \text{W(1,IT) W(L) W(IT,1)} \rangle * \rangle$$

$$\text{or} \qquad \langle * \text{ N} = \text{NMAX+1} * \rangle$$

GC(NR) is consistent for all paths not thru 36.1 by null intersection or by proofs of NOTE, CONSOL. For paths thru 36.1, GC(NR) or N = NMAX+1 is valid by assignment and path condition since $\Delta(\text{GC(NR)}) \cap \rho(\text{APPLY}) = \text{N}$.

In the preconditions for APPLY the notation $M \not\equiv X$ MOD(Y) means that Y divides M-X.

```
APPLY   CONTROL PATH   NUMBER   1

COMMENT   LET HH BE THE INITIAL VALUE OF H(AK)
COND     <*  GC(NR)   *>  PRECONDITION FOR ALL STATEMENTS
C                          EXCEPT 36.1 AND 9999
COND     <*  GC(NR) OR N=NMAX+1   *>  PRECONDITION VALID FOR
C                                      36.1 AND 9999
COND  AP 1 A:  <*  K,ID=0   *>
COND  AP 1 B:  <*  1<=LA<=NR   *>
COND  AP 1 C:  <*  1<=IT<=N,  KV(IT)=IT   *>
 1      L = LA
        LEN = LF(L)
        LNS = LS(L)
        I = IT
        IB = I
        M=LNS
        MX=1
COND  GLOBAL  <*  1<=L<=NR,  1<=LEN<=1000NSX,  1<=LNS<=NSX,
C                 1<=IT<=N   *>
COND  AP 21 A:  <*  K,ID=0   *>
COND  AP 21 B:  <*  1<=MX<=LEN,  1<=M<=LNS,
C                   (M+1)#MX MOD(LNS)   *>
COND  AP 21 C:  <*  1<=I,IB<=N,  KV(I)=I,  KV(IB)=IB=IT   *>
COND  AP 21 D:  <*  CHAIN(I,IT,MX,L)   *>
COND  AP 21 E:  <*  H(AK)=HH   *>
```

RHO APPLY 1 = $\langle$ L, LEN, LNS, I, IB, M, MX $\rangle$

PC APPLY 1 = $\emptyset$

AP 21 A,E: null intersection

AP 21 B: assignment and GC(NR)

AP 21 C: assignment and AP 1 C

AP 21 D: trivial since I = IT and MX = 1.

APPLY CONTROL PATH NUMBER 2

```
COND  GLOBAL  <*  1<=L<=NR,  1<=LEN<=1000NSX,  1<=LNS<=NSX,
C                 1<=IT<=N  *>
COND  AP 21 A:  <*  K,ID=0  *>
COND  AP 21 B:  <*  1<=MX<=LEN,  1<=M<=LNS,
C                   (M+1)#MX MOD(LNS)  *>
COND  AP 21 C:  <*  1<=I,IB<=N,  KV(I)=I,  KV(IB)=IB=IT  *>
COND  AP 21 D:  <*  CHAIN(I,IT,MX,L)  *>
COND  AP 21 E:  <*  H(AK)=HH  *>
 21     M = M + 1
        IF ( M .GT. LNS )  M = 1
        J = LR(M,L)
        IF  (J.GT.0)  GO TO 14
 13     J = -J
        IX = KR(I,J)
         GO TO 15
COND  AP  15 A,C,D,E:  <*  AP  21 A,C,D,E  *>
COND  AP  15 B:  <*  1<=MX<=LEN,  1<=M<=LNS,  M#MX MOD(LNS)  *>
COND  AP 15 F:  <*  IX=0,  (I,LR(M,L),0) < KF'+KR'  *>
C             OP  <*  IX>0,  (I,LR(M,L),KV(IX)) < KF'+KR'  *>
```

RHO APPLY 2 = ⟨M, J, IX⟩

PC APPLY 2 = ⟨* J < 0 *⟩

AP 15 A,C,D,E: null intersection

AP 15 B: AP 21 B and assignments

AP 15 F: IX = KR(I, -LR(M,L)) by assignment.

The condition follows from link definition and K3.

```
COND  GLOBAL  <*  1<=L<=NP,  1<=LEN<=1000NSX,  1<=LNS<=NSX,
C                  1<=IT<=N  *>
COND  AP 21 A:  <*  K,ID=0  *>
COND  AP 21 B:  <*  1<=MX<=LEN,  1<=M<=LNS,
C                   (M+1)#MX MOD(LNS)  *>
COND  AP 21 C:  <*  1<=I,IB<=N,  KV(I)=I,  KV(IB)=IB=IT  *>
COND  AP 21 D:  <*  CHAIN(I,IT,MX,L)  *>
COND  AP 21 E:  <*  H(AK)=HH  *>
 21    M = M + 1
       IF ( M .GT. LNS )  M = 1
       J = LR(M,L)
       IF  (J.GT.0)  GO TO 14
 14    IX = KF(I,J)
COND  AP  15 A,C,D,E:  <*  AP  21 A,C,D,E  *>
COND  AP  15 B:  <*  1<=MX<=LEN,  1<=M<=LNS,  M#MX MOD(LNS)  *>
COND  AP 15 F:  <*  IX=0,  (I,LR(M,L),0) < KF'+KR'  *>
C            OR  <*  IX>0,  (I,LR(M,L),KV(IX)) < KF'+KR'  *>
```

RHO APPLY 3 = $\langle$M, J, IX$\rangle$

PC APPLY 3 = $\langle$* J, GT, 0 *$\rangle$

AP 15 A,C,D,E: null intersection

AP 15 B: AP 21 B and assignments

AP 15 F: IX = KF(I, LR(M,L)) by assignment.

Path condition, link definition and K3 imply condition

APPLY CONTROL PATH NUMBER 4

```
COND  AP  15 A,C,D,E:  <*  AP  21 A,C,D,E  *>
COND  AP  15 B:  <*  1<=MX<=LEN,  1<=M<=LNS,  M#MX MOD(LNS)  *>
COND  AP 15 F:  <*  IX=0,   (I,LR(M,L),0) < KR'+KR'  *>
C             OR  <*  IX>0,   (I,LR(M,L),KV(IX)) < KR'+KR'  *>
 15     IF ( IX .EQ. 0 )  GO TO 30
 20     I = KV(IX)
        MX=MX+1
        IF  (MX.LE.LEN)  GO TO 21
        GO TO 50
COND  AP 50 A:  <*  K,ID=0  *>
COND  AP 50 B:  <*  1<=I,IB<=N,   KV(I)=I,  KV(IB)=IB  *>
COND  AP 50 C:  <*  CHAIN(IB,I,LEN+1,L)  *>
C             OR  <*  CHAIN(IT,I,MX,L),
C                     CHAIN(IB,IT,-(LEN-MX+1),L)  *>
COND  AP 50 D:  <*  H(AK)=HH  *>
```

RHO APPLY 4 = ⟨ I, MX⟩

PC APPLY 4 = ⟨* IX > 0, MX > LEN *⟩

AP 50 A,D: null intersection

AP 50 B: assignment and K2 imply KV(I) = I

null intersection for remainder

AP 50 C: path condition and AP 15 D give first possibility for C

```
COND   AP   15 A,C,D,F:  <*   AP   21 A,C,D,E   *>
COND   AP   15 B:   <*   1<=MX<=LEN,  1<=M<=LNS,  M#MX MOD(LNS)   *>
COND   AP 15 E:   <*   IX=0,   (I,LR(M,L),0) < KF'+KR'   *>
C              OR   <*   IX>0,   (I,LR(M,L),KV(IX)) < KF'+KR'   *>
  15      IF ( IX .EQ. 0 )   GO TO 30
  20      I = KV(IX)
          MX=MX+1
          IF   (MX.LE.LEN)   GO TO 21
COND   GLOBAL   <*   1<=L<=NR,   1<=LEN<=1000NSX,   1<=LNS<=NSX,
C                        1<=IT<=N   *>
COND   AP 21 A:   <*   K,ID=0   *>
COND   AP 21 B:   <*   1<=MX<=LEN,   1<=M<=LNS,
C                          (M+1)#MX MOD(LNS)   *>
COND   AP 21 C:   <*   1<=I,IB<=N,   KV(I)=I,   KV(IB)=IB=IT   *>
COND   AP 21 D:   <*   CHAIN(I,IT,MX,L)   *>
COND   AP 21 E:   <*   H(AK)=HH   *>
```

RHO APPLY 5 = $\langle$ I, MX $\rangle$

PC APPLY 5 = $\langle$* IX > 0, MX <= LEN *$\rangle$

AP 21 A,E: null intersection

AP 21 B: assignment and path condition

AP 21 C: assignment and K2, ID = 0 imply KV(I) = I

null intersection for remainder

AP 21 > D: AP 15 D, path condition IX > 0 and assignments.

APPLY CONTROL PATH NUMBER 6

```
COND  AP  15 A,C,D,E:  <*  AP  21 A,C,D,E  *>
COND  AP  15 B:  <*  1<=MX<=LEN, 1<=M<=LNS, M#MX MOD(LNS)  *>
COND  AP 15 F:  <*  IX=0,  (I,LR(M,L),0) < KF'+KR'  *>
C            OR  <*  IX>0,  (I,LR(M,L),KV(IX)) < KF'+KR'  *>
 15     IF ( IX .EQ. 0 )  GO TO 30
 30     M=1
        MY=MX
COMMENT  LET MZ = MX+LEN-MY
COND  AP 41 A,E:  <*  AP  21 A,E  *>
COND  AP 41 B:  <*  1<=MX<=MY<=LEN,  1<=M<=LNS,
C                   (M-1)#MZ  MOD(LNS)  *>
COND  AP 41 C:  <*  1<=I,IB<=N,  KV(I)=I,  KV(IB)=IB,
C                   KV(IT)=IT  *>
COND  AP 41 D:  <*  CHAIN(IT,I,MX,L),
C                   CHAIN(IB,IT,-(MY-MX),L))  *>
```

RHO APPLY 6 = ⟨M,MY⟩

PC APPLY 6 = ⟨* IX = 0 *⟩

AP 41 A,C,E: null intersection

AP 41 B: assignment, AP 15 B, and GC(NR)

AP 41 D: null intersection for first chain,

second chain trivial since IB = IT and MY = MX.

```
        COMMENT   LET MZ = MX+LEN-MY
        COND   AP 41 A,E:   <*   AP  21 A,E   *>
        COND   AP 41 B:   <*   1<=MX<=MY<=LEN,   1<=M<=LNS,
        C                      (M-1)#MZ   MOD(LNS)   *>
        COND   AP 41 C:   <*   1<=I,IB<=N,   KV(I)=I, KV(IB)=IB,
        C                      KV(IT)=IT   *>
        COND   AP 41 D:   <*   CHAIN(IT,I,MY,L),
        C                      CHAIN(IB,IT,-(MY-MY),L))   *>
         41    M = M - 1
               IF ( M .EQ. 0 )   M = LNS
               J = LP(M,L)
               IF  (J.GT.0)   GO TO 34
         33    JT = -J
               IX = KF(IB,JT)
               GO TO 35
        COND   AP  35 A,C,D,E:  <*   AP   41 A,C,D,E   *>
        COND   AP 35 B:   <*   1<=MX<=MY<=LEN,   1<=M<=LNS,
        C                      M#MZ   MOD(LNS),   J=LP(M,L)   *>
        COND   AP 35 F:   <*   IX=0,   (0,J,IB) < KF'+KR'   *>
        C             OR  <*   IX>0,   (KV(IX),J,IB) < KF'+KR'   *>
```

RHO APPLY 7 = ⟨ M,J,IX ⟩

PC APPLY 7 = ⟨* J < 0 *⟩

AP 35 A,C,D,E: null intersection

AP 35 B: null intersection for first inequality,
assignment and AP 41 B for remainder.

AP 35 F: path condition, assignment and definition of link.

```
COMMENT  LET M7 = MX+LEN-MY
COND  AP 41 A,E:  <*  AP  21 A,E  *>
COND  AP 41 B:  <*  1<=MX<=MY<=LEN,   1<=M<=LNS,
C                   (M-1)#M7  MOD(LNS)  *>
COND  AP 41 C:  <*  1<=I,IB<=N,  KV(I)=I,  KV(IB)=IB,
C                   KV(IT)=IT  *>
COND  AP 41 D:  <*  CHAIN(IT,I,MY,L),
C                   CHAIN(IB,IT,-(MY-MX),L))  *>
 41     M = M - 1
        IF ( M .EQ. 0 )  M = LNS
        J = LB(M,L)
        IF  (J.GT.0)  GO TO 34
 34     IX = KB(IB,J)
COND  AP  35 A,C,D,E:  <*  AP  41 A,C,D,F  *>
COND  AP 35 B:  <*  1<=MX<=MY<=LEN,   1<=M<=LNS,
C                   M#M7  MOD(LNS),  J=LB(M,L)  *>
COND  AP 35 F:  <*  IX=0,   (0,J,IB) < KF'+KP'  *>
C           OR  <*  IX>0,   (KV(IX),J,IB) < KF'+KP'  *>
```

RHO APPLY 8 = ⟨M,J,IX⟩

PC AP 8 = ⟨* J > 0 *⟩

Proof same as for Path 7.

```
COND  AP  35 A,C,D,E:  <*  AP  41 A,C,D,E  *>
COND  AP 35 B:  <*  1<=MX<=MY<=LEN,  1<=M<=LNS,
C                   M#MZ  MOD(LNS),  J=LR(M,L)  *>
COND  AP 35 F:  <*  IX=0,  (0,J,IB) < KF'+KR'  *>
C          OR  <*  IX>0,  (KV(IX),J,IB) < KF'+KR'  *>
 35    IF ( IX .NE. 0 )  GO TO 40
       IF ( MY .LT. LEN )  GO TO 36
       IF ( J .GT. 0 )  GO TO 355
       KF(IB,JT) = I
       CALL NOTE( IB,KR(I,JT) )
       GO TO 60
COND  AP 60 A:  <*  K  *>
COND  AP 60 B:  <*  H(AK)=<HH+W(1,IT) W(L) W(IT,1)>  *>
```

RHO APPLY 9 $= \langle$ KF(IB,JT), ρ(NOTE) $\rangle$

PC APPLY 9 $= \langle *$ IX $= 0$, J $< 0 * \rangle$

AP 60 A: NOTE preserves the condition K.
Since AP 35 satisfies initial conditions of NOTE following NOTE EC(KR(I,JT)) = IB thus KF(IB,JT) = I satisfies K3.

AP 60 B: Corollary 3 to Lemma 4

```
COND  AP  35 A,C,D,F:  <*  AP  41 A,C,D,E  *>
COND  AP 35 B:  <*  1<=MX<=MY<=LEN,  1<=M<=LNS,
C                   M#MZ  MOD(LNS),  J=LR(M,L)  *>
COND  AP 35 F:  <*  IX=0,  (0,J,IB) < KF'+KR'  *>
C           OR  <*  IX>0,  (KV(IX),J,IB) < KF'+KR'  *>
 35    IF ( IX .NE. 0 )  GO TO 40
       IF ( MY .LT. LEN )  GO TO 36
       IF ( J .GT. 0 )  GO TO 355
 355   KR(IB,J) = I
       CALL NOTE( IB,KF(I,J) )
       GO TO 60
COND  AP 60 A:  <*  K  *>
COND  AP 60 B:  <*  H(AK)=<HH+W(1,IT)W(L)W(IT,1)>  *>
```

PATH 10: The same as PATH 9 with KR, KF interchanged.

APPLY CONTROL PATH NUMBER 11

```
COND  AP  35 A,C,D,F:  <*  AP  41 A,C,D,E  *>
COND  AP 35 B:  <*  1<=MX<=MY<=LEN,  1<=M<=LNS,
C                   M#MZ  MOD(LNS),  J=LR(M,I)  *>
COND  AP 35 E:  <*  IY=0,  (0,J,IB) < KF'+KR'  *>
C           OR  <*  IX>0,  (KV(IY),J,IB) < KF'+KR'  *>
 35    IF ( IY .NE. 0 )  GO TO 40
       IF ( MY .LT. LEN )  GO TO 36
 36    N = N + 1
       IF ( N .GT. NMAX )  GO TO 9999
       KL(N) = N
       KV(N) = N
       IX = N
       IF ( J .GT. 0 )  GO TO 37
       KF(IB,JT) = IX
       KR(IX,JT) = IB
       GO TO 40
COND  AP 40 :  <*  AP 35 , IX>0  *>
```

RHO APPLY 11 = $\langle$N, KL(N), KV(N), IX, KR(IB,JT), KF(IX,JT)$\rangle$

PC APPLY 11 = $\langle$* IX = 0, MY < LEN, N <= NMAX, J < 0 *$\rangle$

AP 40: Path condition and Lemma 2

```
APPLY   CONTROL PATH  NUMBER  12

COND  AP  35 A,C,D,E:  <*  AP  41 A,C,D,E  *>
COND  AP 35 B:  <*  1<=MX<=MY<=LEN,  1<=M<=LNS,
C                    M#MZ  MOD(LNS),  J=LR(M,L)  *>
COND  AP 35 F:  <*  IX=0,  (0,J,IB) < KF'+KR'  *>
C            OR  <*  IX>0,  (KV(IX),J,IB) < KF'+KR'  *>
 35     IF ( IX .NE. 0 )  GO TO 40
        IF ( MY .LT. LEN )  GO TO 36
 36     N = N + 1
        IF ( N .GT. NMAX )  GO TO 9999
        KL(N) = N
        KV(N) = N
        IX = N
        IF ( J .GT. 0 )  GO TO 37
 37     KR(IB,J) = IX
        KF(IX,J) = IB
COND  AP 40 :  <*  AP 35 , IX>0  *>
```

PATH 12: analogous to Path 11.

```
COND  AP 35 A,C,D,E:  <*  AP  41 A,C,D,E  *>
COND  AP 35 B:  <*  1<=MX<=MY<=LEN,  1<=M<=LNS,
C                   M#MZ  MOD(LNS),  J=LR(M,L)  *>
COND  AP 35 F:  <*  IX=0,  (0,J,IB) < KF'+KB'  *>
C             OR  <*  IX>0,  (KV(IX),J,IB) < KF'+KB'  *>
 35     IF ( IX .NE. 0 )  GO TO 40
        IF ( MY .LT. LEN )  GO TO 36
 36     N = N + 1
        IF ( N .GT. NMAX )  GO TO 9999
COND  AP 9999 A:  <*  K,ID=0  *>
COND  AP 9999 B:  <*  GC(NB)  *>
COND  AP 9999 C:  <*  H(AK)=<HH+W(1,IT)W(L)W(IT,1)>  *>
C   OR
COND  AP 9999 D:  <*  N=NMAX+1  *>
```

RHO APPLY 13 = $\langle N \rangle$

PC APPLY 13 = $\langle *\ N > NMAX,\ IX > 0,\ MY < LEN\ * \rangle$

AP 9999 D: by path condition

```
      COND  AP  35 A,C,D,E:  <*  AP  41 A,C,D,E  *>
      COND  AP 35 B:  <*  1<=MX<=MY<=LEN,  1<=M<=LNS,
C                         M#MZ  MOD(LNS),  J=LR(M,L)  *>
      COND  AP 35 F:  <*  IX=0,  (0,J,IB) < KB'+KB'  *>
C                 OR  <*  IX>0,  (KV(IX),J,IB) < KB'+KB'  *>
 35     IF ( IX .NE. 0 )  GO TO 40
      COND  AP 40 :  <*  AP 35 , IX>0  *>
```

RHO APPLY 14 = Ø

PC APPLY 14 = <* IX > 0 *>

AP 40: path conditon and null intersection

```
      COND  AP 40 :  <*  AP 35 , IX>0  *>
       40    IB = KV(IX)
             MY=MY+1
             IF  (MY.LE.LEN)  GO TO 41
      COND  AP 50 A:  <*  K,ID=0  *>
      COND  AP 50 B:  <*  1<=I,IB<=N,  KV(I)=T,  KV(IB)=TB  *>
      COND  AP 50 C:  <*  CHAIN(IB,I,LEN+1,L)  *>
      C          OR   <*  CHAIN(IT,I,MY,L),
      C                   CHAIN(IB,IT,-(LEN-MY+1),L)  *>
      COND  AP 50 D:  <*  H(AK)=HH  *>
```

RHO APPLY 15 = ⟨ IB, MY ⟩

PC APPLY 15 = ⟨* MY > LEN *⟩

AP 50 A,B,D: null intersection

AP 50 C: AP 35 D, F, assignment, and MY = LEN+1 by path condition

APPLY CONTROL PATH NUMBER 16

```
COND  AP 40 :  <*  AP 35 , IX>0  *>
 40    IB = KV(IX)
       MY=MY+1
       IF  (MY.LE.LEN)  GO TO 41
COMMENT  LET MZ = MX+LEN-MY
COND  AP 41 A,E:  <*  AP  21 A,E  *>
COND  AP 41 B:  <*  1<=MX<=MY<=LEN,  1<=M<=LNS,
C                   (M-1)*MZ  MOD(LNS)  *>
COND  AP 41 C:  <*  1<=I,IB<=N,  KV(I)=I,  KV(IB)=IB,
C                   KV(IT)=IT  *>
COND  AP 41 D:  <*  CHAIN(IT,I,MX,L),
C                   CHAIN(IB,IT,-(MY-MX),L))  *>
```

RHO APPLY 16 = ⟨ IB, MY ⟩

PC APPLY 16 = ⟨* MY <= LEN *⟩

AP 41 A,C,G: null intersection

AP 41 B: assignment, path condition

AP 41 D: AP 35 D, F and assignment

```
COND  AP 50 A:  <*  K,TD=0  *>
COND  AP 50 B:  <*  1<=I,IB<=N,  KV(I)=T,  KV(IB)=TB  *>
COND  AP 50 C:  <*  CHAIN(IB,I,LEN+1,L)  *>
C           OR  <*  CHAIN(IT,I,MY,L),
C                   CHAIN(IB,IT,-(LEN-MY+1),L)  *>
COND  AP 50 D:  <*  H(AK)=HH  *>
 50    CALL NOTE(I,IB)
COND  AP 60 A:  <*  K  *>
COND  AP 60 B:  <*  H(AK)=<HH+W(1,IT)W(L)W(IT,1)>  *>
```

RHO APPLY 17 = $\langle \rho(\text{NOTE}) \rangle$

PC APPLY 17 = $\emptyset$

AP 60 A: AP 50 satisfies initial conditions for note, thus K is preserved.

AP 60 B: Corollary 2 to Lemma 4.

```
      COND  AP 60 A:  <*   K   *>
      COND  AP 60 B:  <*   H(AK)=<HH+W(1,IT)W(L)W(IT,1)>   *>
 60       IF ( ID .EQ. 0 )   GO TO 9999
          NVAC = NVAC + 1
          I = ID
          ID = KL(I)
          IA = I
          KL(I)=I
          J=1
      COND  AP 64 A:  <*  K0,K1,K2,K3,  INCLUDING I IN THE SET OF
      C                   ROWS LABELED EC(I), MCOS+NVAC+P(0)=N+2  *>
      COND  AP 64 B:  <*  AP  60 B  *>
      COND  AP 64 C:  <*  IA BELONGS TO EC CHAIN WITH HEAD I,
      C                   TAIL EC(I)  *>
      COND  AP 64 D:  <* 1<=J<=NJ  *>
```

RHO APPLY 18 = $\langle$ I, ID, IA, KL(I), J $\rangle$

PC APPLY 18 = $\langle$* ID $>$ 0 *$\rangle$

AP 64 A: I is separated from C(0) and made a single vertex cycle by assignments $\langle$* K-K1* $\rangle$ is still satisfied by considering I as part of C(0). NVAC is increased giving last equation from K4.

AP 64 B: redefinition of K does not change any links.

AP 64 C: assignment

```
COND  AP 60 A:  <*  K  *>
COND  AP 60 B:  <*  H(AK)=<HH+W(1,IT)W(L)W(IT,1)>  *>
 60    IF ( ID .EQ. 0 )  GO TO 9999
COND  AP 9999 A:  <*  K,ID=0  *>
COND  AP 9999 B:  <*  GC(NP)  *>
COND  AP 9999 C:  <*  H(AK)=<HH+W(1,IT)W(L)W(IT,1)>  *>
C   OR
COND  AP 9999 D:  <*  N=NMAX+1  *>
```

RHO APPLY 19 = $\langle \emptyset \rangle$

PC APPLY 19 = $\langle * \; ID = 0 \; * \rangle$

AP 9999 A: null intersection and path condition

AP 9999 B: global on apply-paths thru 3.6.1

AP 9999 C: null intersection

```
COND  AP 64 A:  <*  K0,K1,K2,K3,  INCLUDING I IN THE SET OF
C                    ROWS LABELED EC(I), MCOS+NVAC+P(0)=N+2   *>
COND  AP 64 B:  <*  AP  60 B  *>
COND  AP 64 C:  <*  IA BELONGS TO EC CHAIN WITH HEAD I,
C                      TAIL EC(I)  *>
COND  AP 64 D:  <* 1<=J<=NJ  *>
 64    IF ( KV(IA) .GE. 0 )  GO TO 65
       IA = -KV(IA)
       GO TO 64
COND  AP 64 A:  <*  K0,K1,K2,K3,  INCLUDING I IN THE SET OF
C                    ROWS LABELED EC(I), MCOS+NVAC+P(0)=N+2   *>
COND  AP 64 B:  <*  AP  60 B  *>
COND  AP 64 C:  <*  IA BELONGS TO EC CHAIN WITH HEAD I,
C                      TAIL EC(I)  *>
COND  AP 64 D:  <* 1<=J<=NJ  *>
```

RHO APPLY 20 = ⟨ IA ⟩

PC APPLY 20 = ⟨* KV(IA) < 0 *⟩

AP 64 A,C: null intersection

AP 64 B: definition of EC chain path condition and assignment

```
COND   AP 64 A:  <*  K0,K1,K2,K3,  INCLUDING I IN THE SET OF
C                     ROWS LABELED EC(I), KCOS+NVAC+P(0)=N+2   *>
COND   AP 64 B:  <*  AP  60 B  *>
COND   AP 64 C:  <*  IA BELONGS TO EC CHAIN WITH HEAD I,
C                     TAIL EC(I)  *>
COND   AP 64 D:  <* 1<=J<=NJ  *>
 64     IF ( KV(IA) .GE. 0 )  GO TO 65
 65     IA = KV(IA)
        KV(I) = IA
        KL(I) = KL(IA)
        KL(IA) = I
COND   AP  71 A:  <*  K INCLUDING I IN THE SET OF ROWS
C                     LABELED IA,  EC(I)=IA  *>
COND   AP  71 B:  <*  AP  60 B  *>
COND   AP 71 C:  <*  1<=J<=NJ  *>
COND   AP 71 D:  <* (KP(I,L)=0 OR EC(KP(IA,L))=EC(KP(I,L)))
C                    AND (KP(I,L)=0 OR EC(KP(IA,L))=EC(KP(I,L)))
C                     FOR 1<=L<J  *>
```

RHO APPLY 21 = ⟨ IA, KV(I), KL(I), KL(IA) ⟩

PC APPLY 21 = ⟨* KV(IA) > 0 *⟩

AP 71 A: IA = EC(IA) by path condition, assignment and definition of EC chain. Remaining assignments move the point cycle I into component with IA as representation restoring condition K4. By including I in rows labeled IA $\Delta(K3)$ is unchanged.

AP 71 B: from 71 A $\Delta(K3)$ unchanged means $\mathscr{L}(AK)$ unchanged.

AP 71 C: null intersection

AP 71 D: vacuous

APPLY CONTROL PATH NUMBER 22

```
COND  AP  71 A:  <*  K INCLUDING I IN THE SET OF ROWS
C                     LABELED IA,  EC(I)=IA  *>
COND  AP  71 B:  <*  AP  60 B  *>
COND  AP 71 C:  <*  1<=J<=NJ  *>
COND  AP 71 D:  <* (KF(I,L)=0 OR EC(KF(IA,L))=EC(KF(I,L)))
C                AND (KR(I,L)=0 OR EC(KR(IA,L))=EC(KR(I,L)))
C                     FOR 1<=L<J  *>
 71    CALL NOTE ( KF(I,J) , KF(IA,J) )
       CALL NOTE ( KR(I,J) , KR(IA,J) )
 70    J=J+1
       IF  (J.LE.NJ)  GO TO 71
       GO TO 60
COND  AP 60 A:  <*  K  *>
COND  AP 60 B:  <*  H(AK)=<HH+W(1,IT)W(L)W(IT,1)>  *>
```

RHO APPLY 22 = ⟨J, ρ(NOTE(KF(I,J), KF(IA,J))),
ρ(NOTE(KR(I,J), KR(IA,J))⟩

PC APPLY 22 = ⟨* J > NJ *⟩

AP 60 A,B: NOTE preserves K
Path condition says AP 71 D holds for 1 <= L <= NJ and therefore all links defined by row I of KR' + KF' are in the set of links defined by row IA thus deleting row I leaves H(AK) fixed

APPLY CONTROL PATH NUMBER 23

```
COND  AP  71 A:  <*  K INCLUDING I IN THE SET OF ROWS
C                      LABELED IA,  EC(I)=IA  *>
COND  AP  71 B:  <*  AP  60 B  *>
COND  AP 71 C:  <*  1<=J<=NJ  *>
COND  AP 71 D:  <* (KF(I,L)=0 OR EC(KF(IA,L))=EC(KF(I,L)))
C                AND (KR(I,L)=0 OR EC(KR(IA,L))=EC(KR(I,L)))
C                      FOR 1<=L<J  *>
 71     CALL NOTE ( KF(I,J) , KF(IA,J) )
        CALL NOTE ( KR(I,J) , KR(IA,J) )
 70     J=J+1
        IF  (J.LE.NJ)  GO TO 71
COND  AP  71 A:  <*  K INCLUDING I IN THE SET OF ROWS
C                      LABELED IA,  EC(I)=IA  *>
COND  AP  71 B:  <*  AP  60 B  *>
COND  AP 71 C:  <*  1<=J<=NJ  *>
COND  AP 71 D:  <* (KF(I,L)=0 OR EC(KF(IA,L))=EC(KF(I,L)))
C                AND (KR(I,L)=0 OR EC(KR(IA,L))=EC(KR(I,L)))
C                      FOR 1<=L<J  *>
```

RHO APPLY 23 = $\langle$ J, o(NOTE(KF(I,J), KF(IA,J)),
o(NOTE(KR(I,J), KR(IA,J)) $\rangle$

PC APPLY 23 = $\langle$* J $\Leftarrow$ NJ *$\rangle$

AP 71 A: K preserved by NOTE

AP 71 B: effect of NOTE is described by either the function of Lemma 3, or Corollary 1 to Lemma 4, thus H(AK) is unchanged.

AP 71 C: path condition

AP 71 D: property of NOTE

Proof of termination for APPLY.

Q(APPLY) = ⟨ F 20.2, F 40.2, F 64, F 70.1⟩

EXPRESSIONS: e1 = λ, e2 = MX, e3 = MY, e4 = NVAC, e5 = - IA, e6 = J.

Proof that ⟨ei; i = 1,6⟩ is a controlled set of expressions:

PATH 1 (i,ii) = 20.2 —> 21 ==> 15 —> 20.2

λ is constant along the paths. MX is increasing along the paths and MX <= LEN+1 at 20.2 by COND AP 15.

PATH 2 (i,xii) = 20.2 —> 21 ==> 15 —> 41 == 35 ==> 36 —> 40 —> 40.2

λ increases along each path, λ is constant at 40.2.

50

PATH 3 (i,ix) = 20.2 —> 21 ==> 15 —> 41 ==> 35 —> 35.2 ==> 60 —> 64

λ increases along each path, λ is constant at 64

PATH 4(i,vi) = 40.2 —> 41 ==> 35 —> 36 ==> 40 —> 40.2

λ and MX are constant along paths, MY increases and MY <= LEN+1 at 40.2 by COND AP 40.

50

PATH 5 (i,v) = 40.2 —> 41 ==> 35 —> 35.2 ==> 60 —> 64

λ increases along paths, is constant at 64.

PATH 6 = 64 —> 64

λ, MX, MY and NVAC are constant, -IA increases and -IA < 0 at 64 by COND AP 64 and definition of EC chain.

PATH 7 = 64 —> 70.1

λ, MX, MY and NUAC are constant, -IA is nondecreasing -IA < 0 by COND AP 64, J is increasing, J <= NJ+1 at 70.1 by COND AP 71.

PATH 8 = 70.1 —> 70.1

λ, MX, MY, NVAC and -IA are constant, J increases and J <= NJ+1 at 70.1 by COND AP 71.

PATH 9 = 70.1 —> 60 —> 64

λ, MK, MY are constant and NVAC increases. By COND AP 60, NVAC = N+1 - MCOS - P(0), by K MCOS >= 1 and P(0) >= 1 therefore NVAC < N. N is constant in D3 when bound on NVAC is necessary.

Q.E.D.

5.4. Proof of correctness for main program COSET

This section contains proof that if $S \in$ COND-IN(COSET) and if P = COSET then P(S) $\in$ COND-OUT(COSET).

COND-IN(COSET) = $\langle$* INPUT *$\rangle$

COND-OUT(COSET) = $\langle$* N = $[\mathfrak{F} : \mathfrak{U}]$ or N = MAX+1 *$\rangle$

INPUT is a global condition since σ(COSET) $\cap$ Δ(INPUT) = $\emptyset$

COSET CONTROL PATH NUMBER 1

```
COND  GLOBAL  <*  INPUT  *>
 1      NMAX = 500
        NJX = 5
        NRX = 10
        NSX = 20
 10     NVAC = 0
        ID = 0
        NJ = 1
        I=1
COND  M  16 A:  <* 1<=NJ<=NJX, NVAC=0, ID=0 *>
COND  M  16 B:  <* 1<=I<=NMAX *>
COND  M  16 C:  <*  (KF(II,J)=KR(II,J)=0, 1<=J<=NJX),
C                   KV(II)=KL(II)=0,  1<=II<I  *>
```

RHO COSET 1 = $\langle$NMAX, NJX, NRX, NSX, NVAC, ID, NJ, I$\rangle$

PC COSET 1 = $\emptyset$

M 16 A,B: Assignment

M 16 C: Vacuous since I = 1

```
      COND  M  16 A:  <* 1<=NJ<=NJX, NVAC=0, ID=0 *>
      COND  M  16 B:  <* 1<=I<=NMAX *>
      COND  M  16 C:  <* (KF(II,J)=KR(II,J)=0, 1<=J<=NJX),
C                         KV(II)=KL(II)=0,  1<=II<I  *>
   16    KL(I) = 0
         KV(I) = 0
         J=1
      COND  M  17 A:  <* M  16 *>
      COND  M  17 B:  <* 1<=J<=NJX *>
      COND  M  17 C:  <* KF(I,JJ)=KR(I,JJ)=0, 1<=JJ<J,
C                        KV(I)=KL(I)=0  *>
```

RHO COSET 2 = $\langle$ KL(I), KV(I), J $\rangle$

PC COSET 2 = $\emptyset$

M 17 A: Null intersection

M 17 B: Assignment

M 17 C: First part vacuous since J = 1 second part due to assignment

```
COND  M  17 A:  <* M  16 *>
COND  M  17 B:  <* 1<=J<=NJX *>
COND  M  17 C:  <* KF(I,JJ)=KR(I,JJ)=0, 1<=JJ<J,
C                         KV(I)=KL(I)=0  *>
 17      KF(I,J) = 0
         KR(I,J) = 0
         J=J+1
         IF  (J.LE.NJX)  GO TO 17
         I=I+1
         IF  (I.LE.NMAX)  GO TO 16
         N = 1
         KL(1) = 1
         KV(1) = 1
         NR = 1
         F = .TRUE.
         SW = .FALSE.
         KN = 0
         ICD=0
COND  M  20 A:  <* GC(NR-1), KN=0 *>
COND  M  20 B:  <* F, ID=0 *>
COND  M  20 D:  <* ONE OF THE SET DI,DII,DIII *>
C            DI:  <*  0<=ICD<=IZ, F=.T., SW=.F., NR=1  *>
C           DII:  <* IZ<ICD<=IG, F=.F., SW=.F., NR=1 *>
C          DIII:  <* IG<ICD<=IR, F=.F., SW=.T., NR=ICD-IG+1 *>
COND  M  20 F:  <* <H(I), 0<=I<=MIN(ICD-IZ,IG-IZ)> < H(AK)  *>
```

RHO COSET 3 = $\langle$ KF(I,J), KR(I,J), J, I, N, NR,
KL(1), KV(1), KN, ICD, F, SW $\rangle$

PC COSET 3 = $\langle$* J $>$ NJX, I $>$ NMAX *$\rangle$

M 20 A: NR = 1 Condition GC(0) is satisfied by assignments

M 20 B: KO true by 17, path condition and assignments.
K1 true since N = 1, and KL(1) = 1 give two component graph each with a single member.
K2 true since KV(1) = 1. K3 true since KR, KF = 0. K4 true since MCOS = 1, NVAC = 0, P(0) = 1, N = 1.

M 20 C: Assignment

M 20 D = DI: by assignment

M 20 E: vacuous

COSET CONTROL PATH NUMBER 4

```
COND  M  17 A:  <* M  16 *>
COND  M  17 B:  <* 1<=J<=NJX *>
COND  M  17 C:  <* KF(I,JJ)=KR(I,JJ)=0, 1<=JJ<J,
C                    KV(I)=KL(I)=0   *>
 17     KF(I,J)  = 0
        KR(I,J)  = 0
        J=J+1
        IF  (J.LE.NJX)  GO TO 17
        I=I+1
        IF  (I.LE.NMAX)  GO TO 16
COND  M  16 A:  <* 1<=NJ<=NJX, NVAC=0, ID=0 *>
COND  M  16 B:  <* 1<=I<=NMAX *>
COND  M  16 C:  <* (KF(II,J)=KR(II,J)=0, 1<=J<=NJX),
C                    KV(II)=KL(II)=0,  1<=II<I   *>
```

RHO COSET $4 = \langle KF(I,J), KR(I,J), J, I \rangle$

PC COSET $4 = \langle J < NJX, I \leq NMAX \rangle$

M 16 A: Null intersection

M 16 B,C: Path condition and assignment

```
      COND  M  17 A:  <* M  16 *>
      COND  M  17 B:  <* 1<=J<=NJX *>
      COND  M  17 C:  <* KF(I,JJ)=KR(I,JJ)=0, 1<=JJ<J,
      C                    KV(I)=KL(I)=0  *>
   17     KF(I,J) = 0
          KR(I,J) = 0
          J=J+1
          IF  (J.LE.NJX)  GO TO 17
      COND  M  17 A:  <* M  16 *>
      COND  M  17 B:  <* 1<=J<=NJX *>
      COND  M  17 C:  <* KF(I,JJ)=KR(I,JJ)=0, 1<=JJ<J,
      C                    KV(I)=KL(I)=0  *>
```

RHO COSET 5 = ⟨KF(I,J), KR(I,J), J⟩

PC COSET 5 = ⟨* J <= NJX *⟩

M 17 A: Null intersection

M 17 B: Path condition

M 17 C: Assignment

```
COND  M  20 A:  <* GC(NR-1), KN=0 *>
COND  M  20 B:  <* K, ID=0 *>
COND  M  20 D:  <* ONE OF THE SET DI,DII,DIII *>
C          DI:  <*  0<=ICD<=IZ, F=.T., SW=.F., NR=1  *>
C         DII:  <* IZ<ICD<=IG, F=.F., SW=.F., NR=1 *>
C        DIII:  <* IG<ICD<=IR, F=.F., SW=.T., NR=ICD-IG+1 *>
COND  M  20 E:  <* <U(I), 0<=I<=MIN(ICD-IZ,IG-IZ)> < H(AK) *>
 20    ICD=ICD+1
       READ(5,201) NCD,MULT,(LR(I,NR),I=1,NSX)
COND  M  22 A:  <* M  20 A *>
COND  M  22 B:  <* M  20 B *>
COND  M  22 D:  <* ONE OF THE SET DI,DII,DIII *>
C          DI:  <* 1<=ICD<=IZ+1, F=.T., SW=.F., NR=1 *>
C         DII:  <* IZ+1<ICD<=IG+1, F=.F., SW=.F., NR=1 *>
C        DIII:  <* IG+1<ICD<=IR+1, F=.F., SW=.T., NR=ICD-IG *>
COND  M  22 E:  <* <U(I),0<=I<=MIN(ICD-IZ-1,IG-IZ)> < H(AK) *>
COND  M  22 F:  <* NCD=INPUT(1,ICD), MULT=INPUT(2,ICD)  *>
COND  M  22 G:  <* LR(I,NR)=INPUT(I+2,ICD), 1<=I<=NSX *>
```

RHO COSET 6 = $\langle$ ICD, NCD, MULT, (LR(I,NR) I = 1, NSX) $\rangle$

PC COSET 6 = $\emptyset$

M 22 A,B: Null intersection

M 22 D,E: Increase of ICD by 1

M 22 F,G: Assignment

```
COND  M  22 A:  <* M  20 A *>
COND  M  22 B:  <* M  20 B *>
COND  M  22 D:  <* ONE OF THE SET DI,DII,DIII *>
C            DI:  <* 1<=ICD<=IZ+1, F=.T., SW=.F., NR=1 *>
C           DII:  <* IZ+1<ICD<=IG+1, F=.F., SW=.F., NR=1 *>
C          DIII:  <* IG+1<ICD<=IR+1, F=.F., SW=.T., NR=ICD-IG *>
COND  M  22 E:  <* <U(I),0<=I<=MIN(ICD-IZ-1,IG-IZ)> < H(AK) *>
COND  M  22 F:  <* NCD=INPUT(1,ICD), MULT=INPUT(2,ICD)  *>
COND  M  22 G:  <* LR(I,NR)=INPUT(I+2,ICD), 1<=I<=NSX *>
 22    IF ( NCD .EQ. 0 )  GO TO 60
       IF (F)  WRITE(6,240)
       F = .FALSE.
       I=1
COND  M  26 A:  <* M  20 A *>
COND  M  26 B:  <* M  20 B *>
COND  M  26 C:  <* 1<=I<=NSX  *>
COND  M  26 D:  <* ONE OF THE SET DII,DIII *>
C           DII:  <* IZ<ICD<=IG+1, F=.F., SW=.F., NR=1 *>
C          DIII:  <* IG+1<ICD<=IR, F=.F., SW=.T., NR=ICD-IG *>
COND  M  26 E:  <* M  22 F *>
COND  M  26 F:  <* M  22 F *>
COND  M  26 G:  <* M  22 G, 1<=|LR(II,NR)|<=NJ, 1<=II<I *>
```

RHO COSET 7 = $\langle F,I \rangle$

PC COSET 7 = $\langle$* NCD $\neq$ 0 *$\rangle$

M 26 A,B,E,F,G: Null intersection

M 26 C: Assignment

M 26 D: NCD $\neq$ 0 implies IZ < ICD $\leq$ IR

22D and F = .F. implies 26D.

```
COND  M  22 A:  <* M  20 A *>
COND  M  22 B:  <* M  20 B *>
COND  M  22 D:  <* ONE OF THE SET DI,DII,DIII *>
C           DI:  <* 1<=ICD<=IZ+1, F=.T., SW=.F., NR=1 *>
C          DII:  <* IZ+1<ICD<=IG+1, F=.F., SW=.F., NR=1 *>
C         DIII:  <* IG+1<ICD<=IR+1, F=.F., SW=.T., NR=ICD-IG *>
COND  M  22 E:  <* <U(I),0<=I<=MIN(ICD-IZ-1,IG-IZ)> < H(AK) *>
COND  M  22 F:  <* NCD=INPUT(1,ICD), MULT=INPUT(2,ICD)  *>
COND  M  22 G:  <* LR(I,NR)=INPUT(I+2,ICD), 1<=I<=NSX *>
 22    IF ( NCD .EQ. 0 )  GO TO 60
COND  M  60 A:  <* M  20 A *>
COND  M  60 B:  <* M  20 B *>
COND  M  60 C:  <* M  22 F *>
COND  M  60 D:  <* ONE OF THE SET DI, DII *>
C           DI:  <* 1<=ICD<=IZ, NR=1, F=.T., SW=.F.  *>
C          DII:  <* ICD=IR+1, F=.F., NR=IR-IG+1 *>
```

RHO COSET 8 = $\langle \emptyset \rangle$

PC COSET 8 = $\langle$* NCD = 0 *$\rangle$

M 60 A,B,C: Null intersection

M 60 D: NCD = 0 implies 1 $\Leftarrow$ ICD $\Leftarrow$ IZ or

ICD = IR+1

```
      COND  M  26 A:  <* M  20 A *>
      COND  M  26 B:  <* M  20 B *>
      COND  M  26 C:  <* 1<=I<=NSX  *>
      COND  M  26 D:  <* ONE OF THE SET DII,DIII *>
      C          DII:  <* IZ<ICD<=IG+1, F=.F., SW=.F., NR=1 *>
      C         DIII:  <* IG+1<ICD<=IR, F=.F., SW=.T., NR=ICD-IG *>
      COND  M  26 E:  <* M  22 E *>
      COND  M  26 F:  <* M  22 F *>
      COND  M  26 G:  <* M  22 G, 1<=|LR(II,NR)|<=NJ, 1<=II<I *>
       26    IF ( LR(I,NR) .EQ. 0 )  GO TO 30
             NJ = MAX0( NJ, IABS( LR(I,NR) )  )
             I=I+1
             IF  (I.LE.NSX)  GO TO 26
             I = NSX + 1
      COND  M  30 A:  <* M  26 A,B,D,E,F,G *>
      COND  M  30 B:  <* 1<I<=NSX+1 *>
```

RHO COSET 9 = $\langle$NJ, I$\rangle$

PC COSET 9 = $\langle$* LR(I,NR) $\neq$ 0, I > NSX *$\rangle$

M 30 A: 26 G and input condition imply |LR(I,NR)| <=NJX
Therefore 26A is preserved. 26 B, D, F true by null intersection. 26E true by Lemma 1 since A(AK) unchanged. 26 G true by assignment.

M 30 B: I = NSX+1 by path condition

```
COND  M  26 A:  <* M  20 A *>
COND  M  26 B:  <* M  20 B *>
COND  M  26 C:  <* 1<=I<=NSX  *>
COND  M  26 D:  <* ONE OF THE SET DII,DIII *>
C           DII:  <* IZ<ICD<=IG+1, F=.F., SW=.F., NR=1 *>
C          DIII:  <* IG+1<ICD<=IR, F=.F., SW=.T., NR=ICD-IG *>
COND  M  26 E:  <* M  22 E *>
COND  M  26 F:  <* M  22 F *>
COND  M  26 G:  <* M  22 G, 1<=|LR(II,NR)|<=NJ, 1<=II<I *>
 26     IF ( LR(I,NR) .EQ. 0 )  GO TO 30
        NJ = MAX0( NJ, IABS( LR(I,NR) )  )
        I=I+1
        IF  (I.LE.NSX)  GO TO 26
COND  M  26 A:  <* M  20 A *>
COND  M  26 B:  <* M  20 B *>
COND  M  26 C:  <* 1<=I<=NSX  *>
COND  M  26 D:  <* ONE OF THE SET DII,DIII *>
C           DII:  <* IZ<ICD<=IG+1, F=.F., SW=.F., NR=1 *>
C          DIII:  <* IG+1<ICD<=IR, F=.F., SW=.T., NR=ICD-IG *>
COND  M  26 E:  <* M  22 E *>
COND  M  26 F:  <* M  22 F *>
COND  M  26 G:  <* M  22 G, 1<=|LR(II,NR)|<=NJ, 1<=II<I *>
```

RHO COSET 10 = ⟨NJ, I⟩

PC COSET 10 = ⟨* LR(I,NR) ≠ 0, I <= NSX *⟩

See PATH 9 for M 26, A, B, D, E, F, G

M 26 C: path condition

```
COND  M  26 A:  <* M  20 A *>
COND  M  26 B:  <* M  20 B *>
COND  M  26 C:  <* 1<=I<=NSX  *>
COND  M  26 D:  <* ONE OF THE SET DII,DIII *>
C          DII:  <* IZ<ICD<=IG+1, F=.F., SW=.F., NP=1 *>
C         DIII:  <* IG+1<ICD<=IR, F=.F., SW=.T., NP=ICD-IG *>
COND  M  26 E:  <* M  22 E *>
COND  M  26 F:  <* M  22 F *>
COND  M  26 G:  <* M  22 G, 1<=|LR(II,NR)|<=NJ, 1<=II<I *>
 26     IF ( LR(I,NR) .EQ. 0 )  GO TO 30
COND  M  30 A:  <* M  26 A,B,D,E,F,G *>
COND  M  30 B:  <* 1<I<=NSX+1 *>
```

RHO COSET 11 = ⟨∅⟩

PC COSET 11 = ⟨* LR(I,NR) = 0 *⟩

M 30 A: Null intersection

M 30 B: LR(I,NR) = 0 implies I > 1 by input condition and 26 F.

COSET CONTROL PATH NUMBER 12

```
COND  M  30 A:  <* M  26 A,B,D,E,F,G *>
COND  M  30 B:  <* 1<I<=NSX+1 *>
 30     LS(NR) = I - 1
        LF(NR) = MULT * LS(NR)
        WRITE(6,301) NCD,MULT,(LR(J-1,NR),J=2,I)
COND  M  30.3  A:  <* GC(NR), KN=0 *>
COND  M  30.3  B:  <* K, ID=0 *>
COND  M  30.3  C:  <* NCD=INPUT(1,ICD)  *>
COND  M  30.3 D:  <*  M  26 D  *>
COND  M  30.3  E:  <* M  22 E *>
```

RHO COSET 12 = $\langle$ LS(NR), LF(NR) $\rangle$

PC COSET 12 = $\emptyset$

M 30.3 A: GC(NR-1) M 26 G and assignments

M 30.3 B,C,D,G: Null intersection

```
COND  M  30.3  A:  <* GC(NR), KN=0 *>
COND  M  30.3  B:  <* K, ID=0 *>
COND  M  30.3  C:  <* NCD=INPUT(1,ICD)  *>
COND  M  30.3 D:  <*  M  26 D  *>
COND  M  30.3  E:  <* M  22 E *>
        IF ( NCD .EQ. 2 ) GO TO 40
COND  M  30.4  A:  <* M  30.3  A *>
COND  M  30.4  B:  <* M  30.3  B *>
COND  M  30.4  C:  <*  M  22  E  *>
COND  M  30.4  D:  <* M 26 DII *>
```

RHO COSET 13 = $\langle \emptyset \rangle$

PC COSET 13 = $\langle$* NCD ¬ = 2 *$\rangle$

M 30.4 A,B,C: Null intersection

M 30.4 D: Path condition and input condition

```
COND  M  30.3  A:  <* GC(NR), KN=0 *>
COND  M  30.3  B:  <* K, ID=0 *>
COND  M  30.3  C:  <* NCD=INPUT(1,ICD)  *>
COND  M  30.3 D:  <*  M  26 D  *>
COND  M  30.3  E:  <* M  22 E *>
           IF ( NCD .EQ. 2 ) GO TO 40
COND  M  40 A:  <* M  30.3 A *>
COND  M  40 B:  <* M  30.3 B *>
COND  M  40 C:  <*  M  22 E  *>
COND  M  40 D:  <*  M  26 D, NCD=2  *>
```

RHO COSET 14 = ⟨∅⟩

PC COSET 14 = ⟨* NCD = 2 *⟩

M 40 A,B,C: Null intersection

M 40 D: Path condition and input condition

```
COSET   CONTROL PATH   NUMBER  15

COND   M   30.4   A:   <* M   30.3   A *>
COND   M   30.4   B:   <* M   30.3   B *>
COND   M   30.4   C:   <*   M   22   E   *>
COND   M   30.4   D:   <* M 26 DII *>
                 CALL APPLY(1,1)
                 IF   (N.GT.NMAX)   GO TO 9999
                 GO TO 20
COND   M   20 A:   <* GC(NR-1), KN=0 *>
COND   M   20 B:   <* K, ID=0 *>
COND   M   20 D:   <* ONE OF THE SET DI,DII,DIII *>
C             DI:   <*  0<=ICD<=IZ, F=.T., SW=.F., NR=1  *>
C            DII:   <* IZ<ICD<=IG, F=.F., SW=.F., NR=1 *>
C           DIII:   <* IG<ICD<=IR, F=.F., SW=.T., NR=ICD-IG+1 *>
COND   M   20 E:   <* <U(I), 0<=I<=MIN(ICD-IZ,IG-IZ)> < H(AK) *>
```

RHO COSET 15 = ⟨ρ (APPLY)⟩

PC COSET 15 = ⟨* N <=NMAX *⟩

M 20 A,B: Property of apply and path condition

M 20 D: Null intersection

M 20 E: M 26 E, DII, and APPLY.

```
COND  M  30.4  A:  <* M  30.3  A *>
COND  M  30.4  B:  <* M  30.3  B *>
COND  M  30.4  C:  <*  M  22  E  *>
COND  M  30.4  D:  <* M 26 DII *>
            CALL APPLY(1,1)
            IF  (N.GT.NMAX)  GO TO 9999
COND  M  9999:  <* N=|F:H| OR N=NMAX+1 *>
```

RHO COSET 16 = $\langle \rho(\text{APPLY}) \rangle$

PC COSET 16 = $\langle N > \text{NMAX} \rangle$

M 9999 (SECOND CHOICE) : path condition

```
COND   M  40 A:   <* M   30.3 A *>
COND   M  40 B:   <* M   30.3 B *>
COND   M  40 C:   <*  M  22 F   *>
COND   M  40 D:   <*  M  26 D, NCD=2  *>
 40     SW = .TRUE.
        NR = NR + 1
        GO TO 20
COND   M  20 A:   <* GC(NR-1), KN=0 *>
COND   M  20 B:   <* K, ID=0 *>
COND   M  20 D:   <* ONE OF THE SET DI,DII,DIII *>
C           DI:   <*  0<=ICD<=IZ, F=.T., SW=.F., NR=1  *>
C          DII:   <* IZ<ICD<=IG, F=.F., SW=.F., NR=1 *>
C         DIII:   <* IG<ICD<=IR, F=.F., SW=.T., NR=ICD-IG+1 *>
COND   M  20 E:   <* <U(I), 0<=I<=MIN(ICD-IZ,IG-IZ)> < H(AK)  *>
```

RHO COSET 17 = $\langle$ SW,NR $\rangle$

PC COSET 17 = Ø

M 20 A: increase of NR by 1, M 40 A

M 20 B: Null intersection

M 20 D: M 40 D and assignment implies DIII

M 20 E: Null intersection

```
COND  M  60 A:  <* M  20 A *>
COND  M  60 B:  <* M  20 B *>
COND  M  60 C:  <* M  22 E *>
COND  M  60 D:  <* ONE OF THE SET DI, DII *>
C            DI:  <* 1<=ICD<=IZ, NR=1, F=.T., SW=.F.  *>
C           DII:  <* ICD=IR+1, F=.F., NR=IR-IG+1 *>
 60     IF (F)  GO TO 20
        LS(NR) = 4*NJ
        LF(NR) = 4*NJ
        I=1
COND  M  66 A:  <* M  60, DII *>
COND  M  66 B:  <* 1<=I<=NJ *>
COND  M  66 C:  <* LS(NR)=LF(NR)=4*NJ *>
COND  M  66 D:  <* LP(K+4*II,NR)=II IF K=0 OR -3
C                      =-II IF K=-1 OR -2, 1<=II<I  *>
```

RHO COSET 18 = ⟨LS(NR), LF(NR), I⟩

PC COSET 18 = ⟨* F = .F. *⟩

M 66 A: Path condition and null intersection

M 66 B,C: Assignment

M 66 D: Vacuous

```
COND  M  60 A:  <* M  20 A *>
COND  M  60 B:  <* M  20 B *>
COND  M  60 C:  <* M  22 E *>
COND  M  60 D:  <* ONE OF THE SET DI, DII *>
C          DI:  <* 1<=ICD<=IZ, NR=1, F=.T., SW=.F.  *>
C         DII:  <* ICD=IR+1, F=.F., NR=IR-IG+1 *>
 60   IF (F)  GO TO 20
COND  M  20 A:  <* GC(NR-1), KN=0 *>
COND  M  20 B:  <* K, ID=0 *>
COND  M  20 D:  <* ONE OF THE SET DI,DII,DIII *>
C          DI:  <*  0<=ICD<=IZ, F=.T., SW=.F., NR=1  *>
C         DII:  <* IZ<ICD<=IG, F=.F., SW=.F., NR=1 *>
C        DIII:  <* IG<ICD<=IR, F=.F., SW=.T., NR=ICD-IG+1 *>
COND  M  20 E:  <* <U(I), 0<=I<=MIN(ICD-IZ,IG-IZ)> < H(AK) *>
```

RHO COSET 19 = ⟨∅⟩

PC COSET 19 = ⟨* F = .T. *⟩

M 20 A,B,E: Null intersection

M 20 D=DI: Path condition

```
COND  M  66 A:  <* M  60, DII *>
COND  M  66 B:  <* 1<=I<=NJ *>
COND  M  66 C:  <* LS(NR)=LF(NR)=4*NJ *>
COND  M  66 D:  <* LR(K+4*II,NR)=II IF K=0 OR -3
C                    =-II IF K=-1 OR -2, 1<=II<I  *>
 66     LR(4*I-3,NR) = I
        LR(4*I-2,NR) = -I
        LR(4*I-1,NR) = -I
        LR(4*I,NR) = I
        I=I+1
       IF  (I.LE.NJ)  GO TO 66
        LWD=0
        NCOS=1
C  LET HO = <H(I), 0<=I<=IG-IZ>
COND  M  70 A:  <* GC(NR), KN<N *>
COND  M  70 B:  <* K, ID=0 *>
COND  M  70 C:  <*  H(KN,AK) < H(AK)  *>
COND  M  70  D:  <*  THERE EXISTS P, 1<=P<=KN+NCOS SUCH THAT
C                    (1,W,P) < KF'+KR' FOR ALL W OF LENGTH
C                    LESS THAN OR EQUAL TO LWD  *>
```

RHO COSET 20=$\langle$(LR(4*I-K, NR), 0 <= K <= 3), I$\rangle$

PC COSET 20=$\langle$* I > NJ *$\rangle$

M 70 A: M 66 and Path condition

M 70 B: Null intersection

M 70 C: Null intersection since KN = 0

M 70 D: Vacuous, LWD = 0

```
      COND  M  66 A:  <* M  60, DTI *>
      COND  M  66 B:  <* 1<=I<=NJ *>
      COND  M  66 C:  <* LS(NR)=LF(NR)=4*NJ *>
      COND  M  66 D:  <* LR(K+4*II,NR)=II IF K=0 OR -3
      C                    =-II IF K=-1 OR -2, 1<=II<I  *>
   66    LR(4*I-3,NR) = I
         LR(4*I-2,NR) = -I
         LR(4*I-1,NR) = -I
         LR(4*I,NR) = I
         I=I+1
         IF  (I.LE.NJ)  GO TO 66
      COND  M  66 A:  <* M  60, DII *>
      COND  M  66 B:  <* 1<=I<=NJ *>
      COND  M  66 C:  <* LS(NR)=LF(NR)=4*NJ *>
      COND  M  66 D:  <* LR(K+4*II,NR)=II IF K=0 OR -3
      C                    =-II IF K=-1 OR -2, 1<=II<I  *>
```

RHO COSET 21 = ⟨ (LR(4*I-K, NR), 0 <= K <= 3), I ⟩

PC COSET 21 = ⟨* I <= NJ *⟩

M 66 A,C: Null intersection

M 66 B: Path condition

M 66 D: Assignment

```
C  LET H0 = <U(I), 0<=I<=IG-I7>
COND  M  70 A:  <* GC(NR), KN<N *>
COND  M  70 B:  <* K, ID=0 *>
COND  M  70 C:  <*  H(KN,AK) < H(AK)  *>
COND  M  70  D:  <*  THERE EXISTS P, 1<=P<=KN+NCOS SUCH THAT
C                    (1,W,P) < KP'+KR' FOR ALL W OF LENGTH
C                    LESS THAN OR EQUAL TO LWD  *>
 70     KN = KN + 1
        I=1
        NCOS=NCOS-1
        IF  (NCOS.GE.0)  GO TO 76
        LWD=LWD+1
        NCOS=N-KN
COND  M  76 A:  <* GC(NR), 1<=KN<=N *>
COND  M  76 B:  <* M  70 B *>
COND  M  76 C:  <* 1<=I<=NR *>
COND  M  76 D:  <*  H(KN-1,AK)+<W(1,KN)W(L)W(KN,1), 1<=L<I >
C                         <  H(AK)  *>
COND  M  76 E:  <*  M  70 D  *>
```

RHO COSET 22 = ⟨KN, I, NCOS, LWD⟩

PC COSET 22 = ⟨* NCOS < 0 *⟩

M 76 A-D: Same as Path 23

M 76 E: M 70 and path condition implies (1,W,p) ∈ KF' + KR'
for 1 <= p <= KN-1 and since
(KN-1, W(NR), KN-1) ∈ KF' + KR',
(p,J,p') ∈ KF' + KR', 1 <= p <= KN-1,
1 <= p' <= N, thus word length can be increased
by 1.

```
C  LET HO = <U(I), 0<=I<=IG-IZ>
COND  M  70 A:  <* GC(NR), KN<N *>
COND  M  70 B:  <* K, ID=0 *>
COND  M  70 C:  <*  H(KN,AK) < H(AK)   *>
COND  M  70  D:  <*  THERE EXISTS P, 1<=P<=KN+NCOS SUCH THAT
C                     (1,W,P) < KF'+KR' FOR ALL W OF LENGTH
C                     LESS THAN OR EQUAL TO LWD  *>
 70     KN = KN + 1
        I=1
        NCOS=NCOS-1
        IF  (NCOS.GE.0)  GO TO 76
COND  M  76 A:  <* GC(NR), 1<=KN<=N *>
COND  M  76 B:  <* M  70 B *>
COND  M  76 C:  <* 1<=I<=NP *>
COND  M  76 D:  <*  H(KN-1,AK)+<W(1,KN)W(L)W(KN,1), 1<=L<I >
C                         <  H(AK)   *>
COND  M  76 E:  <*  M  70 D  *>
```

RHO COSET 23 = ⟨ KN, I, NCOS ⟩

PC COSET 23 = ⟨* NCOS ≥ 0 *⟩

M 76 A: Assignment for KN, null intersection for remainder

M 76 B: Null intersection

M 76 C: Assignment

M 76 D: Vacuous

M 76 E: Null intersection, KN+NCOS constant.

```
COND   M   76 A:  <* GC(NR), 1<=KN<=N *>
COND   M   76 B:  <* M   70 B *>
COND   M   76 C:  <* 1<=I<=NR *>
COND   M   76 D:  <*  H(KN-1,AK)+<W(1,KN)W(L)W(KN,1), 1<=L<I >
C                      <  H(AK)   *>
COND   M   76 E:  <*  M   70 D  *>
 76      IF   (KV(KN)).NE.KN)  GO TO 75
         IF ( N .GT. NMAX - LF(NR) )  CALL CONSOL
         CALL  APPLY  (KN,I)
         IF ( N .GT. NMAX )  GO TO 9999
COND   M   75 A:  <* M   76 A *>
COND   M   75 B:  <* M   76 B *>
COND   M   75 C:  <* M   76 C *>
COND   M   75 D:  <*  H(KN-1,AK)+<W(1,KN)W(L)W(KN,1), 1<=L<=I >
C                      <  H(AK)   *>
COND   M   75 E:  <*  M   70 D  *>
```

RHO COSET 24 = $\langle$ ρ(CONSOL) ρ(APPLY), I $\rangle$

PC COSET 24 = $\langle$* KV(KN) = KN, N <= NMAX, I <= NR *$\rangle$

M 76 is a p condition for 76.2 by null intersection or by property of CONSOL

M 75 By property of APPLY and path condition

COSET CONTROL PATH NUMBER 25

```
COND  M  76 A:  <* GC(NR), 1<=KN<=N *>
COND  M  76 B:  <* M  70 B *>
COND  M  76 C:  <* 1<=I<=NR *>
COND  M  76 D:  <*  H(KN-1,AK)+<W(1,KN)W(L)W(KN,1), 1<=L<I >
C                    <  H(AK)   *>
COND  M  76 E:  <*  M  70 D  *>
 76      IF  (KV(KN)).NE.KN)  GO TO 75
         IF ( N .GT. NMAX - LF(NR) )  CALL CONSOL
         CALL  APPLY  (KN,I)
         IF ( N .GT. NMAX )  GO TO 9999
 COND  M  9999:  <* N=|F:H| OR N=NMAX+1 *>
```

RHO COSET 25 = $\langle\ \rho$(CONSOL), ρ(APPLY) $\rangle$

PC COSET 25 = $\langle *$ NMAX $<$ N $* \rangle$

M 9999 (SECOND CHOICE): Path condition

COSET CONTROL PATH NUMBER 26

COND M 76 A: ⟨* GC(NR), 1 ⇐ KN ⇐ N *⟩

COND M 76 B: ⟨* M 70 B *⟩

COND M 76 C: ⟨* 1 ⇐ I ⇐ NR *⟩

COND M 76 D: ⟨*H(KN-1,AK) +⟨ W(1,KN) W(L) W(KN,1), 1 ⇐ L ⟨ I ⟩

C ⟨ H(AK) *⟩

COND M 76 E: ⟨* M 70 D *⟩

76 if (KV(KN).NE. KN) got to 75

COND M 75 A: ⟨* M 76 A *⟩

COND M 75 B: ⟨* M 76 B *⟩

COND M 75 C: ⟨* M 76 C *⟩

COND M 75 D: ⟨* H(KN-1,AK)+⟨ W(1,KN) W(L) W(KN,1)

C 1 ⇐ L ⇐ I ⟩ ⟨ H(AK) *⟩

COND M 75 E: ⟨* M 70 D *⟩

RHO COSET 26 = ∅

PC COSET 26 = ⟨* KV(KN).NE.KN *⟩

M 75 by null intersection and path condition since

KV(KN) ≠ KN implies (KN,W(L),KN) ∊ KF' + KR',

1 ⇐ L ⇐ NR.

```
COND  M  75 A:  <* M  76 A *>
COND  M  75 B:  <* M  76 B *>
COND  M  75 C:  <* M  76 C *>
COND  M  75 D:  <*  H(KN-1,AK)+<W(1,KN)W(L)W(KN,1),  1<=L<=I >
C                         <  H(AK)   *>
COND  M  75 E:  <*  M  70 D  *>
 75      I=I+1
         IF (I.LE.NR)  GO TO 76
         IF ( KN .LT. N )  GO TO 70
COND  M  75.3 A:  <* GC(NR), KN=N *>
COND  M  75.3 B:  <* M  70 B *>
COND  M  75.3 C:  <*  H(N,AK) < H(AK)  *>
```

RHO COSET 27 = ⟨ I ⟩

PC COSET 27 = ⟨* I > NR, KN = N ⟩

M 75.3 A: Path condition

M 75.3 B: Null intersection

M 75.3 C: Path condition and assignment

```
COND  M  75 A:  <* M  76 A *>
COND  M  75 B:  <* M  76 B *>
COND  M  75 C:  <* M  76 C *>
COND  M  75 D:  <*  H(KN-1,AK)+<W(1,KN)W(L)W(KN,1), 1<=L<=I >
C.                       <  H(AK)   *>
COND  M  75 E:  <*  M  70 D  *>
 75   I=I+1
      IF (I.LE.NR)  GO TO 76
      IF ( KN .LT. N )  GO TO 70
C  LET HO = <U(I), 0<=I<=IG-IZ>
COND  M  70 A:  <* GC(NR), KN<N *>
COND  M  70 B:  <* K, ID=0 *>
COND  M  70 C:  <*  H(KN,AK) < H(AK)   *>
COND  M  70  D:  <*  THERE EXISTS P, 1<=P<=KN+NCOS SUCH THAT
C                    (1,W,P) < KF'+KR' FOR ALL W OF LENGTH
C                    LESS THAN OR EQUAL TO LWD  *>
```

RHO COSET 28 = $\langle$ I $\rangle$

PC COSET 28 = $\langle$* I > NR, KN < N *$\rangle$

M 40 A,C: Path condition

M 40 B: Null intersection

```
COND  M  75 A:  <* M  76 A *>
COND  M  75 B:  <* M  76 B *>
COND  M  75 C:  <* M  76 C *>
COND  M  75 D:  <*  H(KN-1,AK)+<W(1,KN)W(L)W(KN,1),  1<=L<=I >
C                       <  H(AK)   *>
COND  M  75 E:  <*  M  70 D  *>
 75     I=I+1
        IF (I.LE.NR)  GO TO 76
COND  M  76 A:  <* GC(NR),  1<=KN<=N *>
COND  M  76 B:  <* M  70 B *>
COND  M  76 C:  <* 1<=I<=NR *>
COND  M  76 D:  <*  H(KN-1,AK)+<W(1,KN)W(L)W(KN,1),  1<=L<I >
C                       <  H(AK)   *>
COND  M  76 E:  <*  M  70 D  *>
```

RHO COSET 29 = ⟨ I ⟩

PC COSET 29 = ⟨* I <= NR *⟩

M 76 A,B: Null intersection

M 76 C,D: Path condition and assignment

```
COND  M  75.3 A:  <* GC(NR), KN=N *>

COND  M  75.3 B:  <* M  70 B *>

COND  M  75.3 C:  <*  H(N,AK) < H(AK)  *>

          CALL CONSOL

COND  M  9999:  <* N=|F:H| OR N=NMAX+1 *>
```

RHO COSET 30 = $\langle \rho(\text{CONSOL}) \rangle$

PC COSET 30 = $\emptyset$

M 9999 (FIRST CHOICE): Property of CONSOL

Proof of termination for MAIN PROGRAM COSET

Q(COSET) = ⟨ F 17.4, F 20, F 26.2, F 66, F 70, F 76 ⟩

Expressions: e1 = λ, e2 = ICD, e3 = LWD, e4 = -NCOS,
e5 = I, e6 = J

Proof that ⟨ ei, i = 1,6 ⟩ is a controlled set of expressions:

PATH 1 17.4 —> 17 —> 17.4

λ, ICD, LWD, NCOS and I are constant, J increases and J <= NJ+1 at 17.4 by COND M 17

PATH 2 17.4 —> 16 —> 17 —> 17.4

λ, ICD, LWD, NCOS are constant, I increases and I <= NMAX at 17.4 by COND M 17.

PATH 3 17.4 —> 20

λ increases and λ is constant at 20.

PATH 4 20 —> 22 —> 26 —> 26.2

λ is constant, ICD increases, ICD <= IR at 26.2 by COND M 26.

(Note that 20 —> 22 —> 26 —> 30 ... is a path that will note be taken due to INPUT and PATH CONDITIONS.)

PATH 5 20 —> 22 —> 60 —> 20

λ is constant, ICD increases and ICD <= IR at 20 by COND M 20.

PATH 6 20 —> 22 —> 60 —> 66

λ increases and λ is constant at 66.

PATH 7 26.2 —> 26 —> 26.2

λ, ICD, LWD, and NCOS are constant, I increases and I <= NSX at 26.2 by COND M 26.

PATH 8 26.2 —> 26 —> 30 ==> 20

λ, ICD, LWD, and NCOS are constant, I increases and I <= NSX+1 at 20 by COND M 30.

PATH 9 66 —> 66

λ, ICD, LWD, and NCOS are constant, I increases and I <= NJ at 66 by COND M 66.

PATH 10 66 —> 70

λ increases and λ is constant at 70.

PATH 11 76 ==> 75 —> 76

λ, ICD, LWD and NCOS are constant, I increases and I <= NR at 76 by COND M 76.

PATH 12 76 ==> 75 —> 70

λ, ICD, LWD, NCOS are constant, I increases and I <= NR+1 at 70 by COND M 75 and assignment I = I+1

PATH 13 70 —> 76 PC = <* N COS >= 0 *>

λ, ICD, LWD are constant, -NCOS increases, -NCOS <= 0 at 76, by path condition.

PATH 14 70 —> 70.1 —> 70.2 —> 70.3 —> 70.4 —> 70.5 —> 76.

λ, ICD are constant. LWD increases. To show LWD bounded consider two cases.

(i) $[\mathfrak{F}:\mathscr{H}]$ is suitably small. Then $\mathfrak{F} = W_1\mathscr{H} + W_2\mathscr{H} + \cdots + W_n\mathscr{H}$. Let LWX = maximum length of W_i above then LWD < LWX otherwise $\mathscr{H} \in$ H(AK) and KN = N, the array AK is complete. (See Trotter [1] for proof.)

(ii) More than NMAX+1 distinct COSETS will be found during enumeration then let LWX = maximum of length of COSET representatives of COSETS 1, NMAX+1. Then LWD < LWX or N > NMAX.

REFERENCES

(1) H. F. Trotter, A machine program for coset enumeration, Canadian Math. Bull. 7 (1964), 357-368.

(2) J. Leech, Coset enumeration, in: Computational Problems in Abstract Algebra, ed. J. Leech (Pergamon Press, Oxford, 1970).

(3) W. D. Maurer, Algorithm Verification, Memorandum No. ERL M290, Electronics Research Laboratory, University of California, Berkeley, January 1971.

(4) ——————, Validity of Return Address Schemes, to appear in Information Science, 1972.

Appendix A. Listing of COSET, and subroutines with preconditions.

Non executable statements do not appear in the listings. Associated with each (sub) program is the statement:

```
COMMON KL(500), KV(500), KF(500,5), KR(500,5), LS(10),
LF(10), LR(20,10), NMAX, N, NR, NJ, ID, NVAC, KN
```

Associated with COSET is the type declaration:

```
LOGICAL SW, F
```

```
        NOTE          PROGRAM WITH PRECONDITIONS

      COND          <*  GC(NB)   *>     GLOBAL
      COND   N 1:   <*  K,  0<=IX,IY<=N   *>
      COMMENT   LET X,Y BE THE INITIAL VALUES OF EC(IX),EC(IY)
       1       IA = IX
               IF ( IA .EQ. 0 )   GO TO 9999
               IB = IY
       10      IF ( IB .NE. 0 ) GO TO 20
               IY = IA
               GO TO 9999
      COND   N 20 A:  <*  K   *>
      COND   N 20 B:  <*  IB=IY,  0<IX,IY<=N   *>
      COND   N 20 C:  <*  EC(IA)=X,   IA BELONGS TO EC CHAIN WITH
      C                        HEAD IY AND TAIL X.   *>
       20      IF ( KV(IA) .GE. 0 )   GO TO 204
               IA = - KV(IA)
               GO TO 20
       204     IA = KV(IA)
      COND   N 205 A:  <*   K   *>
      COND   N 205 B:  <*   IA=X,   0<IX,IY<=N   *>
      COND   N 205 C:  <*   EC(IB)=Y,   IB BELONGS TO EC CHAIN WITH
      C                         HEAD IY AND TAIL Y.   *>
       205     IF ( KV(IB) .GE. 0 ) GO TO 206
               IB = - KV(IB)
               GO TO 205
       206     IB = KV(IB)
      COND   N 206.1 A:  <*   K   *>
```

```
      COND  N 206.1 B:  <*  0<IA=X<=N,  0<IB=Y<=N  *>
            IF  (IB.EQ.IA)  GO TO 9999
            IF  (IB.GT.IA)  GO TO 30
       25   IT = IA
            IA = IB
            IB = IT
      COND  N 30 A:  <*  K  *>
      COND  N 30 B:  <*  0<IA<IB,  IA=MIN(X,Y),  IB=MAX(X,Y)  *>
      COMMENT   LET  A BE SUCH THAT I(A)=IA, SIMILARLY I(B)=IB
       30   KV(IB)  = -IA
            IT = IB
            M=0
      COND  N 40 A:  <*  K FOR INITIAL VALUES OF KV,KL,KR,KF:
      C                  KL,KR,KF UNCHANGED, KV(IT) UNCHANGED
      C                  FOR IT ¬< C(B)   *>
      COND  N 40 B:  <*  KV(IB)=-IA  *>
      COND  N 40 C:  <*  0<IA<IB,  IA=MIN(X,Y),  IB=MAX(X,Y)  *>
      COND  N 40 D:  <*  0<=M<P(B),  IT=KL(IB)**M  *>
      COND  N 40 E:  <*  KV(I)=IA FOR I=KL(IB)**Q,  1<=Q<=M  *>
       40   IF ( KL(IT) .EQ. IB )  GO TO 60
            IT = KL(IT)
            KV(IT)  = IA
            M=M+1
            GO TO 40
       60   IF ( IT .EQ. IB )  GO TO 70
            KL(IT)  = KL(IA)
            KL(IA)  = KL(IB)
       70   KL(IB)  = ID
            ID = IB
      COND  N 9999 A:  <*  K  *>
```

A4

```
      COND  N 9999 B:  <*  GC(NP)  *>
      COND  N 9999 C:  <*  ONE OF THE SET CI,CII,CIII,CIV  *>
C                CI:  <*  IX=0  *>
C               CII:  <*  IX>0, IY=IX, IB=0)  *>
C              CIII:  <*  IA=EC(IX)=IB=EC(IY)  *>
C               CIV:  <*  IX,IY>0,  IA=MIN(X,Y), IB=MAX(X,Y),
C                         EC(IB)=EC(IX)=EC(IY)=IA  *>
 9999 RETURN
```

```
CONSOL        PROGRAM WITH PRECONDITIONS

COND  C  1:  <* K,ID=0,GC(NR) *>
COND  GLOBAL  <* ID=0,GC(NR) *>
C  X# DENOTES THE INITIAL VALUE OF THE VARIABLE X,
C  X DENOTES THE CURRENT VALUE
 1     IF (NVAC.EQ.0)  GO TO 9999
        M = 0
       I=1
COND  C  21 A:  <*  1<=I<=N,  M=|<II; II<I, KV#(II)=II>|  *>
COND  C  21 B:  <* 1<=KV(K)<KV(KK)<=M FOR K,KK SUCH THAT
C                   1<=K<KK<I, KV#(K)=K, KV#(KK)=KK:
C                    KV(K)=-KV(KV#(K)) FOR K SUCH THAT
C                    1<=K<I, KV#(K)¬=K:  FOR EACH MM SUCH THAT
C                    1<=MM<=M  KV(K)=MM FOR SOME K,
C                   1<=K<I,KV#(K)=K *>
COND  C  21 C:  <* KV(K)=KV#(K), I<=K<=N *>
COND  C  21 D:  <* KL=KL#, KF=KF#, KR=KR# *>
 21     IF ( KV(I) .NE. I ) GO TO 15
        M = M + 1
        KV(I) = M
        GO TO 20
 15     J = KV(I)
        KV(I) = -KV(J)
COND  C  20 A:  <*  1<=I<=N, M=|<II; II<=I, KV#(II)=II>|  *>
COND  C  20 B:  <*  C  21 B FOR M=M-1,(KV#(I)=I, KV(I)=M)
C                OR C 21 B, (KV#(I)¬=I, KV(I)=-KV(KV#(I)) )  *>
COND  C  20 C:  <* KV(K)=KV#(K), I<K<=N *>
```

```
COND  C  20 D:  <* C  21 D *>
 20   I=I+1
      IF (I.LE.N)  GO TO 21
COND  KVIM:  <* M=MCOS#,KV(I)=IM(I) IF KV#(I)=I,
C                KV(I)=-IM(KV#(I)) IF KV#(I)¬=I, 1<=I<=N  *>
COND  C  20.2:  <* C  21 D, KVIM *>
       KN = IABS( KV(KN) )
      I=1
COND  GLOBAL  <*  KN=|IM(KV#(KN#))|  *>
COND  C  31 A:  <* KVIM, KI=KL# *>
COND  C  31 B:  <* 1<=I<=N *>
COND  C  31 C:  <* KF(II,J)=IM(KV#(KF#(II,J))),
C                    KR(II,J)=IM(KV#(KR#(II,J))),
C                   1<=J<=NJ, 1<=II<I,KV#(II)=II *>
COND  C  31 D:  <* KF(II,J)=KF#(II,J), KR(IT,J)=KR#(II,J)
C                    FOR I<=II<=N, 1<=J<=NJ  *>
 31    IF ( KV(I) .LT. 0 )  GO TO 30
       J=1
COND  C  26 A:  <* C  31 A,B,C,  KV(I)>0  *>
COND  C  26 B:  <* 1<=J<=NJ *>
COND  C  26 C:  <* KF(I,JJ)=IM(KV#(KF#(I,JJ))),
C                   KR(I,JJ)=IM(KV#(KR#(I,JJ))), 1<=JJ<J  *>
COND  C  26 D:  <* KF(II,JJ)=KF#(II,JJ), KR(II,JJ)=KR#(II,JJ)
C                   IF  I<II OR I=II, J<=JJ *>
 26    IF ( KF(I,J) .EQ. 0 )  GO TO 23
       KT = KF(I,J)
       KF(I,J) = IABS( KV(KT) )
 23    IF ( KR(I,J) .EQ. 0 )  GO TO 25
       KT = KR(I,J)
       KR(I,J) = IABS( KV(KT) )
```

```
COND  C  25 A:  <* C  26 A,B,C,D FOR J<JJ  *>
COND  C  25 B:  <* KF(I,J)=IM(KV#(KF#(I,J))),
C                 KR(I,J)=IM(KV#(KR#(I,J)))  *>
 25     J=J+1
        IF (J.LE.NJ)  GO TO 26
COND  C  30 A:  <* C  31 A,B,C *>
COND  C  30 B:  <* KV(I)>0 AND KF(I,J)=IM(KV#(KF#(I,J))),
C                 KR(I,J)=IM(KV#(KR#(I,J))), 1<=J<=NJ  *>
C               OR  <*  KV(I)<0  *>
COND  C  30 C:  <* KF(II,J)=KF#(II,J), I<II<=N, 1<=J<=NJ *>
 30     I=I+1
        IF  (I.LE.N)  GO TO 31
        J=0
        I=1
COND  C  41 A:  <*  1<=I<=N,  J=|<II: II<I,KV#(II)=II>|  *>
COND  C  41 B:  <* KL(JJ)=KV(JJ)=JJ,
C                  KF(JJ,K)=IM(KV#(KF#(IMI(JJ),K))),
C                 KR(JJ,K)=IM(KV#(KR#(IMI(JJ),K))),
C                  1<=JJ<=J,  1<=K<=NJ  *>
COND  C  41 C:  <*  KF(II,K)=IM(KV#(KF#(II,K))),
C                  KR(II,K)=IM(KV#(KR#(II,K))), 1<=K<=NJ,
C                  KV(II)=IM(II),  I<II<=N AND KV#(II)=II;
C               KV(II)=-IM(KV#(II)), KV#(II)¬=II, I<II<=N  *>
 41     IF  (KV(I).LT.0)  GO TO 40
        J=J+1
        K=1
COND  C  38 A:  <* 1<=I<=N, 1<=K<=NJ, KV(I)>0, J=IMI(I)  *>
COND  C  38 B:  <* C  41 B WITH J REPLACED BY J-1,
C                 KF(J,KK)=IM(KV#(KF#(IMI(J),KK))),
C                 KR(J,KK)=IM(KV#(KR#(IMI(J),KK))),
```

```
C                       1<=KK<K   *>
COND   C  38 C:  <* C   41 C *>
 38      KF(J,K)  = KF(I,K)
 37      KR(J,K)  = KR(I,K)
         K=K+1
         IF  (K.LE.NJ)   GO TO 38
         KL(J)  = J
         KV(J)=J
COND   C  40 A:  <*  1<=I<=N, J=|<II; II<=I,KV#(II)=II>|   *>
COND   C  40 B:  <* C   41 B *>
COND   C  40 C:  <* C   41 C *>
 40      I=I+1
         IF  (I.LE.N)   GO TO 41
COND   C  40.2:  <*  KF(J,K)=IM(KV#(KF#(IMI(J),K))),
C                    KR(J,K)=IM(KV#(KR#(IMI(J),K))),
C                    KV(J)=KL(J)=J,   1<=K<=NJ,  1<=J<=M   *>
         KT = M + 1
         I=KT
COND   C  51 A:  <* C   40.2 *>
COND   C  51 B:  <* M+1<=I<=N *>
COND   C  51 C:  <* KF(II,J)=KR(II,J)=0,  1<=J<=NJ,  KT<=II<I *>
 51      J=1
COND   C  52:  <*   C   51;  1<=J<=NJ;
C                   KF(I,JJ)=KR(I,JJ)=0,  1<=JJ<J   *>
 52      KF(I,J)  = 0
         KR(I,J)  = 0
 50      J=J+1
         IF  (J.LE.NJ)   GO TO 52
         I=I+1
         IF  (I.LE.N)   GO TO 51
```

```
      N = M
      NVAC = 0
COND  C  9999:  <* K,ID=0,NVAC=0,H(AK)=H(AK#),
C                     KN=IM(KV#(KN#))   *>
 9999 RETURN
```

```
      APPLY         PROGRAM WITH PRECONDITIONS

COMMENT  LET HH BE THE INITIAL VALUE OF H(AK)
COND    <*  GC(NR)   *>  PRECONDITION FOR ALL STATEMENTS
C                        EXCEPT 36.1 AND 9999
COND    <*  GC(NR) OR N=NMAX+1   *>  PRECONDITION VALID FOR
C                                    36.1 AND 9999
COND  AP 1 A:  <*  K,ID=0   *>
COND  AP 1 B:  <*  1<=LA<=NR   *>
COND  AP 1 C:  <*  1<=IT<=N,  KV(IT)=IT   *>
 1      L = LA
        LEN = LF(L)
        LNS = LS(L)
        I = IT
        IB = I
        M=LNS
        MX=1
COND  GLOBAL  <*  1<=L<=NR,  1<=LEN<=1000NSX,  1<=LNS<=NSX,
C                 1<=IT<=N   *>
COND  AP 21 A:  <*  K,ID=0   *>
COND  AP 21 B:  <*  1<=MX<=LEN,  1<=M<=LNS,
C                   (M+1)#MX MOD(LNS)   *>
COND  AP 21 C:  <*  1<=I,IB<=N,  KV(I)=I,  KV(IB)=IB=IT   *>
COND  AP 21 D:  <*  CHAIN(I,IT,MX,L)   *>
COND  AP 21 E:  <*  H(AK)=HH   *>
 21     M = M + 1
        IF ( M .GT. LNS )  M = 1
        J = LR(M,L)
```

A11

```
      IF  (J.GT.0)  GO TO 14
13    J = -J
      IX = KP(I,J)
      GO TO 15
14    IX = KF(I,J)
COND  AP  15 A,C,D,E:  <*  AP  21 A,C,D,F  *>
COND  AP  15 B:  <*  1<=MX<=LEN, 1<=M<=LNS, M#MX MOD(LNS)  *>
COND  AP 15 F:  <*  IX=0,  (I,LR(M,L),0) < KF'+KR'  *>
C           OR  <*  IX>0,  (I,LR(M,L),KV(IX)) < KF'+KR'  *>
15    IF ( IX .EQ. 0 )  GO TO 30
20    I = KV(IX)
      MX=MX+1
      IF  (MX.LE.LEN)  GO TO 21
      GO TO 50
30    M=1
      MY=MX
COMMENT  LET MZ = MX+LEN-MY
COND  AP 41 A,E:  <*  AP  21 A,E  *>
COND  AP 41 B:  <*  1<=MX<=MY<=LEN,  1<=M<=LNS,
C                   (M-1)#MZ  MOD(LNS)  *>
COND  AP 41 C:  <*  1<=I,IB<=N,  KV(I)=I,  KV(IB)=IB,
C                   KV(IT)=IT  *>
COND  AP 41 D:  <*  CHAIN(IT,I,MY,L),
C                   CHAIN(IB,IT,-(MY-MX),L))  *>
41    M = M - 1
      IF ( M .EQ. 0 )  M = LNS
      J = LR(M,L)
      IF  (J.GT.0)  GO TO 34
33    JT = -J
      IX = KF(IB,JT)
```

```
      GO TO 35
 34   IX = KR(IB,J)
C     COND  AP  35 A,C,D,E:  <*  AP  41 A,C,D,E  *>
C     COND  AP 35 B:  <*  1<=MX<=MY<=LEN,  1<=M<=LNS,
C                         M#MZ  MOD(LNS),  J=LR(M,L)   *>
C     COND  AP 35 F:  <*  IX=0,   (0,J,IB) < KF'+KP'   *>
C               OR    <*  IX>0,   (KV(IX),J,IB) < KF'+KR'   *>
 35   IF ( IX .NE. 0 )  GO TO 40
      IF ( MY .LT. LEN )  GO TO 36
      IF ( J .GT. 0 )  GO TO 355
      KF(IB,JT) = I
      CALL NOTE( IB,KR(I,JT) )
      GO TO 60
 355  KR(IB,J) = I
      CALL NOTE( IB,KF(I,J) )
      GO TO 60
 36   N = N + 1
      IF ( N .GT. NMAX )  GO TO 9999
      KL(N) = N
      KV(N) = N
      IY = N
      IF ( J .GT. 0 )  GO TO 37
      KF(IB,JT) = IY
      KR(IX,JT) = IB
      GO TO 40
 37   KR(IB,J) = IX
      KF(IX,J) = IB
C     COND  AP 40 :  <*  AP 35 , IX>0  *>
 40   IB = KV(IX)
      MY=MY+1
```

```
      IF  (MV.LE.LEN)  GO TO 41
COND  AP 50 A:  <*  K,ID=0  *>
COND  AP 50 B:  <*  1<=I,IB<=N,  KV(I)=I, KV(IB)=IB  *>
COND  AP 50 C:  <*  CHAIN(IB,I,LEN+1,L)  *>
C            OR <*  CHAIN(IT,I,MX,L),
C                   CHAIN(IB,IT,-(LEN-MX+1),L)  *>
COND  AP 50 D:  <*  H(AK)=HH  *>
 50   CALL NOTE(I,IB)
COND  AP 60 A:  <*  K  *>
COND  AP 60 B:  <*  H(AK)=<HH+W(1,IT)W(L)W(IT,1)>  *>
 60   IF ( ID .EQ. 0 )  GO TO 9999
      NVAC = NVAC + 1
      I = ID
      ID = KL(I)
      IA = I
      KL(I)=I
      J=1
COND  AP 64 A:  <*  K0,K1,K2,K3,  INCLUDING I IN THE SET OF
C                  ROWS LABELED EC(I), MCOS+NVAC+P(0)=N+2  *>
COND  AP 64 B:  <*  AP  60 B  *>
COND  AP 64 C:  <*  IA BELONGS TO EC CHAIN WITH HEAD I,
C                   TAIL EC(I)  *>
COND  AP 64 D:  <* 1<=J<=NJ  *>
 64   IF ( KV(IA) .GE. 0 )  GO TO 65
      IA = -KV(IA)
      GO TO 64
 65   IA = KV(IA)
      KV(I) = IA
      KL(I) = KL(IA)
      KL(IA) = I
```

```
COND  AP  71 A:  <*  K INCLUDING I IN THE SET OF ROWS
C                     LABELED IA,  EC(I)=IA   *>
COND  AP  71 B:  <*  AP  60 B  *>
COND  AP 71 C:  <*  1<=J<=NJ  *>
COND  AP 71 D:  <* (KF(I,L)=0 OR EC(KF(IA,L))=EC(KF(I,L)))
C                AND (KR(I,L)=0 OR EC(KR(IA,L))=EC(KR(I,L)))
C                    FOR 1<=L<J  *>
 71     CALL NOTE ( KF(I,J) , KF(IA,J) )
        CALL NOTE ( KR(I,J) , KR(IA,J) )
 70     J=J+1
        IF  (J.LE.NJ)  GO TO 71
        GO TO 60
COND  AP 9999 A:  <*  K,ID=0  *>
COND  AP 9999 B:  <*  GC(NR)  *>
COND  AP 9999 C:  <*  H(AK)=<HH+W(1,IT)W(L)W(IT,1)>  *>
C   OR
COND  AP 9999 D:  <*  N=NMAX+1  *>
 9999  RETURN
```

```
      COSET        PROGRAM WITH PRECONDITIONS

COND  GLOBAL  <*  INPUT  *>
 1      NMAX = 500
        NJX = 5
        NRX = 10
        NSX = 20
 10     NVAC = 0
        ID = 0
        NJ = 1
        I=1
COND  M  16 A:  <* 1<=NJ<=NJX, NVAC=0, ID=0 *>
COND  M  16 B:  <* 1<=I<=NMAX *>
COND  M  16 C:  <* (KF(II,J)=KR(II,J)=0, 1<=J<=NJX),
C                  KV(II)=KL(II)=0,  1<=II<I  *>
 16     KL(I) = 0
        KV(I) = 0
        J=1
COND  M  17 A:  <* M  16 *>
COND  M  17 B:  <* 1<=J<=NJX *>
COND  M  17 C:  <* KF(I,JJ)=KR(I,JJ)=0, 1<=JJ<J,
C                  KV(I)=KL(I)=0  *>
 17     KF(I,J) = 0
        KR(I,J) = 0
        J=J+1
        IF  (J.LE.NJX)  GO TO 17
        I=I+1
        IF  (I.LE.NMAX)  GO TO 16
```

```
      N = 1
      KL(1) = 1
      KV(1) = 1
      NR = 1
      F = .TRUE.
      SW = .FALSE.
      KN = 0
      ICD=0
COND  M  20 A:  <* GC(NR-1), KN=0 *>
COND  M  20 B:  <* K, ID=0 *>
COND  M  20 D:  <* ONE OF THE SET DI,DII,DIII *>
C          DI:  <*  0<=ICD<=IZ, F=.T., SW=.F., NR=1  *>
C         DII:  <* IZ<ICD<=IG, F=.F., SW=.F., NR=1 *>
C        DIII:  <* IG<ICD<=IR, F=.F., SW=.T., NR=ICD-IG+1 *>
COND  M  20 E:  <* <U(I), 0<=I<=MIN(ICD-IZ,IG-IZ)> < H(AK)  *>
 20   ICD=ICD+1
      READ(5,201) NCD,MULT,(LR(I,NR),I=1,NSX)
COND  M  22 A:  <* M  20 A *>
COND  M  22 B:  <* M  20 B *>
COND  M  22 D:  <* ONE OF THE SET DI,DII,DIII *>
C          DI:  <* 1<=ICD<=IZ+1, F=.T., SW=.F., NR=1 *>
C         DII:  <* IZ+1<ICD<=IG+1, F=.F., SW=.F., NR=1 *>
C        DIII:  <* IG+1<ICD<=IR+1, F=.F., SW=.T., NR=ICD-IG *>
COND  M  22 E:  <* <U(I),0<=I<=MIN(ICD-IZ-1,IG-IZ)> < H(AK)  *>
COND  M  22 F:  <* NCD=INPUT(1,ICD), MULT=INPUT(2,ICD)   *>
COND  M  22 G:  <* LR(I,NR)=INPUT(I+2,ICD), 1<=I<=NSX *>
 22   IF ( NCD .EQ. 0 )  GO TO 60
      IF (F)  WRITE(6,240)
      F = .FALSE.
      I=1
```

```
COND  M  26 A:  <* M  20 A *>
COND  M  26 B:  <* M  20 B *>
COND  M  26 C:  <* 1<=I<=NSX  *>
COND  M  26 D:  <* ONE OF THE SET DII,DIII *>
C           DII:  <* IZ<ICD<=IG+1, F=.F., SW=.F., NB=1 *>
C          DIII:  <* IG+1<ICD<=IR, F=.F., SW=.T., NB=ICD-IG *>
COND  M  26 E:  <* M  22 E *>
COND  M  26 F:  <* M  22 F *>
COND  M  26 G:  <* M  22 G, 1<=|LR(II,NR)|<=NJ, 1<=II<I *>
 26     IF ( LR(I,NR) .EQ. 0 )  GO TO 30
        NJ = MAX0( NJ, IABS( LR(I,NR) )  )
        I=I+1
        IF  (I.LE.NSX)  GO TO 26
        I = NSX + 1
COND  M  30 A:  <* M  26 A,B,D,E,F,G *>
COND  M  30 B:  <* 1<I<=NSX+1 *>
 30     LS(NR) = I - 1
        LF(NR) = MULT * LS(NR)
        WRITE(6,301) NCD,MULT,(LR(J-1,NR),J=2,I)
COND  M  30.3  A:  <* GC(NR), KN=0 *>
COND  M  30.3  B:  <* K, TD=0 *>
COND  M  30.3  C:  <* NCD=INPUT(1,ICD)  *>
COND  M  30.3 D:  <*  M  26 D  *>
COND  M  30.3  E:  <* M  22 E *>
        IF ( NCD .EQ. 2 ) GO TO 40
COND  M  30.4  A:  <* M  30.3  A *>
COND  M  30.4  B:  <* M  30.3  B *>
COND  M  30.4  C:  <*  M  22  E  *>
COND  M  30.4  D:  <* M 26 DII *>
        CALL APPLY(1,1)
```

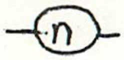
```
      IF  (N.GT.NMAX)  GO TO 9999
      GO TO 20
COND  M  40 A:  <* M  30.3 A *>
COND  M  40 B:  <* M  30.3 B *>
COND  M  40 C:  <*  M  22 E  *>
COND  M  40 D:  <*  M  26 D, NCD=2  *>
 40   SW = .TRUE.
      NR = NR + 1
      GO TO 20
COND  M  60 A:  <* M  20 A *>
COND  M  60 B:  <* M  20 B *>
COND  M  60 C:  <* M  22 E *>
COND  M  60 D:  <* ONE OF THE SET DI, DII *>
C          DI:  <* 1<=ICD<=IZ, NR=1, F=.T., SW=.F.  *>
C         DII:  <* ICD=IR+1, F=.F., NR=IR-IG+1 *>
 60   IF (F)  GO TO 20
      LS(NR) = 4*NJ
      LF(NR) = 4*NJ
     I=1
COND  M  66 A:  <* M  60, DII *>
COND  M  66 B:  <* 1<=I<=NJ *>
COND  M  66 C:  <* LS(NR)=LF(NR)=4*NJ *>
COND  M  66 D:  <* LR(K+4*II,NR)=II IF K=0 OR -3
C                         =-II IF K=-1 OR -2, 1<=II<I  *>
 66   LR(4*I-3,NR) = I
      LR(4*I-2,NR) = -I
      LR(4*I-1,NR) = -I
      LR(4*I,NR) = I
      I=I+1
     IF  (I.LE.NJ)  GO TO 66
```

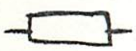
```
      LWD=0
      NCOS=1
C  LET H0 = <U(I), 0<=I<=IG-IZ>
COND  M  70 A:  <* GC(NR), KN<N *>
COND  M  70 B:  <* K, ID=0 *>
COND  M  70 C:  <*  H(KN,AK) < H(AK)  *>
COND  M  70  D:  <*  THERE EXISTS P, 1<=P<=KN+NCOS SUCH THAT
C                (1,W,P) < KF'+KR' FOR ALL W OF LENGTH
C                LESS THAN OR EQUAL TO LWD  *>
 70   KN = KN + 1
      I=1
      NCOS=NCOS-1
      IF  (NCOS.GE.0)  GO TO 76
      LWD=LWD+1
      NCOS=N-KN
COND  M  76 A:  <* GC(NR), 1<=KN<=N *>
COND  M  76 B:  <* M  70 B *>
COND  M  76 C:  <* 1<=I<=NP *>
COND  M  76 D:  <*  H(KN-1,AK)+<W(1,KN)W(L)W(KN,1), 1<=L<I >
C                 <  H(AK)  *>
COND  M  76 E:  <*  M  70 D  *>
 76   IF  (KV(KN)).NE.KN)  GO TO 75
      IF ( N .GT. NMAX - LF(NR) )  CALL CONSOL
      CALL  APPLY  (KN,I)
      IF ( N .GT. NMAX )  GO TO 9999
COND  M  75 A:  <* M  76 A *>
COND  M  75 B:  <* M  76 B *>
COND  M  75 C:  <* M  76 C *>
COND  M  75 D:  <*  H(KN-1,AK)+<W(1,KN)W(L)W(KN,1), 1<=L<=I >
C                 <  H(AK)  *>
```

```
      COND  M  75 E:  <*  M  70 D  *>
       75     I=I+1
              IF (I.LE.NR)  GO TO 76
              IF ( KN .LT. N )  GO TO 70
      COND  M  75.3 A:  <* GC(NR), KN=N *>
      COND  M  75.3 B:  <* M  70 B *>
      COND  M  75.3 C:  <*  H(N,AK) < H(AK)  *>
              CALL CONSOL
      COND  M  9999:  <* N=|F:H| OR N=NMAX+1 *>
       9999  WRITE(6,751) N
```

KEY TO FIGURES 1-4

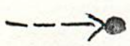
 unlabeled statement

 statement labeled n

 statement with precondition

 statement in sef of sufficiency, Q

 statement with precondition and in Q.

 entry point

 statement in another subgraph.