AD 748225

D D C
RECEIVED
SEP 14 1972
B

APPLIED DATA RESEARCH, INC.

SEE AD737260

APPLIED DATA RESEARCH, INC.
LAKESIDE OFFICE PARK • WAKEFIELD, MASSACHUSETTS 01880 • (617) 245-9540

FIFTH SEMI-ANNUAL TECHNICAL REPORT
(14 January 1972 - 13 July 1972)
FOR THE PROJECT
COMPILER DESIGN FOR THE ILLIAC IV

VOLUME I

Principal Investigator and Project Leader:

Robert E. Millstein	Phone (617) 245-9540
ARPA Order Number	ARPA 1554
Program Code Number	0D30
Contractor:	Applied Data Research, Inc.
Contract No.:	DAHC04 70 C 0023
Effective Date:	13 January 1970
Amount:	$916,712.50

Sponsored by
Advanced Research Projects Agency
ARPA Order No. 1554

APPLIED DATA RESEARCH, INC.
LAKESIDE OFFICE PARK • WAKEFIELD, MASSACHUSETTS 01880 • (617) 245-9540

FIFTH SEMI-ANNUAL TECHNICAL REPORT
(14 January 1972 - 13 July 1972)
FOR THE PROJECT
COMPILER DESIGN FOR THE ILLIAC IV

VOLUME I

CADD-7208-1411

Principal Investigator and Project Leader:

Robert E. Millstein Phone (617) 245-9540

TABLE OF CONTENTS

VOLUME I

PARALLELISM ANALYZER AND SYNTHESIZER

1. Introduction

1.1 Review

Since the previous semi-annual report [8], the Parallelism Analyzer and Synthesizer (abbreviated "the Paralyzer" in the sequel) has grown to a first stage of completion. The current implementation is called the Phase I Paralyzer. It is now possible to carry example programs through the entire analysis and rewriting process (called "paralysis" or "paralyzing").

There has been a change in the environment of the Paralyzer from that predicted previously. It had been planned that the Paralyzer would remain an independent program, apart from development of the rest of the compiler, until late in the year. In the intervening time, however, coding and debugging of the Parse and Transcriber phases of the compiler has been so rapid that these have become quite useful for the Paralyzer. The current configuration of the compiler allows the Parse phase to produce input for the Paralyzer from standard FORTRAN source files. The Paralyzer calls the Transcriber as a utility subroutine to produce a listing, in the IVTRAN language, of the results of its work. The implementation of a versatile overlaying loader and debugging package has also contributed to the early integration of the Paralyzer and the rest of the compiler.

1.2 Summary

The following Sections are intended to give the reader a general technical knowledge of the implementation of the Phase I Paralyzer. The theoretical basis for its development will not be described as it has been previously reported. (It is recommended that a reader unfamiliar with the Parallelism Detection work of L. Lamport, first read reference [3] as this is the most accessible introduction to this work as yet printed. A more complete rendering of the theory from that viewpoint can be found in [2]. The Phase I Paralyzer is actually implemented along the lines of [1], which is the oldest and perhaps most difficult of these references. The IVTRAN language and other parts of the compiler are described in references [4] and [5].) Section 2 describes in functional terms the paralyzing process as currently implemented. Section 3 contains some examples of actual computer output using the Parse, Phase I Paralyzer and the Transcriber. Section 4 discusses future implementation plans.

2. Phase I Paralyzer

2.1 General Description

The Paralyzer can be viewed as a separate "pass" of the compiler or as a very involved optimization step. It is called within the compiler after the completion of Parsing of the entire source program. Its principal input is the Intermediate Language tables built by Parse to represent the source. The Paralyzer rewrites these tables for sections of the program containing DO loops. The general objective is to introduce DOFORALL statements to replace one or more DO statements. In the process, statements and variables may be removed and new arrays and statements may be introduced. The order of the statements in the loop body may be changed. In all cases, for successful rewritings, the transformed loop will compute the same values as the original loop.

The Intermediate Language tables principally used by the Paralyzer are:

- CTAB - Computation Table: the tree of operators and operands of the source code,
- STAB - Symbol Table: principal attributes of identifiers (e.g., dimensionality for arrays),
- ETAB - Extension Table ("Dimension" entries): extent of dimensions and allocation specifications for arrays,
- KTAB - Constant Table: values of constants
- LTAB - Label Table: values of statement numbers and location of definition in CTAB.

Elements of each of these may be modified during the paralyzing process.

The Paralyzer extracts details from the Intermediate Language tables for its own convenience. This set of information is kept in the Paralyzer Tables. These are:

DO Input tables (DI and DX) - cite the location of the nest being paralyzed with respect to the CTAB and provide access to pieces of DO and DOFORALL statements unlinked from the main CTAB chains,

Loop Body tables (LB and QU) - provide convenient access to the CTAB for loop body statements assumed potentially to be composed of a Logical IF statement with an Arithmetic Assignment statement. (The reordering, insertion, and deletion of statements is done with this table.),

Array Reference tables (AR, PU, OC and SS) - describe the location of array references (sometimes called occurrences) in the loop body statements and tabulate the forms of the subscript expressions involved,

<f, g > Set tables (FG and AS) - contain descriptions of the <f, g > sets needed to determine the rewriting,

Candidate DOFORALL tables (CD, DS and US) - a tabulation, in convenient order, of all the combinations of DO statement indices to be tried as DOFORALL control multi-indices, as well as measures of the "quality" of each, computed as they are tried. (The CD table entries are sometimes called "DOFORALL sets"), Matrix tables (MD and MA) - storage for bit matrices defining binary relations among such things as statements (e.g., flow or connectivity) and array occurrences (e.g., data dependency precedence or orderings),

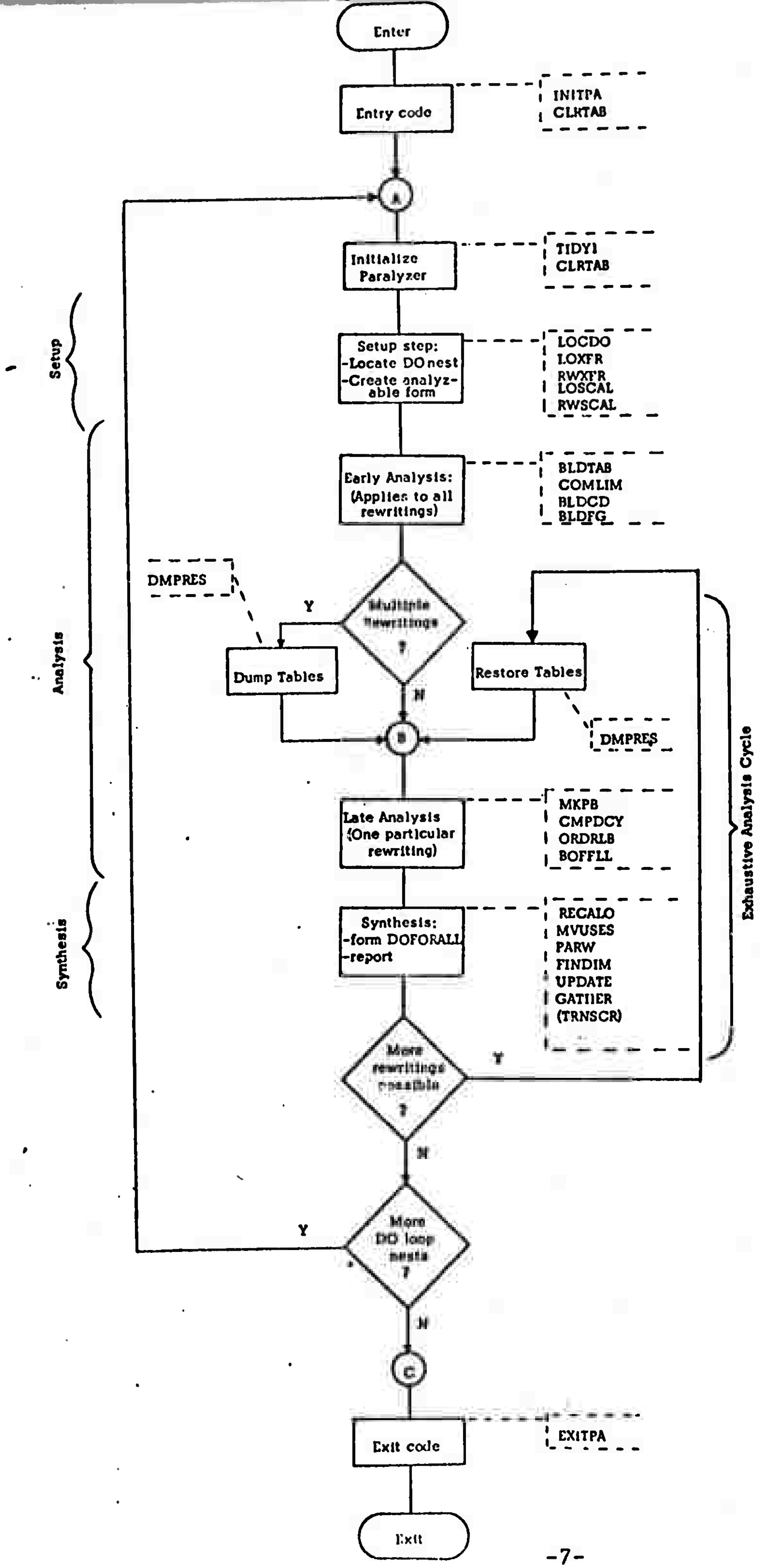
DO/DOFORALL Output table (DO) - convenient access to pieces of DO and DOFORALL statements in the CTAB created just prior to a relinking of the new elements into the CTAB.

There are a number of smaller tables used internally by the Paralyzer such as the Paralyzer Global table (PG) providing general storage for various quantities and access to other tables, and the Table of Tables (TT) describing the sizes, for initialization purposes, of the other tables. Storage is also provided for the various utility routines and debugging aids.

It should be noted that both the Intermediate Language and Paralyzer tables are maintained entirely within core in the current implementation of the compiler. Each Intermediate Language table is accessed via load and store functions which provide core management and packed data facilities for the FORTRAN programs of the compiler (see [4], Chapter IV for more details). The Paralyzer tables are almost all fixed length collections of arrays. Each logical group of tables is defined as a separate COMMON block of the Paralyzer.

The overall flow of control in the Phase I Paralyzer is described by the following flowchart. Only the high level routines have been named and many details of control flow have been simplified. In particular, the chart represents the normal flow described for a successful rewriting attempt: neither error handling nor interaction for debugging purposes has been shown. The actual structure of the routines used is more nearly tree-like. The overlay loader, supporting the entire compiler, maintains a core image of the routines needed for any step in the process.

The general process used for paralyzing is based on the "Complete Coordinate Method" (with "Consistent Orderings") described in [1]. The Setup steps address themselves to the problem of locating a nest of DO loops and enforcing, by some ad hoc rewriting techniques, the restrictions on the loop imposed by the method. These restrictions force the movement of all Arithmetic Assignment statements into the deepest level DO loop of the nest ("tight nesting") and the elimination of all explicit transfers of control and generations of (stores into) scalar variables. The Analysis steps verify the remainder of the restrictions on the nest constituents and compute all the necessary data for: (1) determining if a rewriting is legal, and (2) specifying the details of the rewriting. The Synthesis steps take the data of the Analysis steps and create the rewritten code sequences and array allocation specifications. The Phase I Paralyzer performs an Exhaustive Analysis of all the possible rewritings for any given DO nest. This approach has been taken, rather than the selection of some particular rewriting, because Phase I is intended as a tool for the design of Phase II. More discussion of this point is in Section 4. A more detailed description of the routines named in the flow-chart follows in Section 2.2.

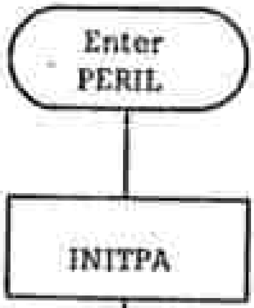

Enter
Entry code
INITPA
CLRTAB
A
Initialize Paralyzer
TIDY1
CLRTAB
Setup
Setup step:
-Locate DO nest
-Create analyzable form
LOCDO
LOXFR
RWXFR
LOSCAL
RWSCAL
Analysis
Early Analysis:
(Applies to all rewritings)
BLDTAB
COMLIM
BLDCD
BLDFG
DMPRES
Y
Multiple Rewritings ?
Dump Tables
Restore Tables
N
B
DMPRES
Late Analysis
(One particular rewriting)
MKPB
CMPDCY
ORDRLB
BOFFLL
Synthesis
Synthesis:
-form DOFORALL
-report
RECALO
MVUSES
PARW
FINDIM
UPDATE
GATHER
(TRNSCR)
Exhaustive Analysis Cycle
More rewritings possible ?
Y
N
More DO loop nests ?
Y
N
C
Exit code
EXITPA
Exit

2.2 Description of Major Processes

Each of the processes named in the flowchart of the previous section is described by the following charts. Together they comprise a highly annotated flow diagram of the Paralyzer control routine, PERIL. Each of these processes has been coded as a FORTRAN subroutine subprogram for ease of design and implementation.

The named routines are called only at one point each from PERIL. The descriptions encompass code both in the high level routines and also any of their utility routines. The descriptions are general but are intended to give a good overall description of the Phase I Paralyzer.

Apart from the flow-chart-like symbols used, the conventions for the charts are as follows. The "Input" and "Output Data" for each routine is described,motivating the processes performed. The "Function" of each routine is sketched out briefly in terms of transformation or creation of this data. A very few distinguished labels for transfers of control within PERIL are noted by the label appearing within a circle. The designations "Post (n)", appearing within oval connectors, denote trace, report, and interaction points within the overall flow. The numeric value corresponds to the "trace number" which appears on an example in Section 3. As can be seen, the reporting and interaction mechanisms have not been detailed. Departures of control away from the main flow have been documented with annotation. These "Abnormal Flow of Control" transfers are actually worked out at the locus of the "post" points.



Input Data

Compiler Global table:

- option bits set when source file name was entered
- Parse phase syntactic error flag;

Intermediate Language tables:

- initial state of Symbol tables;

Teletype input (if not "production" mode):

- Paralyzer I/O control flags (output device, interaction level, "linearized" vs. intermediate language form of debug dumps)
- high level Paralysis control (whether or not to attempt "backup", whether or not to work on successive DO nests, identification of a particular rewriting attempt to try).

Output Data

Paralyzer Global (PG) table:

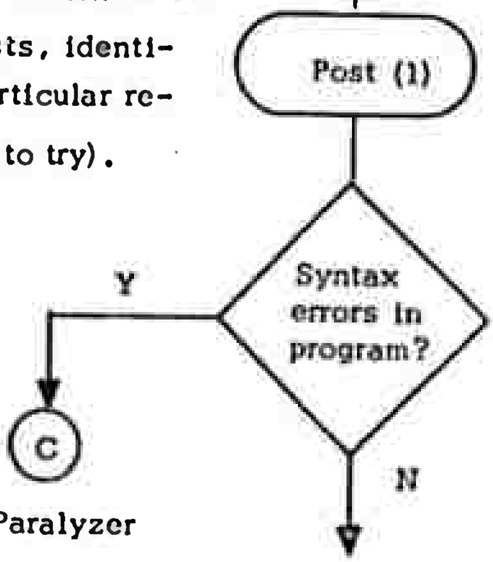
- initial values for various data
- recording of "control card" data from teletype;

Debugging (UG) and String handling (SG) tables:

- initial values for debug and string packages.

Functions

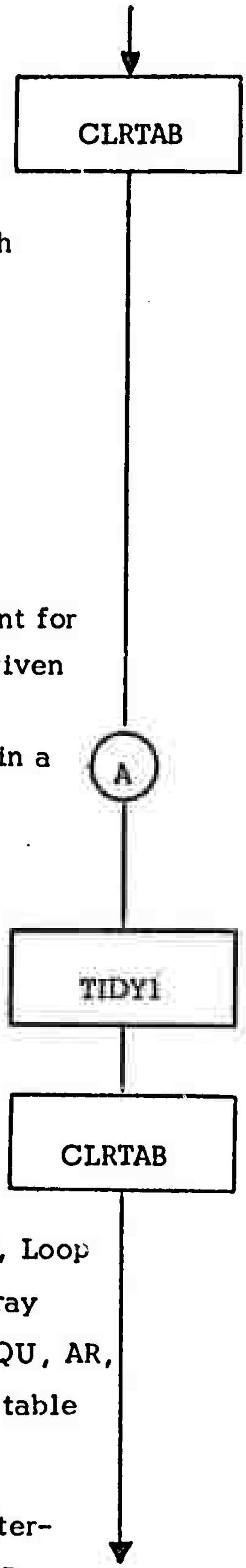
Initializes the Paralyzer data base. If not in "production" mode, reads "control card" information from user teletype and adjusts various debugging aid parameters.



Input Data

Paralyzer Table of Tables (TT):

- sizes of all fixed length Paralyzer tables.



Output Data

All Paralyzer data base tables except the Table of Tables:

- all entries are cleared to initial values except for selected entries in the Paralyzer Global (PG) table initialized previously.

Function

Establishment of fixed initial values for all working tables (usually value Ø) prior to any paralysis attempt.

Note: This is the return point for any "backup" attempt on a given DO nest or for any further paralysis of DO nests later in a given program.

TIDY1 Input Data

Intermediate Language table CTAB:

- dangling components of intermediate language from previous paralysis;

DO Input tables (DI and DX), Loop Body tables (LB and QU), Array and Occurrence tables (LB, QU, AR, PU, OC, and SS), DO Output table (DO):

- pointers to dangling, intermediate segments of the CTAB.

TIDY1 Output Data

Intermediate Language CTAB:

- all unused elements are now on the free chain.

Function

Returns space to the CTAB free chain after any full paralysis attempt. If the Paralyzer tables are clean on entry, then no work is needed and control passes quickly. The CLRTAB call after the TIDY1 call finishes the cleanup and re-initialization.

Input Data

Paralyzer Global (PG) table entry PGI is used as an entry and exit parameter to indicate where to look for DO nests in the CTAB on entry to the routine or on any re-entry;
Intermediate Language tables for the original program structure;
Paralyzer Table of Tables indicating maximum sizes of DI, DO and LB tables.

LOCDO

Output Data

Paralyzer Global parameter PGI indicating either end-of-program found, or where in CTAB to attempt a "next paralysis";
Intermediate Language CTAB elements for Logical IF statements produced while forcing a tight nest plus other duplicated elements;
Paralyzer DO-Input (DI) tables describing the original sequential DO statements in the nest;
Paralyzer Loop Body (LB) tables, describing the statements originally found in the nest;
Paralyzer Global table parameters indicating position of nest in Original CTAB and also flagging whether transfers of control or scalar generations were encountered.

Function

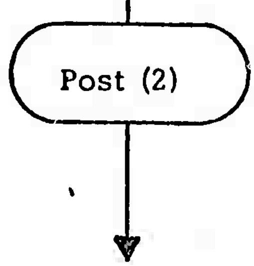
Starting from the "PGI" location in the CTAB, the program is stepped in order of appearance of statements while looking for a DO statement. When one is found, tables are built describing a tight nest of the DO statements with all other statements enclosed. Logical IF CTAB elements are generated as needed for any loop body statements which

Abnormal Flow of Control

If the end of the program has been reached with no nest of DO's created, control will transfer to point C for exit from the Paralyzer. Note that this is the actual method of Paralyzer exit as contrasted with the simplified flow of control described by Section 2.1.

If a tight nest cannot be achieved, the PGI parameter will be adjusted either to attempt an inner DO of the current structure, or some DO later in the program. Control will pass to point A for cleanup and retry.

are forced into the nest. The only loop body statements which are allowed are Arithmetic Assignments, Logical IF's, and GOTO's, the latter only at the deepest nest level. Transfers of control and generation of scalar variables are noted for processing by other routines. If the nest cannot be "tightly nested" within the restrictions, an attempt is made to process the inner DO statements and if this fails, to process DO statements further in the program. (The PGI parameter is initially set to the head of the program by the current Paralyzer Control routine. Thus the overall global strategy of the current analyzer is to start at the beginning of the program and attempt paralysis on each loop in turn with the biggest possible "tight-nest" on each try.)



LOXFR

Input Data

Intermediate Language
CTAB and Label Table elements which describe the original code of the loop body;
Paralyzer Loop Body table as initialized for the statements found while forcing the tight nest;
Paralyzer Table of Tables (TT) for maximum size of OC, AR and LB tables;
Paralyzer DO Input table (DI) for maximum depth of nest.

Output Data

Paralyzer Matrix tables (MD and MA) containing statement connectivity (flow) matrix, "PBB", and its closure, "PBBC", for the statements in the loop body;
Paralyzer LB table with an updated statement chain and with all free GOTO statements deleted;
Paralyzer Global (PG) table parameters for accessing the PBB and PBBC matrices, and also a failure parameter;
Paralyzer AR and OC tables used to pass parameters to the routine RWXFR concerning the flow of control in the loop body.

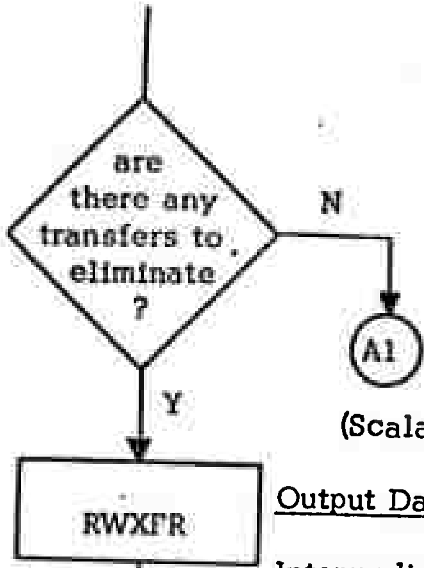
Abnormal Flow of Control

If there are any backward transfers (loops), transfers not at the deepest level in the nest, or transfers out of the scope of one or more DO statements, no transfer elimination is attempted. Depending on the nature of the rejected code, an attempt at paralyzing the inner loops will be made or else this loop will be rejected entirely. Control passes to point A.

Function

Builds in all cases, the matrix of loop body statement connectivity or flow. (This plus the later built tables describing the constituents of the loop body statements serves to define the relation " << " of Lamport [1], Chapter 2, Section ([illegible].) When there are actual transfers of control, these are traced, and, if they are legal forward transfers, are tabled for later elimination.

Post (3)



(Scalars will be checked next.)

Input Data

Intermediate Language CTAB containing the logical expressions of the Logical IF statements and the GOTO statements (soon to be deleted);
Paralyzer LB table describing the current loop body statements;
The Paralyzer AR and OC tables used as parameter storage (pointers to the LB and CTAB) passed from the preceding LOXFR routine;
Paralyzer TT table for maximum size of AR, OC and LB tables.

Output Data

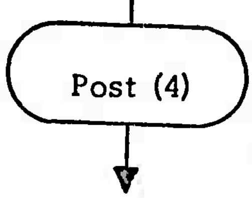
Intermediate Language CTAB:

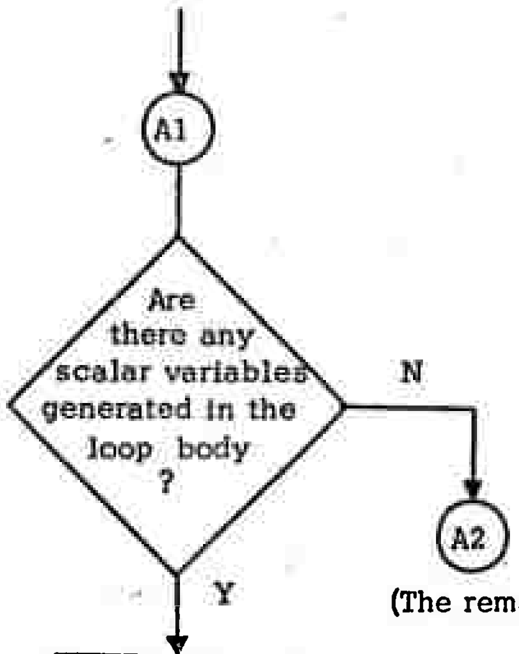
- all GOTO statements in the loop body have been eliminated,
- "compound" logical expressions have been created, and
- new Logical IF statement elements have been created;

The Paralyzer Loop Body (LB) table has been cleaned up, to contain only arithmetic assignment statements or Logical IF statements with arithmetic assignments;
The Paralyzer AR and OC tables used as temporary storage for LOXFR and RWXFR are re-initialized.

Function

Given a tabling of branches and merges of control produced by LOXFR, the explicit transfer statements are eliminated and LB is updated to show Logical IF constructs which are equivalent to the transfers.





(The remaining tables will be built.)

LOSCAL

Input Data

Intermediate Language CTAB and Symbol Tables containing descriptions of the original DO statements and the loop body statements;

Paralyzer DO Input (DI) table for designations of the DO indices and limits;

Paralyzer LB table for access to the CTAB;

Paralyzer Matrix tables containing the statement connectivity matrix;

Paralyzer PG table giving access to the above "PBBC" matrix;

Paralyzer TT table giving the maximum size of the OC table.

Abnormal Flow of Control

If code is present which is beyond the capabilities of this routine, an attempt is made to either "back-up" and try an inner DO loop, or to try following DO

Output Data

Paralyzer Global table parameter indicating whether the process failed or not;

Paralyzer OC table used as temporary storage between the LOSCAL and RWSCAL routines.

Function

All the loop body statements are examined for the generation of scalar variables. A table is built, for the routine RWSCAL, which denotes scalar generations and uses in the loop. The special cases of scalars used as DO limits are noted. Generations and uses which form particular examples of computing a running summation or product are noted. No code is rewritten in this routine.

Post (5)

RWSCAL

Input Data

Intermediate Language CTAB and Symbol tables containing elements of the DO statements and loop body statements before any scalars are eliminated;
Paralyzer LB and DI tables providing access to the above;
Paralyzer OC table as temporary storage giving location of scalar generations and uses;
Paralyzer PG table for access to the "PBBC" matrix;
Paralyzer Matrix tables for above matrix;
Paralyzer TT table for maximum size of OC table.

Abnormal Flow of Control

If the DO limits, after rewriting, do not meet the restrictions of Lamport [1], Chapter 3, an attempt will be made to Paralyze inner loops in the nest or later loops in the program. Control will be transferred to point A above for the "backup" or retry attempt.

Output Data

Intermediate language CTAB and Symb
tables with generations of scalars removed from the loop body;
Paralyzer LB and DI table cleaned up
show new statements and statement forms;
Paralyzer PG table flagging the occurrence of scalar or scalar-sum removal, or whether the process faile
Paralyzer OC table re-initialized.

Function

Using the tabulation of generations a
uses of scalar variables, an attempt
made to substitute the defining expressions in the use occurrences. W
the scalar generation carries DO limi
dependence from the outer to the inn
DO statements, the DO limits are re-
written, if possible, to show the nev
expression form. Where the use immediately follows the generation,
substitution is attempted. If there a
more complicated generation-use structures, new arrays are introduce
Special cases of running sums and products are eliminated where possib
After substitution, the DO limits are

checked for validity within the restrictions. All eliminated CTAB elements are deleted. It should be noted that at this stage, the loop consists of pieces of statements in the CTAB, "held together" by the structure of the DI and LB tables.

Post (6)

(This point marks the end of the Setup steps and the beginning of the Analysis steps.)

A2

BLDTAB

Input Data

Intermediate Language

CTAB, Symbol Table and Constant tables containing elements and descriptions of the loop body statements;

Paralyzer Loop Body (LB) table to access the dangling elements in the CTAB:

Paralyzer Table of Tables (TT) for maximum sizes of all tables being produced.

Abnormal Flow of Control

If there are array references with subscripts which do not meet the restrictions for paralysis, an attempt is made to paralyze with respect to inner loops by transferring control back to point A.

Output Data

Paralyzer PG table containing an indication of success or failure of the process;

Paralyzer Array and Occurrence tables detailing conveniently the form of all array references in the loop body;

Completed Paralyzer LB and DI tables pointing to useful parts of the Array and Occurrence tables.

Function

The LB table description of the current set of statements in the loop body is used to access the CTAB for the statement pieces. A tree walk is performed for each statement, using a stack in the Paralyzer, to search for all array references. For every array reference, entries are made in the Array tables and the Occurrence tables for the name of the array, its dimension, its subscript forms and the canonical ordering of subscripts with respect to the DO indices. Some restrictions on subscript forms are checked.

Post (7)

COMLIM

Input Data

Intermediate Language tables containing the components of the DO statements;
Intermediate Language Symbol tables containing dimension and extent information for all arrays referenced in the loop;
Paralyzer DI table providing access to DO statement elements;
Paralyzer Array and Occurrence tables providing convenient access to the Symbol tables.

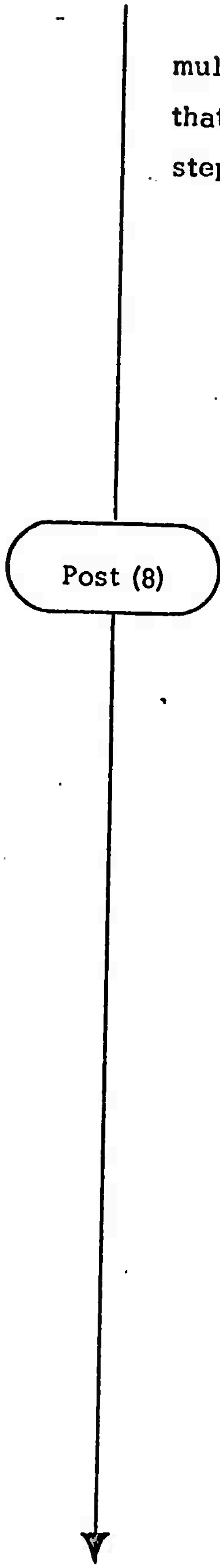
Output Data

Intermediate Language CTAB containing elements of bound expressions for the DO limits;
Paralyzer DI table entries giving upper and lower bounds on the DO statement limits where these are variable, differences between limits, and frequencies.

Function

For DO statement limits which are variable and which might be controlled by outer DO statements in the nest, upper and lower bounds are computed where possible. A greatest lower bound on the difference between variable limits is computed and a least upper bound on the frequency of the DO statement is computed. Where necessary, array references are checked using a strategy based on a knowledge of legal FORTRAN subscripts with respect to dimension information. DO indices for which these values cannot be computed are flagged for later prohibition from inclusion in DOFORALL.

multi-indices. This process is essentially that of Lamport [1], Chapter 6, Section B, steps 1, 2, and 3.



Input Data

DO Input (DI) table:

- depth of nest
- allowability of DO index in a DOFORALL multi-index;

Table of Tables (TT):

- maximum size of CD, DS, US tables

Output Data

Candidate DOFORALL Set tables (CD and DS):

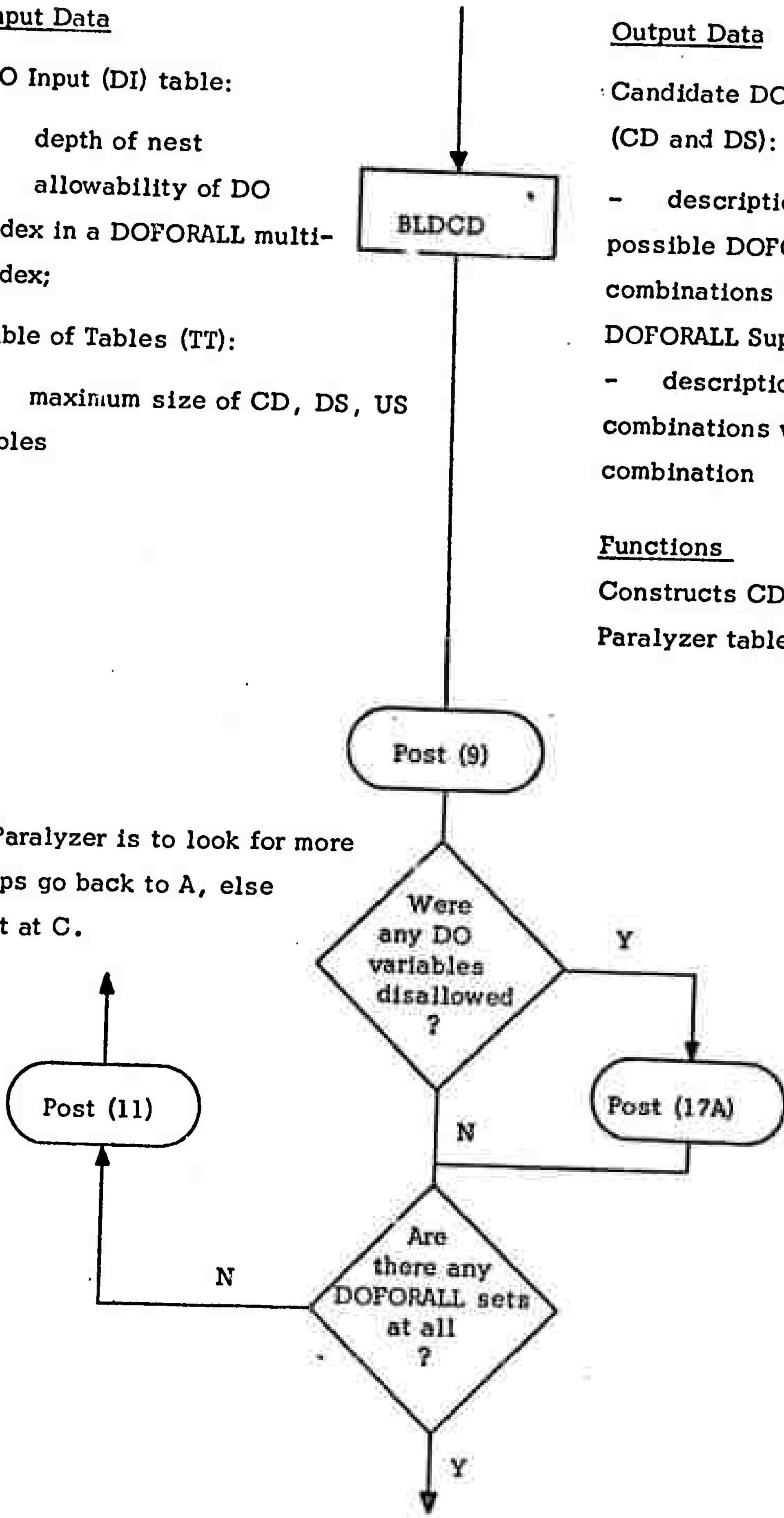
- descriptions of the $2^n - 1$ possible DOFORALL multi-index combinations (the "DOFORALL set

DOFORALL Superset (US) table:

- description of DOFORALL combinations which contain a give combination

Functions

Constructs CD, DS and US Paralyzer tables.



If Paralyzer is to look for more loops go back to A, else exit at C.

This produces a report on any DO indices which were ruled out as constituents of a DOFORALL multi-index.

BLDFG

<u>Input Data</u>

Array (AR) table, Occurrence (OC) table, Subscript (SS) table, Subscript Permutation (PU) table, Loop Body (LB) table, Quantification (QU) table, DO Input (DI) table, and Candidate DOFORALL set tables (CD, DS and US):

- descriptions of all array references in the loop body
- subscript forms for all array references
- descriptions of all potential DOFORALL sets;

Table of Tables (TT):

- maximum sizes of FG and AS tables;

Intermediate Language tables:

- actual subscript expressions
- constant values
- attributes of variable and array names

<u>Output Data</u>

<f,g> set tables (FG and AS):

- constituents of all <f,g> n-tuples related to pairs of occurrences of the same array

CD table:

- flagging of DOFORALL sets which would result in illegal rewriting.

<u>Functions</u>

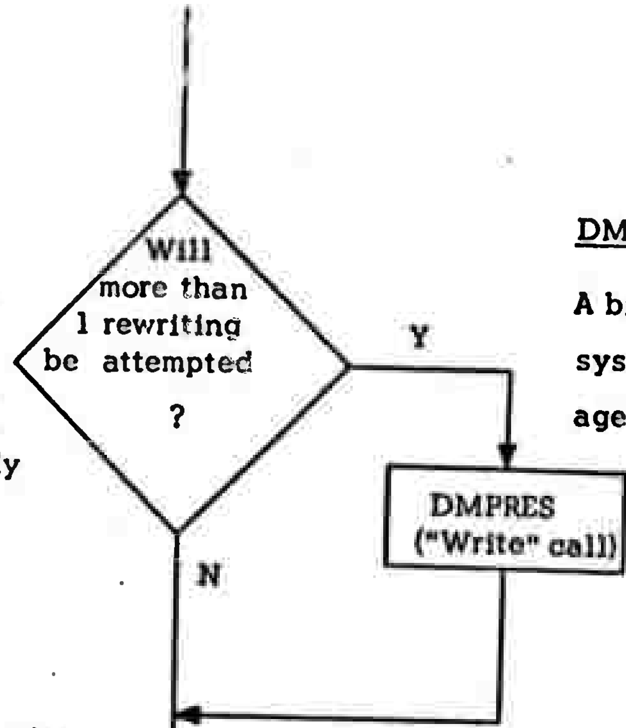
Constructs FG and AS tables for pairs of array references to the same array where at least one reference is a generation.

<f,g> sets are examined as they are constructed for implications relative to the loop rewriting rule of Lamport [1], Chapter 4, Section E, rule 1 ("4E1"). Illegal CD entries are marked (bad DOFORALL combinations are ruled out).

This is where the "midstream Paralysis Report" is generated.

Post (10)

The remaining good DOFORALL combinations are counted. If there are <u>no</u> good sets remaining, consideration of this nest is left as in Post (11), on the second preceding page.



<u>DMPRES Input Data</u>

All Intermediate Language tables; Paralyzer DO Input tables (DI and DX), and Loop Body tables (LB and QU).

<u>DMPRES Output Data</u>

A binary scratch file on system secondary stor-age:

- contents of all data mentioned as input to the routine

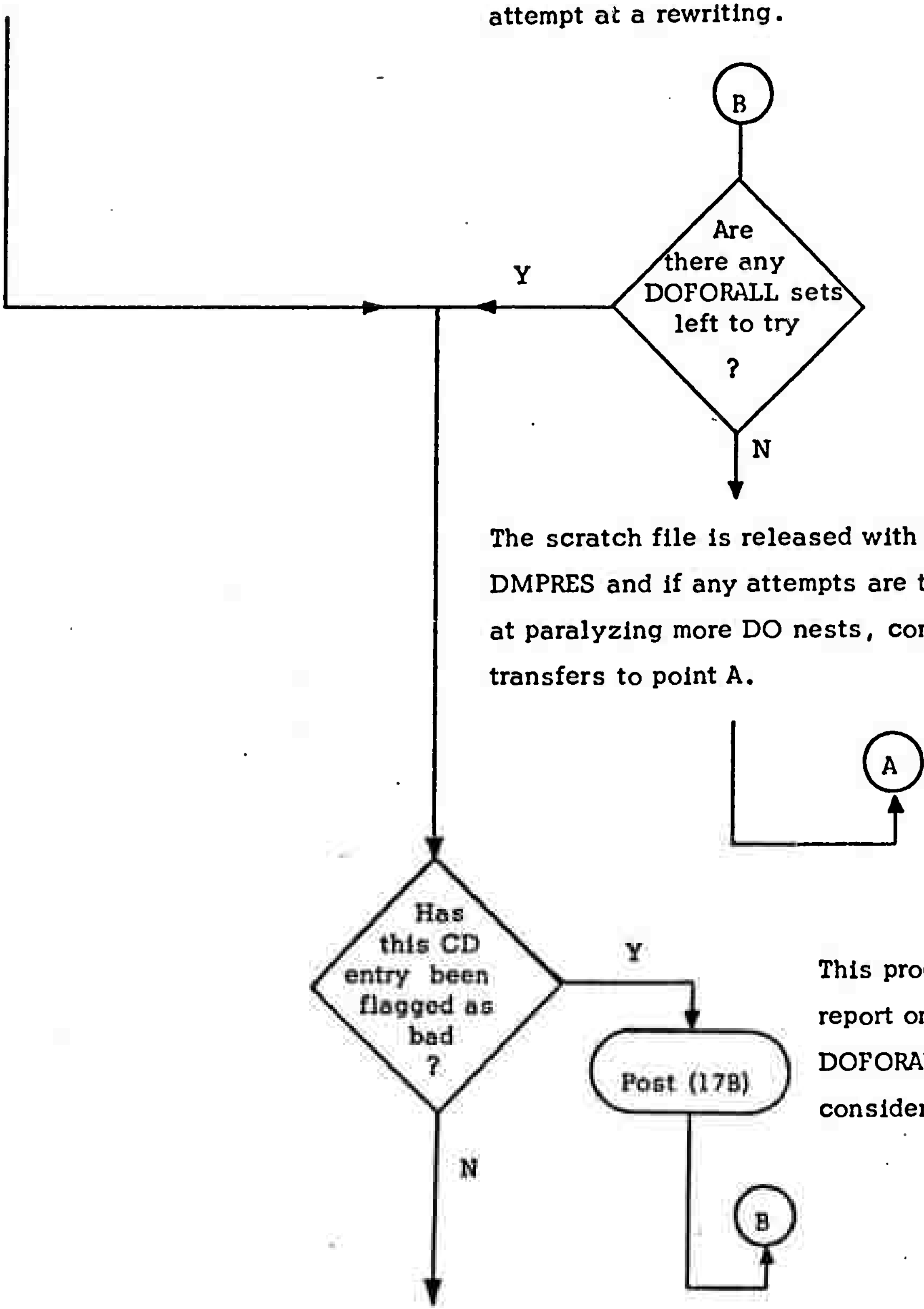
(Note that DMPRES can also be called to "Read" the file or to "Release" it. The distinction is determined by a formal parameter in the call.)

<u>Function</u>

Preservation of the essential tables created by Parse and the modifiable tables created by the Setup and Early Analysis portions of the Paralyzer. These can then be restored after every rewriting prior to the next attempt.

The exhaustive analysis loop is initialized here with a global parameter indicating the line number of the CD Table which describes the current choice of DOFORALL combination. As this loop progresses, successive entries in the CD table will be used.

This is the point of return for the next attempt at a rewriting.



The scratch file is released with a call to DMPRES and if any attempts are to be made at paralyzing more DO nests, control transfers to point A.

This produces a report on why this DOFORALL set is considered bad.

If tables were previously dumped with a DMPRES "Write" call, then they are restored at this point with a DMPRES "Read" call.

MKPB

Input Data

A formal parameter indicating the index number of the CD table for the current DOFORALL set being tried;
Paralyzer tables describing the array occurrences (AR, QU, OC, SS), any quantification of occurrences (QU), and the flow of control from statement to statement in the sequential form of the loop (LB, MD, and MA);
The candidate DOFORALL set (candidate DOFORALL multi-index) described by the CD, DS and US Paralyzer tables;
Paralyzer FG and AS tables supplying the <f,g> sets.

Output Data

A matrix of necessary occurrence orderings (data dependencies) is built in the MA and MD tables. This is known as the "PB" matrix;
The transitive closure of the PB matrix, called the "PBCL" matrix, also stored in the MA and MD Paralyzer tables;
"Quality" information on the DOFORALL set recorded in the CD table;
Some global parameters aiding access of the PB and PBCL matrices, stored in the Paralyzer PG table.

Function

Computes the necessary data dependencies resulting from the current choice of DOFORALL set as determined by the <f,g> sets. What is being computed is the " < " precedence relation between occurrences as described in Lamport [1], Chapter 4, Section E, rules 2 and 3, and also the intra-statement dependencies given by [1], Chapter 7, Section A, step 7 (" < " is the " << " relation of Lamport [2]). As these dependencies are noted, immediate inconsistencies are recognized and statistics are kept in CD table fields.

CMPDCY

Input Data

Formal parameter specifying current DOFORALL set CD table index number;
Paralyzer Global table giving access to the "PBCL" matrix;
Paralyzer OC, MA and MD tables for the "PBCL" matrix data.

Output Data

A CD table field containing a count of "cycles" in the PBCL matrix.

Function

Counts any disjoint "cycles" in the data dependency relation. These arise if the dependencies are inconsistent with respect to an ordering of occurrences for a particular DOFORALL set. If there are any inconsistencies, the current rewriting algorithm is not applicable.

Abnormal Flow of Control

If any inconsistent orderings have been detected by MKPB or CMPDCY, a report is generated as at Post (17B) and control passes back to point "B" for the next try.

Post (12)

ORDRLB

Input Data

Original statement ordering information from the Paralyzer LB table;
Occurrence ordering relations from the transitively closed precedence matrix "PBCL" in the MD and MA tables accessed via the PG table.

Output Data

New ordering of statements noted in the LB table.

Function

With the underlying assumption that there is exactly one generation occurrence per statement, a complete (linear) ordering of the generation occurrences is derived from the precedence matrix for all occurrences. Original statement order is used to complete partial orderings. The new order of statements for the rewritten loop body is reflected in chaining and ordinal number fields in the LB table.

BOFFLL

Input Data

Use and generation occurrence descriptions from OC and AR tables;
New statement ordering from LB table;
Occurrence orderings from "PBCL" matrix in the MA and MD tables, accessed via the PG table.

Output Data

The quantities "first" and "last" of Lamport [1], Chapter 7, Section A, step 12,stored as pointers to the LB table in fields of the OC table for use occurrences.

Function

Computation of the positional constraints on a use occurrence with respect to generations of that array to be used to provide storage and extra statements to prevent overwrite of needed values in the rewritten loop.

Post (13)

(The Synthesis steps start here.)

RECALO

Input Data

Formal parameter specifying CD table line number for the current rewriting attempt;
Intermediate Language symbol table information giving the dimension of arrays;
Paralyzer CD table giving the DOFORALL multi-index, and AR table designating all arrays referenced in the loop body.

Output Data

Intermediate Language symbol table information specifying the required allocation for arrays in the scope of the DOFORALL;
Paralyzer AR and CD table fields indicating that proper allocations were not derivable for some arrays if this is so.

Function

Creates, where possible, an allocation specifier for all arrays in the scope of the DOFORALL such that the control multi-index is an allowable multi-index of the arrays. Where this cannot be done, the arrays with "problems" are noted for later report. Note that any new arrays introduced later in the process will be allocated by FINDIM, below.

MVUSES

Input Data

Formal parameter specifying CD table line number;
Intermediate Language CTAB for the forms of array occurrences to be "moved";
Intermediate Language symbol table for descriptions of arrays being considered;
Paralyzer AR, OC and LB tables designating array uses to be "moved";
Paralyzer CD table for current DOFORALL multi-index;
Paralyzer TT table designating maximum length of OC and LB tables.

Abnormal Flow of Control

If the rewriting has been noted as illegal because of allocation problems in RECALO, a report is generated as at Post (17B) and control passes back to point "B" for the next try.

Output Data

Intermediate Language CTAB entries for new statements and new array occurrence operands;
Intermediate Language symbol table entries for new arrays used as temporary storage;
Paralyzer OC and LB table entries for new statements and array occurrences.

Function

For the new ordering of statements for this particular rewriting, if the data of any use occurrence would be over-written before it was used, a temporary storage array is introduced to hold the value until it is needed. The use occurrence is rewritten to reference the temporary array and an appropriate statement is inserted to preserve the values that will be overwritten. A count is kept in the CD table of the number of uses which are thus "moved" earlier in the loop body.

Post (14)

PARW

<u>Input Data</u>

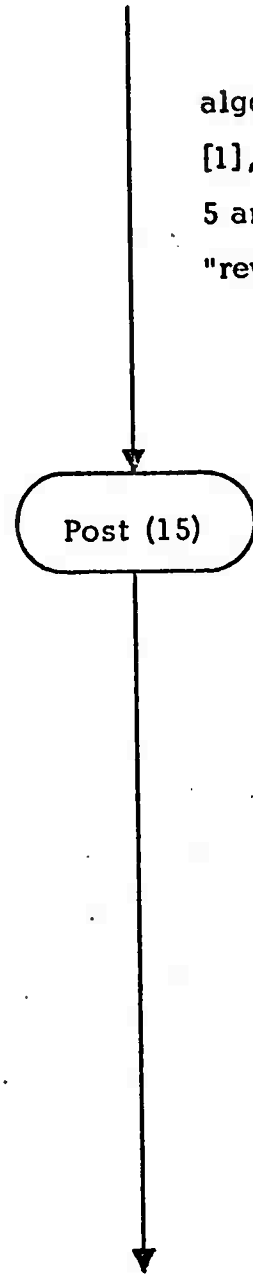
Formal parameter giving CD table line number;
Intermediate Language CTAB and KTAB giving elements of the original DO statements;
Intermediate Language symbol table describing dimension extents for arrays in the loop body;
Paralyzer DI table for original DO statement information;
Paralyzer CD table giving control multi-index for the DOFORALL;
Paralyzer LB table giving access to the CTAB for array occurrence subscript forms;
Paralyzer TT table for maximum size of Paralyzer DO Output (DO) table.

<u>Output Data</u>

Intermediate Language CTAB and KTAB entries for DOFORALL statement components and modified array occurrences;
Intermediate Language symbol table entries for new logical arrays;
Paralyzer DI table chains for DO and DOFORALL indices and updated DO statement components;
Paralyzer DO table giving pointers to the CTAB for the pieces of the final DOFORALL statement and associated new statements.

<u>Function</u>

From the chosen DOFORALL set specification, pieces of the final DOFORALL statement are constructed in convenient form. A best choice of set constant values is derived using DO limits, when known, and dimension and subscript form information from array references. If useful, subscripts are offset by an integer such that the lowest value of the set constant is one. Sequential DO limits may be replaced by upper and lower bound values. These algorithms are an extraction of Lamport [1], Chapter 6, Section B, steps 4 and 5 and are best described generally as "rewriting the DO statements."



FINDIM

Input Data

Formal parameter giving CD table line number;
Intermediate Language CTAB and symbol tables for array references;
Paralyzer CD table denoting indices in the chosen DOFORALL set;
Paralyzer DI table for final limit values on DO and DOFORALL indices;
Paralyzer AR table describing arrays referenced in the loop body.

Output Data

Intermediate Language CTAB and symbol tables containing minimal dimension and extent array references for introduced arrays and final allocation specifications for these arrays;
Paralyzer PG table entries containing statistics on the amount of new array storage introduced by the complete paralyzing process.

Function

Complete dimensioning and specifying of allocation for arrays introduced in the set-up phases of the Paralyzer. Non-DOFORALL subscripts are dropped where possible and dimensions are derived from the set constant and DO limit values now available in the process. Statistics are gathered on how much extra storage was introduced above that declared by the programmer, because of scalar variable removal, control transfer elimination, and overwrite elimination.

UPDATE

Input Data

Intermediate Language CTAB containing pieces of statements constructed by the Paralyzer, plus the original program structure;
Paralyzer DI and DO tables with pointers at CTAB elements for DO statement and DOFORALL statement constituents, plus ordering information for these;
Paralyzer LB table for new order of loop body statements;
Paralyzer PG table for pointers to statements to be chained into program before and after the rewritten loop.

Output Data

Intermediate Language CTAB containing the new version of the loop body (the rewritten version) in proper form;
Intermediate Language Label table containing entries for new labels generated by the Paralyzer.

Function

All the dangling elements of the rewriting attempt are now collected together and chained into the user's program replacing the original code. In particular, the DOFORALL and rewritten DO statements are created and chained together, the loop body statements are chained into their new ordering, Logical IF's introduced by transfer elimination are linked to their assignment statements and newly introduced code is chained-in ahead of and after the rewritten loop. The old code is deleted from the program by returning CTAB elements to the free chain.

GATHER

Input Data

Formal parameter giving CD table line number for this rewriting;
All Paralyzer tables which may contribute to statistic gathering.

Output Data

Fields in any Paralyzer tables containing any statistics gathered at the last minute.

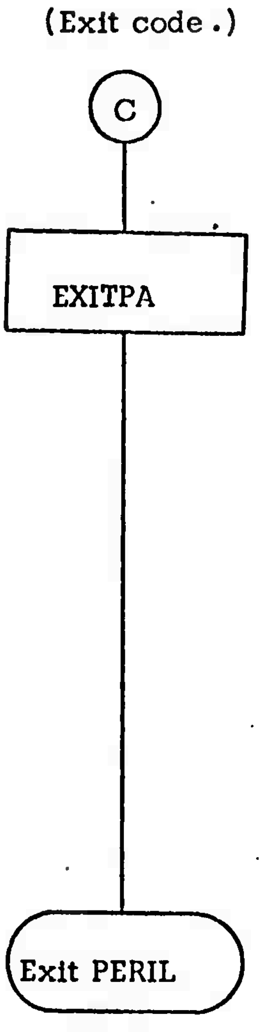
Function

This is a catch-all routine which can prepare any information needed for reporting purposes at a point in the process following all significant activity. The current code is quite trivial, only serving to prepare DO loop frequency counts for final reporting.

Post (16)

At this point, the Transcriber is called to turn the rewritten loop from Intermediate Language form into a humanly readable form in the IVTRAN language. This can be printed as a report of the Paralyzer activity. The Paralyzer also creates a report at this point summarizing such properties of the rewriting as: how much new storage was introduced, how many sequential iterations the new version will take, etc.

(B) The next tabulated DOFORALL set will be tried.



Input Data

(None)

Output Data

(None)

Function

Provides a point at which all resources used by the Paralyzer can be released back to the compiler. At the moment, all the routine does is to output a message that the Paralyzer is about to be exited.

The entire program has now been Paralyzed "as much as possible". Later compiler phases of optimization and code selection will work on the Intermediate Language tables transformed by the Paralyzer.

3. Examples

Computer output from a number of Parse/Paralyzer/Transcriber sample runs is included in this section. These examples were created to demonstrate the current capabilities of the Paralyzer and have been divided into the following categories: Simple, Bounds Creation, Transfer Removal, Scalar Removal, Tight Nesting, Data Dependencies, Miscellaneous, and Special.

3.1 Description of Examples

Category 01 (Simple). The one example in this set has been included more to introduce the format of the output than to demonstrate any outstanding capability of the Paralyzer.

Category 02 (Bounds Creation). There are two examples, but a total of thirteen discrete DO loops, in this set. The forms of the initial, terminal, and incrementation parameters of the FORTRAN DO statement have been varied (e.g., unknown initial parameter and/or unknown terminal parameter, negative incrementation parameter, etc.)

Techniques for establishing least-upper-bounds and greatest-lower-bounds for DO index variables are demonstrated. These bounds may be a function of the dimensioning and the accessing subscripts for those subscript positions that depend on the associated DO variables as well as the original DO parameters if known at compile time. The DO variables within the loop body may be offset (e.g.,"I" replaced by "I + 5") to create a left-hand-end-set constant value of 1. One reason the offsetting is done is to minimize allocation of set constants and temporary array storage.

Category 03 (Transfer Removal). One example demonstrating removal of forward transfers is included. Backward transfers as well as certain types of forward transfers are not permitted in the present implementation.

Category 04 (Scalar Removal). Three examples are included to demonstrate the removal of scalar generations (i.e., scalar variables to the left of the equal sign in an assignment statement) within the loop body.

The three special scalar generation forms of (a) summation, (b) finding products, and (c) computing DO statement parameters are given special attention. Case PA04C is of special interest since, in addition to scalar removal, it uses the bounds creation techniques introduced in Category 02 in an even more interesting fashion since DO parameters of inner DO statements depend on DO variables of outer DO statements.

Category 05 (Tight Nesting). A procedure called "tight-nesting" is invoked by the current Paralyzer if the dimension of the index set (i.e., the number of DO statements in the nest) is greater than one and there are statements between the DO statements and/or between the loop-closing statements. Such statements outside the main body are "brought-into" the main body, if possible, and an appropriate Logical IF expression, (a function of the initial parameters or terminal parameters, depending on fore or aft) is attached. This set of examples features the tight nesting facility. Paralyzer warning remarks (i.e., TIGHT NESTING INTERFERENCE) are made to indicate that certain conflicts between tight-nesting and allocation have not been totally resolved.

Category 06 (Data Dependencies). The largest number of examples included in this document belong to this category. Data dependencies may exist in certain examples outside this category. However, these outside examples are, with respect to data dependency, not very interesting. More subtle data dependencies are introduced here. Some examples need temporary arrays before parallelism can be achieved; in others a re-ordering of the statements is required; some examples need both. For index sets with dimension greater than one (i.e., more than one original DO statement associated with the loop body) results for each index subset is reported on. Parallelism can be accomplished operating over some of the sets but not all. Case PA06G is completely "un-paralyzable" using the techniques of the current implementation. The Hyperplane Method [3], when implemented, would produce parallelism for the type of iteration in case PA06G.

Category 07 (Miscellaneous). There are two examples in this set. The second (PA07B) indicates the type of warnings released by the Paralyzer when potential allocation problems may exist. The first (PA07A) shows the "outer-to-inner" movement of the Paralyzer in respect to DO statements when troubles arise over the whole nest.

Special Examples. One example to demonstrate the interactive debugging facilities of the Paralyzer has been included. This example, called PAOPT, appears last in the form of teletype output. Trace points referred to in Section 2 of this document are shown on the output for this example.

3.2 Format of Output

Page 1 of each example is a listing of the original FORTRAN program unit with sequence numbers appended on the left side of the page. This listing is an output of the Parse phase of the compiler. If there had been syntactic errors (there are none in the examples included here), diagnostic comments would have been attached to this output set.

A MIDSTREAM PARALYSIS REPORT appears for each discrete DO nest encountered that has successfully met the requirement of the Early Analysis stages of the Paralyzer. The output to this report defines the DO nest and divides the potential DOFORALL subsets into two categories (a) STILL GOOD and (b) "4E1" BAD. (Note: 4E1 is an analysis step which can detect lack of parallelism at an early stage. See description of routine BLDFG in Section 2.)

In the output for category 05, where data dependencies are of utmost interest, a definition of the <f,g> sets [3] and a list of the <f,g> values is attached following the MIDSTREAM PARALYSIS REPORT. This <f,g> output lists the generation/generation and generation/use pairs which must be carefully analyzed for data dependency significance. An output string of the form:

$$<F, G> = <X (I + 1) / 8, X (I - 1) / 14>$$

is to be read:

> "the pair defined by the array element use X (I + 1) on statement with sequence number 8 and the generation X (I - 1) on statement with sequence number 14".

For each index subset either (a) a Transcriber output of the paralyzed program coupled with STATISTICS OF INTEREST, or (b) reasons for rejection are supplied. There is some slight variation in the output for examples PA02A and PA02B where there are several discrete DO loops within each program unit example. In these cases, one composite transcriber output of the entire paralyzed program unit is listed on the last page. For these two cases, the MIDSTREAM PARALYSIS REPORT is coupled with the STATISTICS OF INTEREST report on a single page.

For all other cases, Transcriber output is coupled with STATISTICS OF INTEREST and reported for each paralyzable DOFORALL index subset. An identifying "Zn" string is displayed above Transcriber output (as well as above rejection remarks). This string identifies the DOFORALL set: Z13 means multi-index composed of "first" and "third" DO variables of a full index set of dimension at least three.

The STATISTICS OF INTEREST report is, for the most part, self-explanatory. The unit of frequency is one FORTRAN statement, where compound statements (i.e., those quantified by the Logical IF) count as one.

Reasons for rejection are reported if paralysis cannot be effected. Most rejections are because of cyclic data dependencies. (See [1] for a discussion of "Inconsistent Orderings".)

3.3 Output of Special Example

The listing for the special example is from the teletype since the purpose of the example was to demonstrate the instructive and debugging features of the Paralyzer. The following is a "tour" through the example via trace points:

TRACE 1. The Parse phase is over; the Paralyzer is about to begin. Input codes 65 and 55 produce a Transcriber listing and a macro dump of the entire program unit at this point, respectively.

TRACE 2. The DO nest has been located and has passed preliminary inspection. Input codes 60 and 63 produce formatted dumps of the DI (DO Input) and LB (Loop Body) Tables of the Paralyzer, respectively.

TRACE 3. The flow of the loop body has been established. Input codes 51 and 54 produce dumps of the statement connectivity ("P << " or "PBB" matrix) and its transitive closure ("P << Close" or "PBBC"), respectively.

Note: Trace points 4, 5, and 6 are not encountered since no forward transfers or scalar generations exist.

TRACE 7. The loop body has been inspected and array references tabled and verified. Input codes 58 and 64 produce formatted dumps of the AR (Array) and OC (Array Occurrences) Tables of the Paralyzer, respectively.

TRACE 8. An exhaustive analysis of the DO parameters (i.e., initial, terminal, and incrementation parameters) have been made. Input code 60 is entered to produce a copy of the DI (DO Input) Table which has been updated at this point.

TRACE 9. A table of candidate DOFORALL index sets has been created. Input code 59 produces a formatted dump of the CD (Candidate for DOFORALL) Paralyzer table.

TRACE 10. A copy of the < f,g > sets appears in the same format as the output to the examples of Category 06.

TRACE 12. A precedence matrix ("P <" or "PB") and its transitive closure operating over the array occurrences within the loop body has been created. Input codes 52, 53 yield a dump of P < and its closure, respectively.

TRACE 15. All the basic components for a re-writing of the loop body have been created. However, they have not replaced the old loop nor have they been completely linked. Input codes 61 and 63 produce formatted dumps of the DO (DO output) and the updated LB (Loop Body) Tables, respectively.

TRACE 16. The original program unit has now been rewritten to include the new paralyzed loop. Input codes 39, 55, and 65 produce an octal dump of K Table, a Macro dump of the updated program unit, and a Transcriber listing of the updated program unit, respectively.

TRACE 302. About to exit from the paralyzer. If there had been more than one possible DOFORALL index set, the Paralyzer would have repeated TRACE 12 through TRACE 16 until each set had been individually processed.

```
PA01A DEM    V07.26.72    DATE  08/05/72    TIME  15:48:52    PAGE    1
            SUBROUTINE PA01A
      C PA01A:   CATEGORY 01 (SIMPLE). CASE A.
      C
            IMPLICIT INTEGER (A-Z)
            DIMENSION A(10),B(10),C(10),D(10)
      C
      C SIMPLE SINGLE DIMENSION
      C
      C     .
      C     .
            DO 100 I=1,10
            A(I)=1
            B(I)=A(I)+B(I)+2
            C(I)=A(I)+3
        100 A(I)=D(I)
            RETURN
            END
```



```
PA01A DEM    V07.26.72    DATE  08/05/72    TIME  15:48:52    PAGE    2

 MIDSTREAM PARALYSIS REPORT --

  DIMENSION OF DO NEST:  1
  INDEX SET:  (I)
  ENDING LABEL OF DO NEST:  100
  STILL GOOD DO-FOR-ALL SETS:
    (I)
```

```
PA01A DEM    V07.26.72    DATE  08/05/72    TIME  15:48:52    PAGE    3
C
C !! 21
C
      SUBROUTINE PA01A
      IMPLICIT INTEGER(A-Z)
      DIMENSION A(10)[(1)],B(10)[(1)],C(10)[(1)],D(10)[(1)]
      DO 100 FOR ALL (I)/[1,2...10]
      A(I)=1
      B(I)=A(I)+B(I)+2
      C(I)=A(I)+3
      A(I)=D(I)
  100 CONTINUE
      RETURN
      END

  STATISTICS OF INTEREST ---

    STRIP MINEABLE?  YES
    OLD FREQUENCY:   48
    NEW FREQUENCY:   4
```

```
PA02A DEM    V07.26.72    DATE  08/11/72    TIME  10.46.46    PAGE    1
            SUBROUTINE PA02A
      C PA02A:   CATEGORY 02 (BOUNDS CREATION). CASE A.
      C
            IMPLICIT INTEGER (A-Z)
            DIMENSION A(50),B(60),C(70),D(50)[(1)]
      C
      C A VARIETY OF DO LIMITS, BUT ALWAYS POSITIVE DO INCREMENT.
      C   7 DISCRETE DO LOOPS WITHIN ONE PROGRAM UNIT.
      C
      C     .
      C     .
      C LOWER LIMIT INTEGER, BUT NOT 1
            DO 1 I=5,50
          1 A(I-4)=B(I)+I
      C
      C UPPER LIMIT VARIABLE
            DO 2 I=7,N
          2 A(I+4)=B(I+Q)
      C
      C LOWER LIMIT VARIABLE
            DO 3 I=M,43
          3 A(I-4)=D(I)
      C
      C LOWER AND UPPER LIMITS BOTH VARIABLE
            DO 4 I=M,N
          4 A(I-4)=D(I+5)
      C
      C EXPRESSION FOR UPPER BOUNDS
            DO 5 I=6,2*N+7,1
          5 A(I-4)=B(I+4)
      C
      C LOWER AND UPPER LIMIT BOTH INTEGER BUT DECREMENT =2
            DO 6 I=1,10,2
          6 A(I)=0
      C
      C DECREMENT =2 AND UPPER LIMIT VARIABLE
            DO 7 I=5,N,2
          7 A(I-4)=D(I)
            RETURN
            END
```

```
PA02A DEM    V07.26.72    DATE  08/07/72    TIME  13:17:03    PAGE    2

 MIDSTREAM PARALYSIS REPORT --

  DIMENSION OF DO NEST:  1
  INDEX SET:  (I)
  ENDING LABEL OF DO NEST:  1
  STILL GOOD DO-FOR-ALL SETS:
    (I)

 STATISTICS OF INTEREST ---

   STRIP MINEABLE?  YES
   OLD FREQUENCY:   46
   NEW FREQUENCY:   1
```

```
PA02A.DEM   V07.26.72   DATE  08/07/72     TIME  13:17:03     PAGE        3

 MIDSTREAM PARALYSIS REPORT --

  DIMENSION OF DO NEST:  1
  INDEX SET:  (I)
  ENDING LABEL OF DO NEST:  2
  STILL GOOD DO-FOR-ALL SETS:
    (I)

 STATISTICS OF INTEREST ---

   STRIP MINEABLE?  YES
   OLD FREQUENCY:  1*(N-6)
   NEW FREQUENCY:  1
```

```
PA02A.DEM   V07.26.72   DATE  08/07/72     TIME  13:17:03     PAGE        5

 MIDSTREAM PARALYSIS REPORT --

  DIMENSION OF DO NEST:  1
  INDEX SET:  (I)
  ENDING LABEL OF DO NEST:  4
  STILL GOOD DO-FOR-ALL SETS:
    (I)

 STATISTICS OF INTEREST ---

   STRIP MINEABLE?  YES
   OLD FREQUENCY:  1*(N-M+1)
   NEW FREQUENCY:  1
```

```
PA02A.DEM   V07.26.72   DATE  08/07/72     TIME  13:17:03     PAGE        4

 MIDSTREAM PARALYSIS REPORT --

  DIMENSION OF DO NEST:  1
  INDEX SET:  (I)
  ENDING LABEL OF DO NEST:  3
  STILL GOOD DO-FOR-ALL SETS:
    (I)

 STATISTICS OF INTEREST ---

   STRIP MINEABLE?  YES
   OLD FREQUENCY:  1*(-M+44)
   NEW FREQUENCY:  1
```

```
PA02A.DEM   V07.26.72   DATE  08/07/72     TIME  13:17:03     PAGE        6

 MIDSTREAM PARALYSIS REPORT --

  DIMENSION OF DO NEST:  1
  INDEX SET:  (I)
  ENDING LABEL OF DO NEST:  5
  STILL GOOD DO-FOR-ALL SETS:
    (I)

 STATISTICS OF INTEREST ---

   STRIP MINEABLE?  YES
   OLD FREQUENCY:  1*(2*N+2)
   NEW FREQUENCY:  1
```

```
PA02A DEM    V07.26.72    DATE 08/07/72    TIME 13:17:03    PAGE    7

MIDSTREAM PARALYSIS REPORT --

 DIMENSION OF DO NEST:  1
 INDEX SET:  (I)
 ENDING LABEL OF DO NEST:  6
 STILL GOOD DO-FOR-ALL SETS:
   (I)

STATISTICS OF INTEREST ---

  STRIP MINEABLE?  YES
  OLD FREQUENCY:  5
  NEW FREQUENCY:  1
```



```
PA02A DEM    V07.26.72    DATE 08/11/72    TIME 10:46:46    PAGE    8

MIDSTREAM PARALYSIS REPORT --

 DIMENSION OF DO NEST:  1
 INDEX SET:  (I)
 ENDING LABEL OF DO NEST:  7
 STILL GOOD DO-FOR-ALL SETS:
   (I)

<F,G> SETS --

---- EMPTY ----

STATISTICS OF INTEREST ---

  STRIP MINEABLE?  YES
  OLD FREQUENCY:  CANNOT NOW ACCURATELY EXPRESS.
  NEW FREQUENCY:  1
C
C !! PA02A
C
      SUBROUTI. : PA02A
      IMPLICIT INTEGER(A-Z)
      DIMENSION A(50)[(1)],B(60)[(1)],C(70),D(50)[(1)]
      DO 1 FOR ALL (I)/[1,2...46]
      A(I)=B(I+4)+I+4
    1 CONTINUE
      DO 2 FOR ALL (I)/[(I)/[1,2...40]:I.LE.N-6]
      A(I+10)=B(Q+I+6)
    2 CONTINUE
      DO 3 FOR ALL (I)/[(I)/[1,2...39]:I.GE.M-4]
      A(I)=D(I+4)
    3 CONTINUE
      DO 4 FOR ALL (I)/[(I)/[1,2...41]:I.GE.M-4.AND.I.LE.N-4]
      A(I)=D(I+9)
    4 CONTINUE
      DO 5 FOR ALL (I)/[(I)/[1,2...49]:I.LE.2*N+3]
      A(I+1)=B(I+9)
    5 CONTINUE
      DO 6 FOR ALL (I)/[1,3...9]
      A(I)=0
    6 CONTINUE
      DO 7 FOR ALL (I)/[(I)/[1,3...45]:I.LE.N-4]
      A(I)=D(I+4)
    7 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PA02B
C PA02B:  CATEGORY 02 (BOUNDS CREATION), CASE B.
C
      IMPLICIT INTEGER (A-Z)
      DIMENSION A(50),B(60),C(70),D(50)
C
C TESTING NEGATIVE DO STATEMENT INCREMENTS.
C   6 DISCRETE DO LOOPS WITHIN ONE PROGRAM UNIT.
C       .
C       .
C       .
C UPPER LIMIT INTEGER, BUT NOT 1
      DO 1 I=50,5,-1
1     A(I-4)=B(I)+I
C
C LOWER LIMIT A VARIABLE
      DO 2 I=N,7,-1
      C(I)=0
2     A(I+4)=B(I+Q)
C
C UPPER LIMIT VARIABLE
      DO 3 I=43,M,-1
      C(I)=0
3     A(I-4)=D(I)
C
C VARIABLE LOWER AND UPPER LIMITS
      DO 4 I=N,M,-1
      C(I)=0
4     A(I-4)=D(I+5)
C
C EXPRESSION LOWER BOUND
      DO 5 I=2*N+7,6,-1
5     A(I-4)=B(I+4)
C
C DECREMENT =-2
      DO 6 I=10,1,-2
6     A(I)=B(I)
      RETURN
      END
```



```
MIDSTREAM PARALYSIS REPORT --

  DIMENSION OF DO NEST:  1
  INDEX SET:  (I)
  ENDING LABEL OF DO NEST:  1
  STILL GOOD DO-FOR-ALL SETS:
    (I)

STATISTICS OF INTEREST ---

  STRIP MINEABLE?  YES
  OLD FREQUENCY:  46
  NEW FREQUENCY:  1
```

```
MIDSTREAM PARALYSIS REPORT --

  DIMENSION OF DO NEST:  1
  INDEX SET:  (I)
  ENDING LABEL OF DO NEST:  2
  STILL GOOD DO-FOR-ALL SETS:
    (I)

STATISTICS OF INTEREST ---

  STRIP MINEABLE?  YES
  OLD FREQUENCY:  2*(N-6)
  NEW FREQUENCY:  2
```



```
MIDSTREAM PARALYSIS REPORT --

  DIMENSION OF DO NEST:  1
  INDEX SET:  (I)
  ENDING LABEL OF DO NEST:  3
  STILL GOOD DO-FOR-ALL SETS:
    (I)

STATISTICS OF INTEREST ---

  STRIP MINEABLE?  YES
  OLD FREQUENCY:  2*(-M+44)
  NEW FREQUENCY:  2
```

```
PA02B.DEM    V07.26.72    DATE  08/07/72    TIME  13:06:53    PAGE      5

 MIDSTREAM PARALYSIS REPORT --

  DIMENSION OF DO NEST:  1
  INDEX SET:  (I)
  ENDING LABEL OF DO NEST:  4
  STILL GOOD DO-FOR-ALL SETS:
    (I)

 STATISTICS OF INTEREST ---

   STRIP MINEABLE?  YES
   OLD FREQUENCY:   2*(N-M+1)
   NEW FREQUENCY:   2
```

```
PA02B.DEM    V07.26.72    DATE  08/07/72    TIME  13:06:53    PAGE      6

 MIDSTREAM PARALYSIS REPORT --

  DIMENSION OF DO NEST:  1
  INDEX SET:  (I)
  ENDING LABEL OF DO NEST:  5
  STILL GOOD DO-FOR-ALL SETS:
    (I)

 STATISTICS OF INTEREST ---

   STRIP MINEABLE?  YES
   OLD FREQUENCY:   1*(2*N+2)
   NEW FREQUENCY:   1
```

```
PA02B.DEM    V07.26.72    DATE  08/07/72    TIME  13:06:53    PAGE      7

 MIDSTREAM PARALYSIS REPORT --

  DIMENSION OF DO NEST:  1
  INDEX SET:  (I)
  ENDING LABEL OF DO NEST:  6
  STILL GOOD DO-FOR-ALL SETS:
    (I)

 STATISTICS OF INTEREST ---

   STRIP MINEABLE?  YES
   OLD FREQUENCY:   5
   NEW FREQUENCY:   1
C
C !! PA02B
C
      SUBROUTINE PA02B
      IMPLICIT INTEGER(A-Z)
      DIMENSION A(50)[(1)],B(60)[(1)],C(70)[(1)],D(50)[(1)]
      DO 1 FOR ALL (I)/[1,2...46]
      A(I)=B(I+4)+I+4
    1 CONTINUE
      DO 2 FOR ALL (I)/[(I)/[1,2...40]:I.LE.N-6]
      C(I+6)=0
      A(I+10)=B(Q+I+6)
    2 CONTINUE
      DO 3 FOR ALL (I)/[(I)/[1,2...39]:I.GE.M-4]
      C(I+4)=0
      A(I)=D(I+4)
    3 CONTINUE
      DO 4 FOR ALL (I)/[(I)/[1,2...41]:I.GE.M-4.AND.I.LE.N-4]
      C(I+4)=0
      A(I)=D(I+9)
    4 CONTINUE
      DO 5 FOR ALL (I)/[(I)/[1,2...49]:I.LE.2*N+3]
      A(I+1)=B(I+9)
    5 CONTINUE
      DO 6 FOR ALL (I)/[1,3...9]
      A(I+1)=B(I+1)
    6 CONTINUE
      RETURN
      END
```

```
PA03A.DEM    V07.26.72    DATE  08/07/72    TIME  13:19:18    PAGE    1
        SUBROUTINE PA03A
C PA03A:  CATEGORY 03 .(TRANSFER REMOVAL), CASE A.
C
        IMPLICIT INTEGER (A-Z)
        DIMENSION A(100),B(100),C(100),D(100),E(100)
        LOGICAL L(100)
C
C ELIMINATION OF FORWARD TRANSFERS OF CONTROL.
C       .
C       .
C       .
        DO 100 I=1,20
        A(I)=B(I)
        IF(A(I).GT.10) GOTO 10
        B(I)=C(I)-D(I)
        IF(L(I)) E(I)=C(I)+D(I)
        GOTO 20
 10     B(I)=C(I)/D(I)
        E(I)=-C(I)+D(I)
 20     A(I)=A(I)+B(I)+E(I)
 100    CONTINUE
        RETURN
        END
        .
```



```
PA03A.DEM    V07.26.72    DATE  08/07/72    TIME  13:19:18    PAGE    2

MIDSTREAM PARALYSIS REPORT --

 DIMENSION OF DO NEST:  1
 INDEX SET:  (I)
 ENDING LABEL OF DO NEST:  100
 STILL GOOD DO-FOR-ALL SETS:
   (I)
```

```
PA03A.DEM    V07.26.72    DATE  08/07/72    TIME  13:19:18    PAGE  .  3
C
C !! Z1
C
      SUBROUTINE PA03A
      IMPLICIT INTEGER(A-Z)
      LOG.CAL L
      DIMENSION A(100)[(1)],B(100)[(1)],C(100)[(1)],D(100)[(1)],E(100)
     1[(1)],L(100)[(1)]
      DO 100 FOR ALL (I)/[1,2...20]
      A(I)=B(I)
      IF(.NOT.A(I).GT.10) B(I)=C(I)-D(I)
      IF(L(I).AND.(.NCT.A(I).GT.10)) E(I)=C(I)+D(I)
      IF(A(I).GT.10) B(I)=C(I)/D(I)
      IF(A(I).GT.10) E(I)=(-C(I))+D(I)
      A(I)=A(I)+B(I)+E(I)
  100 CONTINUE
      RETURN
      END

 STATISTICS OF INTEREST ---

   STRIP MINEABLE?  YES
   OLD FREQUENCY:  160
   NEW FREQUENCY:  6
```

```
PA04A DEM    V07.26.72   DATE  08/05/72    TIME  15:59:37    PAGE        1
  1             SUBROUTINE PA04A(Y)
  2   C PA04A:  CATEGORY 04 (SCALAR REMOVAL), CASE A
  3   C
  4             IMPLICIT INTEGER (A-Z)
  5             DIMENSION A(10),B(10),C(10),D(10),E(10)
  6   C
  7   C SHOWS SCALAR ELIMINATION BY INSERTING EXPRESSION INTO
  8   C   DOWNSTREAM STATEMENTS AND DELETING STATEMENT WITH
  9   C   SCALAR GENERATION (I.E., X GENERATIONS).
 10   C   HOWEVER, Y MUST BE SUBSTITUTED WITH TEMPORARY ARRAY
 11   C   SINCE THE SCALAR IS USED OUTSIDE (AFTER) THE LOOP.
 12   C   A WARNING ABOUT AN EXTERNAL FUNCTION REFERENCE IS MADE.
 13   C         .
 14   C         .
 15   C         .
 16             DO 10 I=1,10
 17             X=C(I)+2
 18             A(I)=X+A1
 19             X=I+2+Q
 20             B(I)=E(X)+B(I)
 21             Y=2+A(I)
 22             D(I)=Y+C(I)
 23             X=2+B(I)-1
 24             C(I)=X-SIN(X)
 25   10        CONTINUE
 26             RETURN
 27             END
 28
!WARNING!  EXTERNAL FUNCTION REFERENCE  [LINE 24]
```

```
PA04A DEM    V07.26.72   DATE  08/05/72    TIME  15:59:37    PAGE        2

MIDSTREAM PARALYSIS REPORT --

 DIMENSION OF DO NEST:  1
 INDEX SET:  (I)
 ENDING LABEL OF DO NEST:  10
 STILL GOOD DO-FOR-ALL SETS:
   (I)
```

```
PA04A DEM    V07.26.72   DATE  08/05/72    TIME  15:59:37    PAGE  .     3
C
C !! Z1
C
      SUBROUTINE PA04A(Y)
      IMPLICIT INTEGER(A-Z)
      DIMENSION A(10)[(1)],B(10)[(1)],C(10)[(1)],D(10)[(1)],E(10)[(1)]
     1,T0000T(10)[(1)]
      DO 10 FOR ALL (I)/[1,2...10]
      A(I)=C(I)+2+A1
      B(I)=E(I+2+Q)+B(I)
      T0000T(I)=3+A(I)
      D(I)=T0000T(I)+C(I)
      C(I)=2+B(I)-1-SIN(2+B(I)-1)
   10 CONTINUE
      Y=T0000T(10)
      RETURN
      END

STATISTICS OF INTEREST ---

  STRIP MINEABLE?  YES
  SCALAR PROMOTION -
    # OF NEW ARRAYS:  1
    TOTAL WORDS:  10
  OLD FREQUENCY:  80
  NEW FREQUENCY:  5
```

```
PA04B.DEM    V07.26.72    DATE  08/05/72    TIME  16:00:19    PAGE     1
               SUBROUTINE PA04B
      C PA04B:   CATEGORY 04 (SCALAR REMOVAL), CASE B.
      C
               IMPLICIT INTEGER (A-Z)
               DIMENSION A(10),B(10),C(10),D(10),E(10),F(10)
               LOGICAL L(10)
      C
      C SCALAR REMOVAL WITH SUMS AND PRODUCTS SPECIAL HANDLING
      C        .
      C        .
      C        .
               SUM=0
               PROD=1
               DO 1 I=1,10
               IF(L(I)) PROD=PROD*(E(I)-Q)
               TMP=A(I)+B(I)
               D(I)=E(I)/TMP
               SUM=I+A(I)+SUM
               C(I)=TMP**2
               TMP=I+Q
      1        B(I)=F(TMP)
               RETURN
               END
```

```
PA04B.DEM    V07.26.72    DATE  08/05/72    TIME  16:00:19    PAGE     2

 MIDSTREAM PARALYSIS REPORT --

  DIMENSION OF DO NEST:  1
  INDEX SET:  (I)
  ENDING LABEL OF DO NEST:  1
  STILL GOOD DO-FOR-ALL SETS:
    (I)
```

```
PA04B.DEM    V07.26.72    DATE  08/05/72    TIME  16:00:19    PAGE     3
C
C !! 21
C
      SUBROUTINE PA04B
      IMPLICIT INTEGER(A-Z)
      LOGICAL L
      DIMENSION A(10)[(1)],B(10)[(1)],C(10)[(1)],D(10)[(1)],E(10)[(1)]
     1,F(10)[(1)],L(10)[(1)],T0000T(10)[(1)],T0001T(10)[(1)],T0002T(10
     1)[(1)]
      SUM=0
      PROD=1
      DO 1 FOR ALL (I)/[1,2..10]
      T0001T(I)=1
      IF(L(I)) T0001T(I)=(E(I)-Q)
      T0000T(I)=A(I)+B(I)
      D(I)=E(I)/T0000T(I)
      T0002T(I)=I+A(I)
      C(I)=T0000T(I)**2
      B(I)=F(I+Q)
    1 CONTINUE
      DO 90000 I=1,10,1
      PROD=PROD*T0001T(I)
      SUM=SUM+T0002T(I)
90000 CONTINUE
      RETURN
      END

 STATISTICS OF INTEREST --

   STRIP MINEABLE?  YES
   SCALAR PROMOTION -
     # OF NEW ARRAYS:  3
     TOTAL WORDS:  30
   OLD FREQUENCY:  70
   NEW FREQUENCY:  CANNOT NOW ACCURATELY EXPRESS
```

```
PA04C.DEM    V07.26.72    DATE  08/05/72     TIME  16:00:51     PAGE      1
           SUBROUTINE PA04C
      C PA04C:   CATEGORY 04 (SCALAR REMOVAL), CASE C
      C
           IMPLICIT INTEGER (A-Z)
           REAL A(15,25,35),B(20,10,30)
      C
      C MOVE SCALAR GENERATIONS TO FOLLOWING DO LIMITS, IF POSSIBLE
      C   7 POSSIBLE DO FOR ALL COMBINATIONS
      C   EXPRESSION FOR UPPER BOUND ON K DO VARIABLE
      C   EXPRESSION FOR LOWER BOUND ON J DO VARIABLE
      C          .
      C          .
      C          .
           DO 1 I=1,10
           M=I+2
           DO 1 J=M,20
           N=J+3
           DO 1 K=1,N
      1    A(I,J,K)=B(J-2,I,K)
           RETURN
           END
```

```
PA04C.DEM    V07.26.72    DATE  08/05/72     TIME  16:00:51     PAGE      2

MIDSTREAM PARALYSIS REPORT --

 DIMENSION OF DO NEST:  3
 INDEX SET:  (I,J,K)
 ENDING LABEL OF DO NEST:  1
 STILL GOOD DO-FOR-ALL SETS:
   (I,J,K)
   (I,J)
   (I,K)
   (J,K)
   (I)
   (J)
   (K)
```

```
PA04C.DEM    V07.26.72    DATE  08/05/72     TIME  16:00:51     PAGE      3
C
C !! 23
C
      SUBROUTINE PA04C
      IMPLICIT INTEGER(A-Z)
      REAL A,B
      DIMENSION A(15,25,35)[(3)]
      DIMENSION B(20,10,30)[(3)]
      DO 1 I=1,10,1
      DO 1 J=I+2,20,1
      DO 1 FOR ALL (K)/[(K)/[1,2...23]:K.LE.J+3]
      A(I,J,K)=B(J-2,I,K)
    1 CONTINUE
      RETURN
      END

STATISTICS OF INTEREST --

  STRIP MINEABLE?  YES
  OLD FREQUENCY:   12420
  NEW FREQUENCY:   180
```

```
PA04C.DEM    V07.26.72    DATE  08/05/72     TIME  16:00:51     PAGE      4
C
C !! 22
C
      SUBROUTINE PA04C
      IMPLICIT INTEGER(A-Z)
      REAL A,B
      DIMENSION A(15,25,35)[(2)]
      DIMENSION B(20,10,30)[(1)]
      DO 1 I=1,10,1
      DO 1 K=1,23,1
      DO 1 FOR ALL (J)/[(J)/[1,2...18]:J.GE.1.AND.K.LE.J+5]
      A(I,J+2,K)=B(J,I,K)
    1 CONTINUE
      RETURN
      END

STATISTICS OF INTEREST --

  STRIP MINEABLE?  YES
  OLD FREQUENCY:   12420
  NEW FREQUENCY:   230
```

```
PA04C.DEM    V07.26.72    DATE  08/05/72    TIME  16:00:51    PAGE    5
C
C !! Z1
C
      SUBROUTINE PA04C
      IMPLICIT INTEGER(A-Z)
      REAL A,B
      DIMENSION A(15,25,35)[(1)]
      DIMENSION B(20,10,30)[(2)]
      DO 1 J=3,20,1
      DO 1 K=1,J+3,1
      DO 1 FOR ALL (I)/[(I)/[1,2...10]:J.GE.I+2]
      A(I,J,K)=B(J-2,I,K)
    1 CONTINUE
      RETURN
      END

 STATISTICS OF INTEREST ---

   STRIP MINEABLE?  YES
   OLD FREQUENCY:  12420
   NEW FREQUENCY:  414
```

```
PA04C.DEM    V07.26.72    DATE  08/05/72    TIME  16:00:51    PAGE    6
C
C !! Z23
C
      SUBROUTINE PA04C
      IMPLICIT INTEGER(A-Z)
      REAL A,B
      DIMENSION A(15,25,35)[(2,3)]
      DIMENSION B(20,10,30)[(1,3)]
      DO 1 I=1,10,1
      DO 1 FOR ALL (J,K)/[(J,K)/[1,2...18] CROSS [1,2...23]:J.GE.I
     1 .AND.K.LE.J+5]
      A(I,J+2,K)=B(J,I,K)
    1 CONTINUE
      RETURN
      END

 STATISTICS OF INTEREST ---

   STRIP MINEABLE?  YES
   OLD FREQUENCY:  12420
   NEW FREQUENCY:  70
```

```
PA04C.DEM    V07.26.72    DATE  08/05/72    TIME  16:00:51    PAGE    7
C
C !! Z13
C
      SUBROUTINE PA04C
      IMPLICIT INTEGER(A-Z)
      REAL A,B
      DIMENSION A(15,25,35)[(1,3)]
      DIMENSION B(20,10,30)[(2,3)]
      DO 1 J=3,20,1
      DO 1 FOR ALL (I,K)/[(I,K)/[1,2...10] CROSS [1,2...23]:J.GE.I+2
     1 .AND.K.LE.J+3]
      A(I,J,K)=B(J-2,I,K)
    1 CONTINUE
      RETURN
      END

 STATISTICS OF INTEREST ---

   STRIP MINEABLE?  YES
   OLD FREQUENCY:  12420
   NEW FREQUENCY:  72
```

```
PA04C.DEM    V07.26.72    DATE  08/05/72    TIME  16:00:51    PAGE    8
C
C !! Z12
C
      SUBROUTINE PA04C
      IMPLICIT INTEGER(A-Z)
      REAL A,B
      DIMENSION A(15,25,35)[(1,2)]
      DIMENSION B(20,10,30)[(2,1)]
      DO 1 K=1,23,1
      DO 1 FOR ALL (I,J)/[(I,J)/[1,2...10] CROSS [1,2...18]:J.GE.I
     1 .AND.K.LE.J+5]
      A(I,J+2,K)=B(J,I,K)
    1 CONTINUE
      RETURN
      END

 STATISTICS OF INTEREST ---

   STRIP MINEABLE?  YES
   OLD FREQUENCY:  12420
   NEW FREQUENCY:  69
```

```
PA04C.DEM    V07.26.72    DATE  08/05/72    TIME  16:00:51    PAGE    9
C
C !! Z123
C
      SUBROUTINE PA04C
      IMPLICIT INTEGER(A-Z)
      REAL A,B
      DIMENSION A(15,25,35)[(1,2,3)]
      DIMENSION B(20,10,30)[(2,1,3)]
      DO 1 FOR ALL (I,J,K)/[(I,J,K)/[1,2..10] CROSS.[1,2...18] CROSS.
     1[1,2...23]:J.GE.I.AND.K.LE.J+5]
      A(I,J+2,K)=B(J,I,K)
    1 CONTINUE
      RETURN
      END

 STATISTICS OF INTEREST ---

   STRIP MINEABLE?  YES
   OLD FREQUENCY:  12420
   NEW FREQUENCY:  65
```

```
PA05A.DEM    V07.26.72    DATE  08/05/72    TIME  16:02:07    PAGE    1
            SUBROUTINE PA05A
      C PA05A:  CATEGORY C5 (TIGHT NESTING), CASE A
      C
            IMPLICIT INTEGER (A-Z)
            DIMENSION A(10),B(10,20),C(20,10),D(10)
      C TIGHT-NESTING REQUIRED FOR (I,J) AND (I) DO-SIMMING
      C   3 POSSIBLE DO-FOR-ALL COMBINATIONS.
      C         .
      C         .
      C         .
            DO 10 I=1,10,1
            A(I)=D(I)-7
            DO 20 J=1,20,2
            B(I,J)=C(J,I)+B(I,J)
            C(J,I)=B(I,J)
       20   CONTINUE
            D(I)=2+A(I)+N
       10   CONTINUE
            RETURN
            END
```

```
PA05A.DEM    V07.26.72    DATE  08/05/72    TIME  16:02:07    PAGE    2

 MIDSTREAM PARALYSIS REPORT ---

  DIMENSION OF DO NEST:  2
  INDEX SET:  (I,J)
  ENDING LABEL OF DO NEST:  10
  STILL GOOD DO-FOR-ALL SETS:
    (I,J)
    (I)
    (J)
```

```
PA05A.DEM    V07.26.72   DATE  08/05/72    TIME  16:02:07    PAGE      3
C
C !! Z2
C
      SUBROUTINE PA05A
      IMPLICIT INTEGER(A-Z)
      DIMENSION A(10)
      DIMENSION B(10,20)[(2)]
      DIMENSION C(20,10)[(1)],D(10)
      DO 10 I=1,10,1
      DO 20 FOR ALL (J)/[1,3...19]
      IF(J.EQ.1) A(I)=D(I)-7
      B(I,J)=C(J,I)+B(I,J)
      C(J,I)=B(I,J)
      IF(J.EQ.19) D(I)=2*A(I)+N
   20 CONTINUE
   10 CONTINUE
      RETURN
      END


 STATISTICS OF INTEREST ---

   STRIP MINEABLE?  NO
   OLD FREQUENCY:   400
   NEW FREQUENCY:   40
   ALLOCATION WARNINGS FOR THESE ARRAYS-
     A  !TIGHT NESTING INTERFERENCE!
     D  !TIGHT NESTING INTERFERENCE!
```

```
PA05A.DEM    V07.26.72   DATE  08/05/72    TIME  16:02:07    PAGE      4
C
C !! Z1
C
      SUBROUTINE PA05A
      IMPLICIT INTEGER(A-Z)
      DIMENSION A(10)[(1)]
      DIMENSION B(10,20)[(1)]
      DIMENSION C(20,10)[(2)],D(10)[(1)]
      DO 20 J=1,20,2
      DO 10 FOR ALL (I)/[1,2...10]
      IF(J.EQ.1) A(I)=D(I)-7
      B(I,J)=C(J,I)+B(I,J)
      C(J,I)=B(I,J)
      IF(J.EQ.19) D(I)=2*A(I)+N
   10 CONTINUE
   20 CONTINUE
      RETURN
      END

 STATISTICS OF INTEREST ---

   STRIP MINEABLE?  YES
   OLD FREQUENCY:   400
   NEW FREQUENCY:   40
```

```
PA05A.DEM    V07.26.72   DATE  08/05/72    TIME  16:02:07    PAGE      5
C
C !! Z12
C
      SUBROUTINE PA05A
      IMPLICIT INTEGER(A-Z)
      DIMENSION A(10)
      DIMENSION B(10,20)[(1,2)]
      DIMENSION C(20,10)[(2,1)],D(10)
      DO 10 FOR ALL (I,J)/[1,2...10].CROSS.[1,3...19]
      IF(J.EQ.1) A(I)=D(I)-7
      B(I,J)=C(J,I)+B(I,J)
      C(J,I)=B(I,J)
      IF(J.EQ.19) D(I)=2*A(I)+N
   10 CONTINUE
      RETURN
      END

 STATISTICS OF INTEREST ---

   STRIP MINEABLE?  NO
   OLD FREQUENCY:   400
   NEW FREQUENCY:   8
   ALLOCATION WARNINGS FOR THESE ARRAYS-
     A  !TIGHT NESTING INTERFERENCE!
     D  !TIGHT NESTING INTERFERENCE!
```

```
PA05B.DEM   V07.26.72   DATE  08/05/72    TIME  16:12:30     PAGE        1
      SUBROUTINE PA05B(D)
C PA05B·  CATEGORY 05 (TIGHT NESTING), CASE B.
C
      IMPLICIT INTEGER (A-Z)
      COMMON A
      DIMENSION A(10),B(10),C(10,75),D(10)
      DIMENSION E(10),F(10),G(10)
C
C A TIGHT NESTING CASE WITH UNUSUAL DO STATEMENT COM-
C   PONENTS FOR (J) DO VARIABLE, I.E. '4,P,3'.
C   SIMMABLE ONLY ON (I) SINCE THERE ARE 'PSEUDO-SCALARS',
C   E.G., ARRAY E.  WARNINGS ABOUT COMMON AND DUMMY
C   ARRAYS ARE MADE.
C
C      .
C      .
C      .
      DO 710 I=1,10,1
      A(I)=D(I)-7
      DO 620 J=4,P,3
      C(I,J)=E(I)
      E(I)=C(I,J)**2
620   CONTINUE
      G(I)=A(I)+F(I-2*Q)
      D(I)=A(I)/5
710   CONTINUE
      RETURN
      END
```

```
PA05B.DEM   V07.26.72   DATE  08/05/72    TIME  16:12:30     PAGE        2

MIDSTREAM PARALYSIS REPORT ---

 DIMENSION OF DO NEST:  2
 INDEX SET:  (I,J)
 ENDING LABEL OF DO NEST:  710
 STILL GOOD DO-FOR-ALL SETS:
   (I,J)
   (I)
   (J)
```

```
PA05B.DEM   V07.26.72   DATE  08/05/72    TIME  16:12:30     PAGE        3

SET Z2
REJECTED BECAUSE:

 CDCODE = 000000000001
  ALLOCATION WARNINGS FOR THESE ARRAYS-
    A  !COMMON STORAGE!
    D  !DUMMY FORMAL PARAMETER!
```

```
PA05B.DEM   V07.26.72   DATE  08/05/72    TIME  16:12:30     PAGE        4
C
C !! Z1
C
      SUBROUTINE PA05B(D)
      IMPLICIT INTEGER(A-Z)
      DIMENSION D(10)[(1)],A(10)[(1)],B(10)
      DIMENSION C(10,75)[(1)]
      DIMENSION E(10)[(1)],F(10)[(1)],G(10)[(1)]
      COMMON A
      DO 620 J=4,P,3
      DO 710 FOR ALL (I)/[1,2...10]
      IF(J.EQ.4) A(I)=D(I)-7
      C(I,J)=E(I)
      E(I)=C(I,J)**2
      IF(J.EQ.P-MOD(P-4,3)) G(I)=A(I)+F(I-2*Q)
      IF(J.EQ.P-MOD(P-4,3)) D(I)=A(I)/5
  710 CONTINUE
  620 CONTINUE
      RETURN
      END

STATISTICS OF INTEREST ---

  STRIP MINEABLE?  YES
  OLD FREQUENCY:  CANNOT NOW ACCURATELY EXPRESS.
  NEW FREQUENCY:  CANNOT NOW ACCURATELY EXPRESS.
  ALLOCATION WARNINGS FOR THESE ARRAYS-
    A  !COMMON STORAGE!
    D  !DUMMY FORMAL PARAMETER!
```

```
PA05B.DEM   V07.26.72   DATE  08/05/72    TIME  16:12:30     PAGE        5

SET Z12
REJECTED BECAUSE:

 CDCODE = 000000000001
  ALLOCATION WARNINGS FOR THESE ARRAYS-
    A  !COMMON STORAGE!
    D  !DUMMY FORMAL PARAMETER!
```

```
PAO6A DEM    V07.26.72   DATE  08/05/72    TIME  16:03:25    PAGE     1
              SUBROUTINE PAO6A
      C PAO6A:  CATEGORY 06 (DATA DEPENDENCIES), CASE A.
      C
              IMPLICIT INTEGER (A-Z)
              DIMENSION A(100)
      C
      C ONE DIMENSIONAL, INTRODUCTION OF TEMPORARY ARRAYS
      C   BECAUSE OF OVERWRITE.
      C       .
      C       .
      C       .
              DO 1 I=3,98
              A(I+2)=FOO
              A(I+1)=GOO
              A(I)=HOO
              A(I-1)=A(I+1)+KOO
              A(I-2)=MOO
      1       CONTINUE
              RETURN
              END
```

```
VALUES FOR <F,G> SETS --

  1)      1
  2)      2
  3)      3
  4)      4
  5)      1
  6)      2
  7)      3
  8)      1
  9)      2
 10)      1
 11)     -1
 12)      0
 13)      1
 14)      2
 15)      3
```

```
PAO6A DEM    V07.26.72   DATE  08/05/72    TIME  16:03:25    PAGE     2

 MIDSTREAM PARALYSIS REPORT --

  DIMENSION OF DO NEST:  1
  INDEX SET:  (I)
  ENDING LABEL OF DO NEST:  1
  STILL GOOD DO-FOR-ALL SETS:
    (I)

<F,G> SETS --

   1)  <G,G> = <A(I+2)/13,A(I+1)/14>
   2)  <G,G> = <A(I+2)/13,A(I)/15>
   3)  <G,G> = <A(I+2)/13,A(I-1)/16>
   4)  <G,G> = <A(I+2)/13,A(I-2)/17>
   5)  <G,G> = <A(I+1)/14,A(I)/15>
   6)  <G,G> = <A(I+1)/14,A(I-1)/16>
   7)  <G,G> = <A(I+1)/14,A(I-2)/17>
   8)  <G,G> = <A(I)/15,A(I-1)/16>
   9)  <G,G> = <A(I)/15,A(I-2)/17>
  10)  <G,G> = <A(I-1)/16,A(I-2)/17>
  11)  <F,G> = <A(I+1)/16,A(I+2)/13>
  12)  <F,G> = <A(I+1)/16,A(I+1)/14>
  13)  <F,G> = <A(I+1)/16,A(I)/15>
  14)  <F,G> = <A(I+1)/16,A(I-1)/16>
  15)  <F,G> = <A(I+1)/16,A(I-2)/17>
```

```
PAO6A DEM    V07.26.72   DATE  08/05/72    TIME  16:03:25    PAGE     3
C
C !! Z1
C
      SUBROUTINE PAO6A
      IMPLICIT INTEGER(A-Z)
      DIMENSION A(100)[(1)],T0000T(96)[(1)]
      DO 1 FOR ALL (I)/[1,2..96]
      A(I+4)=FOO
      A(I+3)=GOO
      T0000T(I)=A(I+3)
      A(I+2)=HOO
      A(I+1)=T0000T(I)+KOO
      A(I)=MOO
    1 CONTINUE
      RETURN
      END

 STATISTICS OF INTEREST --

   STRIP MINEFELE?  NO
   OVERWRITE ELIMINATION -
     # OF NEW ARRAYS:  1
     TOTAL WORDS:  96
   OLD FREQUENCY:  480
   NEW FREQUENCY:  12
```

```
PA06B.DEM    V07.26.72    DATE  08/05/72     TIME  16:03:51     PAGE      1
     1             SUBROUTINE PA06B
     2     C PA06B:  CATEGORY 06 (DATA DEPENDENCIES), CASE B.
     3     C
     4             IMPLICIT INTEGER(A-Z)
     5             DIMENSION A(9),B(9),C(9),D(9),E(9),F(9),G(9),H(9)
     6     C
     7     C ONE DIMENSIONAL RE-ORDERING OF STATEMENTS REQUIRED.
     8     C
     9     C       .
    10     C       .
    11             DO 1 I=1,6
    12             C(I+1)=B(I)
    13             G(I+1)=E(I)+F(I)
    14             E(I+1)=D(I)
    15             H(I)=G(I)+C(I)
    16             D(I+2)=I+1
    17             B(I+1)=A(I)
    18             F(I+1)=D(I+1)
    19             A(I+1)=I+2
    20             D(I+3)=I**3
    21     1       CONTINUE
    22             RETURN
    23             END
```

```
PA06B.DEM    V07.26.72    DATE  08/05/72     TIME  16:03:51     PAGE      2

MIDSTREAM PARALYSIS REPORT --

 DIMENSION OF DO NEST:  1
 INDEX SET:  (I)
 ENDING LABEL OF DO NEST:  1
 STILL GOOD DO-FOR-ALL SETS:
   (I)

<F,G> SETS --

  1)  <F,G> = <C(I)/15,C(I+1)/12>
  2)  <F,G> = <B(I)/12,B(I+1)/17>
  3)  <F,G> = <G(I)/15,G(I+1)/13>
  4)  <F,G> = <E(I)/13,E(I+1)/14>
  5)  <F,G> = <F(I)/13,F(I+1)/18>
  6)  <G,G> = <D(I+2)/16,D(I+3)/20>
  7)  <F,G> = <D(I)/14,D(I+2)/16>
  8)  <F,G> = <D(I+1)/18,D(I+2)/16>
  9)  <F,G> = <D(I)/14,D(I+3)/20>
 10)  <F,G> = <D(I+1)/18,D(I+3)/20>
 11)  <F,G> = <A(I)/17,A(I+1)/19>
```

```
VALUES FOR <F,G> SETS --

  1)      -1
  2)      -1
  3)      -1
  4)      -1
  5)      -1
  6)      -1
  7)      -2
  8)      -1
  9)      -3
 10)      -2
 11)      -1
```

```
PA06B.DEM    V07.26.72    DATE  08/05/72     TIME  16:03:51     PAGE      3
C
C !! Z1
C
      SUBROUTINE PA06B
      IMPLICIT INTEGER(A-Z)
      DIMENSION A(9)[(1)],B(9)[(1)],C(9)[(1)],D(9)[(1)],E(9)[(1)],F(9)
     1[(1)],G(9)[(1)],H(9)[(1)]
      DO 1 FOR ALL (I)/[1,2...6]
      A(I+1)=I+2
      B(I+1)=A(I)
      C(I+1)=B(I)
      D(I+3)=I**3
      D(I+2)=I+1
      E(I+1)=D(I)
      F(I+1)=D(I+1)
      G(I+1)=E(I)+F(I)
      H(I)=G(I)+C(I)
    1 CONTINUE
      RETURN
      END

STATISTICS OF INTEREST ---

  STRIP MINEABLE?  NO
  OLD FREQUENCY:  54
  NEW FREQUENCY:  9
```

```
PA06C.DEM    V07.26.72   DATE  08/05/72    TIME  16:04:17    PAGE      1
    1         SUBROUTINE PA06C
    2  C PA06C:  CATEGORY 06 (DATA DEPENDENCIES), CASE C.
    3  C
    4         IMPLICIT INTEGER(A-Z)
    5         DIMENSION A(9),B(9)
    6  C
    7  C SINGLE DIMENSION WHERE TEMPORARY ARRAYS ARE INTRODUCED
    8  C   BECAUSE OF OVERWRITE, AND A RE-ORDERING OF STATEMENTS
    9  C   WITHIN LOOP BODY IS REQUIRED.
   10  C         .
   11  C         .
   12  C         .
   13            DO 1 I=1,7
   14            A(I+2)=FOO
   15            A(I)=B(I)
   16            B(I)=FEE
   17            A(I+1)=A(I+2)+B(I)
   18            A(I+2)=B(I)
   19  1         CONTINUE
   20            RETURN
   21            END
```

```
PA06C.DEM    V07.26.72   DATE  08/05/72    TIME  16:04:17    PAGE      2

MIDSTREAM PARALYSIS REPORT --

 DIMENSION OF DO NEST:  1
 INDEX SET:  (I)
 ENDING LABEL OF DO NEST:  1
 STILL GOOD DO-FOR-ALL SETS:
   (I)

<F,G> SETS --

  1)  <G,G> = <A(I+2)/14,A(I)/15>
  2)  <G,G> = <A(I+2)/14,A(I+1)/17>
  3)  <G,G> = <A(I+2)/14,A(I+2)/18>
  4)  <G,G> = <A(I)/15,A(I+1)/17>
  5)  <G,G> = <A(I)/15,A(I+2)/18>
  6)  <G,G> = <A(I+1)/17,A(I+2)/18>
  7)  <F,G> = <A(I+2)/17,A(I+2)/14>
  8)  <F,G> = <A(I+2)/17,A(I)/15>
  9)  <F,G> = <A(I+2)/17,A(I+1)/17>
 10)  <F,G> = <A(I+2)/17,A(I+2)/18>
 11)  <F,G> = <B(I)/15,B(I)/16>
 12)  <F,G> = <B(I)/17,B(I)/16>
 13)  <F,G> = <B(I)/18,B(I)/16>
```

```
VALUES FOR <F,G> SETS --

  1)       2
  2)       1
  3)       0
  4)      -1
  5)      -2
  6)      -1
  7)       0
  8)       2
  9)       1
 10)       0
 11)       0
 12)       0
 13)       0
```

```
PA06C.DEM    V07.26.72   DATE  08/05/72    TIME  16:04:17    PAGE      3
C
C !! Z1
C
      SUBROUTINE PA06C
      IMPLICIT INTEGER(A-Z)
      DIMENSION A(9)[(1)],B(9)[(1)],T0000T(7)[(1)],T0001T(7)[(1)]
      DO 1 FOR ALL (I)/[1,2..7]
      T0000T(I)=B(I)
      A(I+2)=FOO
      T0001T(I)=A(I+2)
      B(I)=FEE
      A(I+2)=B(I)
      A(I+1)=T0001T(I)+B(I)
      A(I)=T0000T(I)
    1 CONTINUE
      RETURN
      END

STATISTICS OF INTEREST ---

  STRIP MINEABLE?  NO
  OVERWRITE ELIMINATION -
    # OF NEW ARRAYS:  2
    TOTAL WORDS:  14
  OLD FREQUENCY:  35
  NEW FREQUENCY:  7
```

```
PA06E.DEM    V07.26.72    DATE  08/05/72    TIME  16:05:55    PAGE    1
      SUBROUTINE PA06E
C PA06E:  CATEGORY 06 (DATA DEPENDENCIES), CASE E.
C
      IMPLICIT INTEGER(A-Z)
      DIMENSION A(9,9,9,9),B(9,9,9,9),C(9,9,9,9)
      DIMENSION D(9,9,9,9),E(9,9,9,9)
C
C FANCY 4-DIMENSIONAL CASE WHERE INTRODUCTION OF TEMP-
C    ORARIES TO ELIMINATE OVERWRITE IS REQUIRED.  NOT ALL
C    15 DO-FOR-ALL POSSIBLE SETS ARE SIMMABLE, HOWEVER.
C     .
C     .
C     .
      DO 1 I=1,6
      DO 2 J=3,9
      DO 3 K=1,6
      DO 4 L=1,10
      A(I,J-2,K,L)=E(I,J,K,L)
      A(I+3,J,K+2,L)=B(I,J,K,L)
      A(I+2,J,K+3,L)=C(I,J,K,L)
      A(I,J,K,L)=D(I,J,K,L)
      A(I,J-1,K,L)=A(I+1,J,K,L)
4     CONTINUE
3     CONTINUE
2     CONTINUE
1     CONTINUE
      RETURN
      END
```

```
<F,G> SETS --

  1)  <G,G> = <A(I,J-2,K,L)/18,A(I+3,J,K+2,L)/19>
  2)  <G,G> = <A(I,J-2,K,L)/18,A(I+2,J,K+3,L)/20>
  3)  <G,G> = <A(I,J-2,K,L)/18,A(I,J,K,L)/21>
  4)  <G,G> = <A(I,J-2,K,L)/18,A(I,J-1,K,L)/22>
  5)  <G,G> = <A(I+3,J,K+2,L)/19,A(I+2,J,K+3,L)/20>
  6)  <G,G> = <A(I+3,J,K+2,L)/19,A(I,J,K,L)/21>
  7)  <G,G> = <A(I+3,J,K+2,L)/19,A(I,J-1,K,L)/22>
  8)  <G,G> = <A(I+2,J,K+3,L)/20,A(I,J,K,L)/21>
  9)  <G,G> = <A(I+2,J,K+3,L)/20,A(I,J-1,K,L)/22>
 10)  <G,G> = <A(I,J,K,L)/21,A(I,J-1,K,L)/22>
 11)  <F,G> = <A(I+1,J,K,L)/22,A(I,J-2,K,L)/18>
 12)  <F,G> = <A(I+1,J,K,L)/22,A(I+3,J,K+2,L)/19>
 13)  <F,G> = <A(I+1,J,K,L)/22,A(I+2,J,K+3,L)/20>
 14)  <F,G> = <A(I+1,J,K,L)/22,A(I,J,K,L)/21>
 15)  <F,G> = <A(I+1,J,K,L)/22,A(I,J-1,K,L)/22>

VALUES FOR <F,G> SETS --

  1)     -3     -2     -2      0
  2)     -2     -2     -3      0
  3)      0     -2      0      0
  4)      0     -1      0      0
  5)      1      0     -1      0
```



```
PA06E.DEM    V07.26.72    DATE  08/05/72    TIME  16:05:55    PAGE    2

MIDSTREAM PARALYSIS REPORT --

 DIMENSION OF DO NEST:  4
 INDEX SET:  (I,J,K,L)
 ENDING LABEL OF DO NEST:  1
 STILL GOOD DO-FOR-ALL SETS:
   (I,J,K,L)
   (I,J,K)
   (I,K,L)
   (J,K,L)
   (I,K)
   (J,K)
   (J,L)
   (K,L)
   (J)
   (K)
   (L)
 "4E1" BAD DO-FOR-ALL SETS:
   (I,J,L)
   (I,J)
   (I,L)
   (I)
```

```
PA06E.DEM    V07.26.72    DATE  08/05/72    TIME  16:05:55    PAGE    3
  6)      3      0      2      0
  7)      3      1      2      0
  8)      2      0      3      0
  9)      2      1      3      0
 10)      0      1      0      0
 11)      1      2      0      0
 12)     -2      0     -2      0
 13)     -1      0     -3      0
 14)      1      0      0      0
 15)      1      1      0      0
```

```
C
C !! Z4
C
      SUBROUTINE PAGGE
      IMPLICIT INTEGER(A-Z)
      DIMENSION A(9,9,9,9)[(4)]
      DIMENSION B(9,9,9,9)[(4)]
      DIMENSION C(9,9,9,9)[(4)]
      DIMENSION D(9,9,9,9)[(4)]
      DIMENSION E(9,9,9,9)[(4)]
      DO 1 I=1,6,1
      DO 2 J=3,9,1
      DO 3 K=1,6,1
      DO 4 FOR ALL (L)/[1,2..10]
      A(I,J-2,K,L)=E(I,J,K,L)
      A(I+3,J,K+2,L)=B(I,J,K,L)
      A(I+2,J,K+3,L)=C(I,J,K,L)
      A(I,J,K,L)=D(I,J,K,L)
      A(I,J-1,K,L)=A(I+1,J,K,L)
    4 CONTINUE
    3 CONTINUE
    2 CONTINUE
    1 CONTINUE
      RETURN
      END
```

STATISTICS OF INTEREST ---

STRIP MINEABLE? YES
OLD FREQUENCY: 12600
NEW FREQUENCY: 1260

PAGGE.DEM V07.26.72 DATE 08/05/72 TIME 16:05:55 PAGE 5

```
C
C !! Z3
C
      SUBROUTINE PAGGE
      IMPLICIT INTEGER(A-Z)
      DIMENSION A(9,9,9,9)[(3)]
      DIMENSION B(9,9,9,9)[(3)]
      DIMENSION C(9,9,9,9)[(3)]
      DIMENSION D(9,9,9,9)[(3)]
      DIMENSION E(9,9,9,9)[(3)]
      DO 1 I=1,6,1
      DO 2 J=3,9,1
      DO 4 L=1,10,1
      DO 3 FOR ALL (K)/[1,2..6]
      A(I,J-2,K,L)=E(I,J,K,L)
      A(I+3,J,K+2,L)=B(I,J,K,L)
      A(I+2,J,K+3,L)=C(I,J,K,L)
      A(I,J,K,L)=D(I,J,K,L)
      A(I,J-1,K,L)=A(I+1,J,K,L)
    3 CONTINUE
    4 CONTINUE
    2 CONTINUE
    1 CONTINUE
      RETURN
      END
```

STATISTICS OF INTEREST ---

STRIP MINEABLE? YES
OLD FREQUENCY: 12600
NEW FREQUENCY: 2100

PAGGE.DEM V07.26.72 DATE 08/05/72 TIME 16:05:55 PAGE 6

```
C
C !! Z2
C
      SUBROUTINE PAGGE
      IMPLICIT INTEGER(A-Z)
      DIMENSION A(9,9,9,9)[(2)]
      DIMENSION B(9,9,9,9)[(2)]
      DIMENSION C(9,9,9,9)[(2)]
      DIMENSION D(9,9,9,9)[(2)]
      DIMENSION E(9,9,9,9)[(2)]
      DO 1 I=1,6,1
      DO 3 K=1,6,1
      DO 4 L=1,10,1
      DO 2 FOR ALL (J)/[1,2..7]
      A(I+3,J+2,K+2,L)=B(I,J+2,K,L)
      A(I+2,J+2,K+3,L)=C(I,J+2,K,L)
      A(I,J+2,K,L)=D(I,J+2,K,L)
      A(I,J+1,K,L)=A(I+1,J+2,K,L)
      A(I,J,K,L)=E(I,J+2,K,L)
    2 CONTINUE
    4 CONTINUE
    3 CONTINUE
    1 CONTINUE
      RETURN
      END
```

STATISTICS OF INTEREST ---

STRIP MINEABLE? NO
OLD FREQUENCY: 12600
NEW FREQUENCY: 1800

PAGGE.DEM V07.26.72 DATE 08/05/72 TIME 16:05:55 PAGE 7

SET Z1
REJECTED BECAUSE:

CDCODE = 000000000002

```
PAGGE.DEM    V07.26.72    DATE  08/05/72    TIME  16:05:55    PAGE    8
C
C !! Z34
C
      SUBROUTINE PAGGE
      IMPLICIT INTEGER(A-Z)
      DIMENSION A(9,9,9,9)[(3,4)]
      DIMENSION B(9,9,9,9)[(3,4)]
      DIMENSION C(9,9,9,9)[(3,4)]
      DIMENSION D(9,9,9,9)[(3,4)]
      DIMENSION E(9,9,9,9)[(3,4)]
      DO 1 I=1,6,1
      DO 2 J=3,9,1
      DO 3 FOR ALL (K,L)/[1,2...6] CROSS.[1,2..10]
      A(I,J-2,K,L)=E(I,J,K,L)
      A(I+3,J,K+2,L)=B(I,J,K,L)
      A(I+2,J,K+3,L)=C(I,J,K,L)
      A(I,J,K,L)=D(I,J,K,L)
      A(I,J-1,K,L)=A(I+1,J,K,L)
    3 CONTINUE
    2 CONTINUE
    1 CONTINUE
      RETURN
      END

 STATISTICS OF INTEREST ---

   STRIP MINEABLE?  YES
   OLD FREQUENCY:  12600
   NEW FREQUENCY:  210


PAGGE.DEM    V07.26.72    DATE  08/05/72    TIME  16:05:55    PAGE    9
C
C !! Z24
C
      SUBROUTINE PAGGE
      IMPLICIT INTEGER(A-Z)
      DIMENSION A(9,9,9,9)[(2,4)]
      DIMENSION B(9,9,9,9)[(2,4)]
      DIMENSION C(9,9,9,9)[(2,4)]
      DIMENSION D(9,9,9,9)[(2,4)]
      DIMENSION E(9,9,9,9)[(2,4)]
      DO 1 I=1,6,1
      DO 3 K=1,6,1
      DO 2 FOR ALL (J,L)/[1,2...7] CROSS.[1,2..10]
      A(I+3,J+2,K+2,L)=B(I,J+2,K,L)
      A(I+2,J+2,K+3,L)=C(I,J+2,K,L)
      A(I,J+2,K,L)=D(I,J+2,K,L)
      A(I,J+1,K,L)=A(I+1,J+2,K,L)
      A(I,J,K,L)=E(I,J+2,K,L)
    2 CONTINUE
    3 CONTINUE
    1 CONTINUE
      RETURN
      END

 STATISTICS OF INTEREST ---

   STRIP MINEABLE?  NO
   OLD FREQUENCY:  12600
   NEW FREQUENCY:  360


PAGGE.DEM    V07.26.72    DATE  08/05/72    TIME  16:05:55    PAGE    10

 SET Z14
 REJECTED BECAUSE:

  CDCODE = 000000000002


PAGGE.DEM    V07.26.72    DATE  08/05/72    TIME  16:05:55    PAGE    11
C
C !! Z23
C
      SUBROUTINE PAGGE
      IMPLICIT INTEGER(A-Z)
      DIMENSION A(9,9,9,9)[(2,3)]
      DIMENSION B(9,9,9,9)[(2,3)]
      DIMENSION C(9,9,9,9)[(2,3)]
      DIMENSION D(9,9,9,9)[(2,3)]
      DIMENSION E(9,9,9,9)[(2,3)]
      DO 1 I=1,6,1
      DO 4 L=1,10,1
      DO 2 FOR ALL (J,K)/[1,2...7] CROSS.[1,2..6]
      A(I+3,J+2,K+2,L)=B(I,J+2,K,L)
      A(I+2,J+2,K+3,L)=C(I,J+2,K,L)
      A(I,J+2,K,L)=D(I,J+2,K,L)
      A(I,J+1,K,L)=A(I+1,J+2,K,L)
      A(I,J,K,L)=E(I,J+2,K,L)
    2 CONTINUE
    4 CONTINUE
    1 CONTINUE
      RETURN
      END

 STATISTICS OF INTEREST ---

   STRIP MINEABLE?  NO
   OLD FREQUENCY:  12600
   NEW FREQUENCY:  300
```

```
PA06E.DEM    V07.26.72    DATE  08/05/72    TIME  16:05:55    PAGE    12
C
C !! Z13
C
      SUBROUTINE PA06E
      IMPLICIT INTEGER(A-Z)
      DIMENSION A(9,9,9,9)[(1,3)]
      DIMENSION B(9,9,9,9)[(1,3)]
      DIMENSION C(9,9,9,9)[(1,3)]
      DIMENSION D(9,9,9,9)[(1,3)]
      DIMENSION E(9,9,9,9)[(1,3)]
      DIMENSION T0000T(6,6)[(1,2)]
      DO 2 J=3,9,1
      DO 4 L=1,10,1
      DO 1 FOR ALL (I,K)/[1,2...6] CROSS.[1,2...6]
      A(I,J-2,K,L)=E(I,J,K,L)@
      A(I+3,J,K+2,L)=B(I,J,K,L)
      A(I+2,J,K+3,L)=C(I,J,K,L)
      T0000T(I,K)=A(I+1,J,K,L)
      A(I,J,K,L)=D(I,J,K,L)
      A(I,J-1,K,L)=T0000T(I,K)
    1 CONTINUE
    4 CONTINUE
    2 CONTINUE
      RETURN
      END

 STATISTICS OF INTEREST ---

   STRIP MINEABLE?  NO
   OVERWRITE ELIMINATION -
     # OF NEW ARRAYS:  1
     TOTAL WORDS:  36
   OLD FREQUENCY:  12600
   NEW FREQUENCY:  420


PA06E.DEM    V07.26.72    DATE  08/05/72    TIME  16:05:55    PAGE    13

  SET Z13
  REJECTED BECAUSE:

   CDCODE = 000000000002


PA06E.DEM    V07.26.72    DATE  08/05/72    TIME  16:05:55    PAGE    14
C
C !! Z234
C
      SUBROUTINE PA06E
      IMPLICIT INTEGER(A-Z)
      DIMENSION A(9,9,9,9)[(2,1,4)]
      DIMENSION B(9,9,9,9)[(2,1,4)]
      DIMENSION C(9,9,9,9)[(2,1,4)]
      DIMENSION D(9,9,9,9)[(2,1,4)]
      DIMENSION E(9,9,9,9)[(2,1,4)]
      DO 1 I=1,6,1
      DO 2 FOR ALL (J,K,L)/[1,2...7] CROSS.[1,2...6] CROSS [1,2..10]
      A(I+1,J+2,K+2,L)=B(I,J+2,K,L)
      A(I+2,J+2,K+1,L)=C(I,J+2,K,L)
      A(I,J+2,K,L)=D(I,J+2,K,L)
      A(I,J+1,K,L)=A(I+1,J+2,K,L)
      A(I,J,K,L)=E(I,J+2,K,L)
    2 CONTINUE
    1 CONTINUE
      RETURN
      END

 STATISTICS OF INTEREST ---

   STRIP MINEABLE?  NO
   OLD FREQUENCY:  12600
   NEW FREQUENCY:  210


PA06E.DEM    V07.26.72    DATE  08/05/72    TIME  16:05:55    PAGE    15
C
C !! Z134
C
      SUBROUTINE PA06E
      IMPLICIT INTEGER(A-Z)
      DIMENSION A(9,9,9,9)[(1,3,4)]
      DIMENSION B(9,9,9,9)[(1,3,4)]
      DIMENSION C(9,9,9,9)[(1,3,4)]
      DIMENSION D(9,9,9,9)[(1,3,4)]
      DIMENSION E(9,9,9,9)[(1,3,4)]
      DIMENSION T0000T(6,6,10)[(1,2,3)]
      DO 2 J=3,9,1
      DO 1 FOR ALL (I,K,L)/[1,2...6] CROSS.[1,2...6] CROSS.[1,2..10]
      A(I,J-2,K,L)=E(I,J,K,L)
      A(I+3,J,K+2,L)=B(I,J,K,L)
      A(I+2,J,K+3,L)=C(I,J,K,L)
      T0000T(I,K,L)=A(I+1,J,K,L)
      A(I,J,K,L)=D(I,J,K,L)
      A(I,J-1,K,L)=T0000T(I,K,L)
    1 CONTINUE
    2 CONTINUE
      RETURN
      END

 STATISTICS OF INTEREST ---

   STRIP MINEABLE?  NO
   OVERWRITE ELIMINATION -
     # OF NEW ARRAYS:  1
     TOTAL WORDS:  360
   OLD FREQUENCY:  12600
   NEW FREQUENCY:  252
```

```
PAGE.DEM    V07.26.72    DATE  08/05/72    TIME  16:05:55    PAGE    16

 SET Z124
 REJECTED BECAUSE:

  CDCODE = 000000000002
```

```
PAGE.DEM    V07.26.72    DATE  08/11/72    TIME  10:40:32    PAGE    17
C
C !! Z123
C
      SUBROUTINE PAGSE
      IMPLICIT INTEGER(A-Z)
      DIMENSION A(9,9,9,9)[(1,2,3)]
      DIMENSION B(9,9,9,9)[(1,2,3)]
      DIMENSION C(9,9,9,9)[(1,2,3)]
      DIMENSION D(9,9,9,9)[(1,2,3)]
      DIMENSION E(9,9,9,9)[(1,2,3)]
      DIMENSION T0000T(6,7,6)[(1,2,3)]
      DO 4 L=1,10,1
      DO 1 FOR ALL (I,J,K)/[1,2..6] CROSS [1,2..7] CROSS [1,2..6]
      A(I+3,J+2,K+2,L)=B(I,J+2,K,L)
      A(I+2,J+2,K+3,L)=C(I,J+2,K,L)
      T0000T(I,J,K)=A(I+1,J+2,K,L)
      A(I,J+2,K,L)=D(I,J+2,K,L)
      A(I,J+1,K,L)=T0000T(I,J,K)
      A(I,J,K,L)=E(I,J+2,K,L)
    1 CONTINUE
    4 CONTINUE
      RETURN
      END

 STATISTICS OF INTEREST ---

   STRIP MINEABLE?  NO
   OVERWRITE ELIMINATION -
     # OF NEW ARRAYS:  1
     TOTAL WORDS:  252
   OLD FREQUENCY:  12600
   NEW FREQUENCY:  240
```

```
PAGE.DEM    V07.26.72    DATE  08/11/72    TIME  10:40:32    PAGE    18
C
C !! Z1234
C
      SUBROUTINE PAGSE
      IMPLICIT INTEGER(A-Z)
      DIMENSION A(9,9,9,9)[(1,2,3,4)]
      DIMENSION B(9,9,9,9)[(1,2,3,4)]
      DIMENSION C(9,9,9,9)[(1,2,3,4)]
      DIMENSION D(9,9,9,9)[(1,2,3,4)]
      DIMENSION E(9,9,9,9)[(1,2,3,4)]
      DIMENSION T0000T(6,7,6,10)[(1,2,3,4)]
      DO 1 FOR ALL (I,J,K,L)/[1,2..6] CROSS [1,2..7] CROSS [1,2..6]
     1 CROSS [1,2..10]
      A(I+3,J+2,K+2,L)=B(I,J+2,K,L)
      A(I+2,J+2,K+3,L)=C(I,J+2,K,L)
      T0000T(I,J,K,L)=A(I+1,J+2,K,L)
      A(I,J+2,K,L)=D(I,J+2,K,L)
      A(I,J+1,K,L)=T0000T(I,J,K,L)
      A(I,J,K,L)=E(I,J+2,K,L)
    1 CONTINUE
      RETURN
      END

 STATISTICS OF INTEREST ---

   STRIP MINEABLE?  NO
   OVERWRITE ELIMINATION -
     # OF NEW ARRAYS:  1
     TOTAL WORDS:  2520
   OLD FREQUENCY:  12600
   NEW FREQUENCY:  240
```

```
PA06F.DEM    V07.26.72   DATE  08/05/72    TIME  16:07:51    PAGE    1
            SUBROUTINE PA06F
      C PA06F:  CATEGORY 06 (DATA DEPENDENCIES), CASE F.
      C
            IMPLICIT INTEGER(A-Z)
            DIMENSION A(11,7,5),B(11,7,5),C(11,7,5)
      C
      C THREE DIMENSIONAL, BUT NOT ALL DO-FOR-ALL SETS ARE
      C   PARALYZABLE SINCE 2 DIFFERENT MULTI-OCCURRENCE DATA
      C   CYCLES (I AND K).
      C          .
      C          .
      C          .
            DO 1 I=1,11
            DO 2 J=1,7
            DO 3 K=1,5
            A(I,J,K)=B(I,J,K)
            B(I,J,K)=A(I,J,K+N)
            C(I,J,K)=FOO
            C(I+M,J,K)=FEE
      3     CONTINUE
      2     CONTINUE
      1     CONTINUE
            RETURN
            END
```

```
PA06F.DEM    V07.26.72   DATE  08/05/72    TIME  16:07:51 .  PAGE    2

 MIDSTREAM PARALYSIS REPORT --

  D.MENSION OF DO NEST:  3
  INDEX SET:  (I,J,K)
  ENDING LABEL OF DO NEST:  1
  STILL GOOD DO-FOR-ALL SETS:
    (I,J,K)
    (I,J)
    (I,K)
    (J,K)
    (I)
    (J)
    (K)

<F,G> SETS --

  1)  <F,G> = <A(I,J,K+N)/17,A(I,J,K)/16>
  2)  <F,G> = <B(I,J,K)/16,B(I,J,K)/17>
  3)  <G,G> = <C(I,J,K)/18,C(I+M,J,K)/19>

VALUES FOR <F,G> SETS --

  1)       0       0    +0-
  2)       0       0      0
  3)     +0-       0      0
```

```
PA06F.DEM    V07.26.72   DATE  08/05/72    TIME  16:07:51    PAGE    3

 SET 2)
 REJECTED BECAUSE;

  # OF +EQ CYCLES = 1
  # OF .EQ CYCLES = 1
  # OF MULTI-OCC. CYCLES = 1
  COCODE = 400000000000
```

```
PA06F.DEM    V07.26.72    DATE  08/05/72    TIME  16:07:51    PAGE    4
C
C !! Z2
C
      SUBROUTINE PA06F
      IMPLICIT INTEGER(A-Z)
      DIMENSION A(11,7,5)[(2)]
      DIMENSION B(11,7,5)[(2)]
      DIMENSION C(11,7,5)[(2)]
      DO 1 I=1,11,1
      DO 3 K=1,5,1
      DO 2 FOR ALL (J)/[1,...,7]
      A(I,J,K)=B(I,J,K)
      B(I,J,K)=A(I,J,K+N)
      C(I,J,K)=FOO
      C(I+M,J,K)=FEE
    2 CONTINUE
    3 CONTINUE
    1 CONTINUE
      RETURN
      END

 STATISTICS OF INTEREST ---

   STRIP MINEABLE?  YES
   OLD FREQUENCY:  1548
   NEW FREQUENCY:  220
```

```
PA06F.DEM    V07.26.72    DATE  08/05/72    TIME  16:07:51    PAGE    5

 SET Z1
 REJECTED BECAUSE:

  # OF 4E32 CYCLES = 1
  # OF 4E2 CYCLES = 1
  # OF MULTI-OCC. CYCLES = 1
  CDCODE = 400000000000
```

```
PA06F.DEM    V07.26.72    DATE  08/05/72    TIME  16:07:51    PAGE    6

 SET Z23
 REJECTED BECAUSE:

  # OF 4E32 CYCLES = 1
  # OF 4E2 CYCLES = 1
  # OF MULTI-OCC. CYCLES = 1
  CDCODE = 400000000000
```

```
PA06F.DEM    V07.26.72    DATE  08/05/72    TIME  16:07:51    PAGE    7

 SET Z13
 REJECTED BECAUSE:

  # OF 4E32 CYCLES = 2
  # OF 4E2 CYCLES = 2
  # OF MULTI-OCC. CYCLES = 2
  CDCODE = 400000000000
```

```
PA06F.DEM    V07.26.72    DATE  08/05/72    TIME  16:07:51    PAGE    8

 SET Z12
 REJECTED BECAUSE:

  # OF 4E32 CYCLES = 1
  # OF 4E2 CYCLES = 1
  # OF MULTI-OCC. CYCLES = 1
  CDCODE = 400000000000
```

```
PA06F.DEM    V07.26.72    DATE  08/05/72    TIME  16:07:51    PAGE    9

 SET Z123
 REJECTED BECAUSE:

  # OF 4E32 CYCLES = 3
  # OF 4E2 CYCLES = 2
  # OF MULTI-OCC. CYCLES = 3
  CDCODE = 400000000000
```

```
PA06F.DEM   V07.26.72   DATE  08/05/72    TIME  16:07:51    PAGE        4
C
C !! Z2
C
      SUBROUTINE FA06F
      IMPLICIT INTEGER(A-Z)
      DIMENSION A(11,7,5)[(2)]
      DIMENSION B(11,7,5)[(2)]
      DIMENSION C(11,7,5)[(2)]
      DO 1 I=1,11,1
      DO 3 K=1,5,1
      DO 2 FOR ALL (J)/[1,2...7]
      A(I,J,K)=B(I,J,K)
      B(I,J,K)=A(I,J,K+N)
      C(I,J,K)=FOO
      C(I+M,J,K)=FEE
    2 CONTINUE
    3 CONTINUE
    1 CONTINUE
      RETURN
      END

 STATISTICS OF INTEREST ---

   STRIP MINEABLE?  YES
   OLD FREQUENCY:   1540
   NEW FREQUENCY:   220


PA06F.DEM   V07.26.72   DATE  08/05/72    TIME  16:07:51    PAGE        5

 SET Z1
 REJECTED BECAUSE:

  # OF 4E32 CYCLES = 1
  # OF 4E2 CYCLES = 1
  # OF MULTI-OCC.  CYCLES = 1
  CDCODE = 400000000000


PA06F.DEM   V07.26.72   DATE  08/05/72    TIME  16:07:51    PAGE        6

 SET Z23
 REJECTED BECAUSE:

  # OF 4E32 CYCLES = 1
  # OF 4E2 CYCLES = 1
  # OF MULTI-OCC.  CYCLES = 1
  CDCODE = 400000000000


PA06F.DEM   V07.26.72   DATE  08/05/72    TIME  16:07:51    PAGE        7

 SET Z13
 REJECTED BECAUSE:

  # OF 4E32 CYCLES = 2
  # OF 4E2 CYCLES = 2
  # OF MULTI-OCC.  CYCLES = 2
  CDCODE = 400000000000


PA06F.DEM   V07.26.72   DATE  08/05/72    TIME  16:07:51    PAGE        8

 SET Z12
 REJECTED BECAUSE:

  # OF 4E32 CYCLES = 1
  # OF 4E2 CYCLES = 1
  # OF MULTI-OCC.  CYCLES = 1
  CDCODE = 400000000000


PA06F.DEM   V07.26.72   DATE  08/05/72    TIME  16:07:51    PAGE        9

 SET Z123
 REJECTED BECAUSE:

  # OF 4E32 CYCLES = 2
  # OF 4E2 CYCLES = 2
  # OF MULTI-OCC.  CYCLES = 2
  CDCODE = 400000000000
```

```
PA06G.DEM   V07.26.72   DATE 08/07/72   TIME 13:21:22   PAGE   1
          SUBROUTINE PA06G
C PA06G:  CATEGORY 06 (DATA DEPENDENCIES), CASE G.
C
          REAL A(10,10),B(10,10),OMEGA
          INTEGER I,J,K
C
C THIS CASE CONTAINS A FORTRAN LOOP (WITH PHONY LIMITS) EM-
C   PLOYING A TYPICAL NUMERICAL APPROXIMATION TO THE
C   SOLUTION OF A CLASSICAL MATHEMATICAL EQUATION (THE
C   UBIQUITOUS LAPLACE).  NONE OF THE DO-FOR-ALL INDEX SETS ARE
C   PARALYZABLE USING THE CURRENTLY IMPLEMENTED TECHNIQUES.
C   PARALYSIS IS OBTAINABLE WITH THE 'HYPERPLANE METHOD'.
C         .
C         .
C         .
          DO 100 I=1,25
          DO 1 J=2,9
          DO 2 K=2,9
          A(J,K)=.25*(A(J,K+1)+A(J-1,K)
          1         +A(J+1,K)+A(J,K-1))*OMEGA
          2         +(1.-OMEGA)*A(J,K)
2         CONTINUE
1         CONTINUE
100       CONTINUE
C         .
C         .
C         .
          RETURN
          END
```

```
PA06G.DEM   V07.26.72   DATE 08/07/72   TIME 13:21:22   PAGE   2

SET Z0
REJECTED BECAUSE:

 DIBAD = 2,0,0
```

```
PA06G.DEM   V07.26.72   DATE 08/07/72   TIME 13:21:22   PAGE   3

MIDSTREAM PARALYSIS REPORT --

 DIMENSION OF DO NEST:  3
 INDEX SET:  (I,J,K)
 ENDING LABEL OF DO NEST:  100
 STILL GOOD DO-FOR-ALL SETS:
   (J,K)
   (J)
   (K)

<F,G> SETS --

 1)  <F,G> = <A(J,K+1)/21,A(J,K)/21>
 2)  <F,G> = <A(J-1,K)/21,A(J,K)/21>
 3)  <F,G> = <A(J+1,K)/21,A(J,K)/21>
 4)  <F,G> = <A(J,K-1)/21,A(J,K)/21>
 5)  <F,G> = <A(J,K)/21,A(J,K)/21>

VALUES FOR <F,G> SETS --

 1)    +0-     0    1
 2)    +0-    -1    0
 3)    +0-     1    0
 4)    +0-     0   -1
 5)    +0-     0    0
```

```
PA06G.DEM   V07.26.72   DATE 08/07/72   TIME 13:21:22   PAGE   4

SET Z3
REJECTED BECAUSE:

 # INTRA-STMT CYCLES = 1
 CDCODE = 400000000004
```

```
PA06G.DEM    V07.26.72    DATE  08/07/72    TIME  13:21:22    PAGE    5

SET 22
REJECTED BECAUSE:

 * INTRA-STMT CYCLES = 1
 LOCCODE = 400000000004
```



```
PA06G.DEM    V07.26.72    DATE  08/07/72    TIME  13:21:22    PAGE    6

SET 223
REJECTED BECAUSE:

 * INTRA-STMT CYCLES = 2
 LOCCODE = 400000000004
```

```
PA07A.DEM    V07.26.72    DATE  08/05/72    TIME  16:09:45    PAGE    1
     1              SUBROUTINE PA07A
     2  C PA07A:  CATEGORY 07 (MISCELLANEOUS), CASE A.
     3  C
     4              IMPLICIT INTEGER (A-Z)
     5              DIMENSION A(10,20)
     6  C
     7  C CAN ONLY SIM ON (J) SINCE (I) SUBSCRIPTS NOT OF PROPER FORM
     8  C   THE BACKUP FACILITIES OF THE PARALYZER WILL BE TESTED.
     9  C           .
    10  C           .
    11  C           .
    12              DO 1 I=1,10
    13              DO 1 J=1,20
    14  1           A(2*I+N,J)=A(3+I+Q,J)+1
    15              RETURN
    16              END
 !R106! REJECT SINCE INVALID SUBSCRIPT   [LINE 14]
 !R106! REJECT SINCE INVALID SUBSCRIPT   [LINE 14]
PERIL BACKING UP TO REMOVE OUTER DO STMNT FROM INDEX SET
```

```
PA07A.DEM    V07.26.72    DATE  08/05/72    TIME  16:09:45    PAGE    2

 MIDSTREAM PARALYSIS REPORT --

  DIMENSION OF DO NEST:  1
  INDEX SET:  (J)
  ENDING LABEL OF DO NEST:  1
  STILL GOOD DO-FOR-ALL SETS:
    (J)
```

```
PA07A.DEM    V07.26.72    DATE  08/05/72    TIME  16:09:45    PAGE    3
C
C !! Z1
C
      SUBROUTINE PA07A
      IMPLICIT INTEGER(A-Z)
      DIMENSION A(10,20)[(2)]
      DO 1 I=1,10
      DO 1 FOR ALL (J)/[1,2...20]
      A(2*I+N,J)=A(3+I+Q,J)+1
    1 CONTINUE
      RETURN
      END

 STATISTICS OF INTEREST ---

   STRIP MINEABLE?  YES
   OLD FREQUENCY:  20
   NEW FREQUENCY:  1
```

PA07B.DEM V07.26.72 DATE 08/05/72 TIME 16:10:20 PAGE 1

```
      SUBROUTINE PA07B(E,F,NF)
C PA07B:  CATEGORY 07 (MISCELLANEOUS), CASE B.
C
      IMPLICIT INTEGER (A-Z)
      DIMENSION A(10),B(10),C(10),D(10,2)[(2,1)],E(10)
      COMMON A
      EQUIVALENCE (B,C)
      DIMENSION F(NF)
      DIMENSION G(2,10)[(1),(2)]
C
C WARNINGS FROM PARALYZER ABOUT POTENTIAL
C   ALLOCATION STORAGE CONFLICTS ARE MADE.
C     .
C     .
C     .
      DO 1 I=1,10
      A(I)=0
      B(I)=1
      C(I)=0
      D(I,2)=0
      E(I)=0
      F(I)=0
      G(1,I)=0
    1 CONTINUE
      RETURN
      END
```

PA07B.DEM V07.26.72 DATE 08/05/72 TIME 16:10:20 PAGE 2

```
MIDSTREAM PARALYSIS REPORT --

 DIMENSION OF DO NEST:  1
 INDEX SET:  (I)
 ENDING LABEL OF DO NEST:  1
 STILL GOOD DO-FOR-ALL SETS:
   (I)
```

PA07B.DEM V07.26.72 DATE 08/05/72 TIME 16:10:20 PAGE 3

```
C
C !! Z1
C
      SUBROUTINE PA07B(E,F,NF)
      IMPLICIT INTEGER(A-Z)
      DIMENSION E(10)[(1)],F(NF)[(1)],A(10)[(1)],B(10)[(1)],C(10)[(1)]
      DIMENSION D(10,2)[(2,1)]
      DIMENSION G(2,10)[(1),(2)]
      COMMON A
      EQUIVALENCE (B,C)
      DO 1 FOR ALL (I)/[1,2..10]
      A(I)=0
      B(I)=1
      C(I)=0
      D(I,2)=0
      E(I)=0
      F(I)=0
      G(1,I)=0
    1 CONTINUE
      RETURN
      END

 STATISTICS OF INTEREST ---

  STRIP MINEABLE?  YES
  OLD FREQUENCY:  70
  NEW FREQUENCY:  7
  ALLOCATION WARNINGS FOR THESE ARRAYS-
    A  !COMMON STORAGE!
    B  !EQUIVALENCE CONFLICTS!
    C  !EQUIVALENCE CONFLICTS!
    D  !INTRA-PROGRAM LOCAL STORAGE CONFLICT!
    E  !DUMMY FORMAL PARAMETER!
    F  !DUMMY FORMAL PARAMETER!
    F  !ADJUSTABLE DIMENSIONS!
```

```
.RUN 17

*
PAOPT.DEM/G/I/J
 IVTRAN: PAOPT.DEM/GIJ
              SUBROUTINE PAOPT
      C SPECIAL EXAMPLE.
      C
              IMPLICIT INTEGER (A-Z)
              DIMENSION A(15),B(20),C(15)
      C
      C A SPECIAL EXAMPLE TO BE RUN WITH APPROPRIATE COMPILER OPTIONS
      C    SUCH THAT THE INTERACTIVE DEBUGGING AIDS FACILITY OF THE
      C    PARALYZER WILL BE INVOKED.
      C       .
      C       :
      C       :
              DO 75 I=2,N
              B(I+P)=A(Q+I)/C(I-1)
              A(I+Q)=B(P+I)*C(I+1)
   75         CONTINUE
              RETURN
              END
 PERIL CTRL PLEASE:
D=1
***  TRACE 1  ***

65,55
      SUBROUTINE PAOPT
      IMPLICIT INTEGER(A-Z)
      DIMENSION A(15),B(20),C(15)
      DO 75 I=2,N
      B(I+P)=A(Q+I)/C(I-1)
      A(I+Q)=B(P+I)*C(I+1)
   75 CONTINUE
      RETURN
      END

DUMP OF WHOLE PROGRAM

    1 SEQNO 1
    4 ENTRY PAOPT
   10 SEQNO 13
   13 DO    75,1,2,N
   23 SEQNO 14
   94 STORE C:40,C:65
  103 SEQNO 15
  169 STORE C:115,C:160
  178 SEQNO 16
  181 LABEL 75
  187 CONTI
  190 SEQNO 17
  193 RETUR
  196 SEQNO 18
  199 END

   40 ARRAY B,I+P
   58 ARRAY A,I+Q
   76 ARRAY C,I-1
   85 OVER  C:55,C:76
  115 ARRAY A,I+Q
  133 ARRAY B,I+P
  151 ARRAY C,I+1
  160 TIMES C:133,C:151

***  TRACE 2  ***

60,63

DI/IX TABLES DUMP  PART 1

IX VAR      LAB  D CHN  LAM LLO     ULLO    L      FL

1 1              1  0    0                  2      2

DISIM = 0, DISEQ = 0, DIPA = C:10, DISON = C:190

DI/IX TABLES DUMP, PART 2

IX DBIG NEQU SUBS BAD SENO  MU UUP    ULUP   U      FU

1   0    0    0    0   13    1               N*     N

LB/QU TABLES DUMP

IX CHN LIF      STOR    GEN USE  G DPRE SENO QU ORGO   LAB CODE

1  2            C:214    0   0   0   1    14  0   0     000000000000
2  0            C:223    0   0   0   1    15  0   0     000000000000

  214 STORE C:40,C:65
  223 STORE C:115,C:160

   40 ARRAY B,I+P
   58 ARRAY A,I+Q
   76 ARRAY C,I-1
   85 OVER  C:58,C:76
  115 ARRAY A,I+Q
  133 ARRAY B,I+P
  151 ARRAY C,I+1
  160 TIMES C:133,C:151
***  TRACE 3  ***

51,54

DUMP OF MATRICES

DUMP OF P<< MATRIX  2X2

   1  200000000000
   2  000000000000

DUMP OF MATRICES

DUMP OF P<< CLOSE  PTS

   3  200000000000
   4  000000000000
```

```
*** TRACE 7 ***

58,64

AR/PU TABLES DUMP

 IX NAME    ND GEN  USE PUORSS PUDO

  1 B        1  1    5      1 I
  2 A        1  4    2      1 I
  3 C        1  0    3      1 I

OC/SS TABLES DUMP

 IX OP       STMT GFLG ADIM SDIM FF LL SUBS SSLOCO  SSCHN VAR

  1 C:40      1    1    0    0   0  0   1 P          0 B
  2 C:58      1    0    0    3   0  0   2 Q          1 A
  3 C:76      1    0    6    0   0  0   3 -1         2 C
  4 C:115     2    1    0    0   0  0   4 Q          3 A
  5 C:133     2    0    0    6   0  0   5 P          4 B
  6 C:151     2    0    0    0   0  0   6 1          5 C

  40 ARRAY B,1+P
  58 ARRAY A,1+Q
  76 ARRAY C,I-1
 115 ARRAY A,1+Q
 133 ARRAY B,1+P
 151 ARRAY C,1+1
*** TRACE 8 ***

60

DI/DX TABLES DUMP  PART 1

IX VAR       LAB  D CHN  LAM LLO    ULLO    L.      FL.

 1 I              1  0    0 2     C:265    2       2

DISIM = 0, DISEQ = 0, DIPA = C:10, DISON = C:190

 265       N-2

DI/DX TABLES DUMP, PART 2

IX DBIG NEQU SUBS BAD SENO   MU UUP    ULUP   U      FU

 1  15    0    6   0   13    1 N     C:268  N      N

 268       N-2

*** TRACE 9 ***

59

CD/DS/US TABLES DUMP

IX  DS   V1 V2 US            V5  V3            V4            CODE

 1 I      0  0               1 000000000000 000000000000 000000000000
*** TRACE 10 ***

62

<F,G> SETS --

  1) <F,G> = <B(I+P)/15,B(I+P)/14>
  2) <F,G> = <A(I+Q)/14,A(I+Q)/15>

VALUES FOR <F,G> SETS --

  1)       0
  2)       0
*** TRACE 12 ***

52,53

DUMP OF MATRICES

DUMP OF P< MATRIX  6X6

   5  020000000000
   6  440000000000  400000000000  000000000000  040000000000
  10  040000000000

DUMP OF MATRICES

DUMP OF P< CLOSURE MATRIX  6X6

  11  060000000000
  12  460000000000  460000000000  000000000000  040000000000
  16  040000000000
*** TRACE 13 ***

*** TRACE 14 ***
```

```
*** TRACE 15 ***

61,63

DO TABLE DUMP

IX FLAG LAB BEFO      AFTR     DI CHN   D VAR/MIX LAM/SETC MU/LAEX

 1  0    75           C:310    1  0.    I                  C:307

DOPRE = , DOPOST =

  310 LABEL 75
  316 CONTI

  148 LE    I,N-1
  304 SELEC I,K:13,C:148
  307       C:304

LB/QU TABLES DUMP

IX CHN L1F        STOR     GEN USE  G DPRE SENO QU ORGO   LAB QCOD

 1  2             C:214     1   2   1   1   14   0   1        000000000000
 2  0             C:223     4   5   2   1   15   0   2        000000000000

 214 STORE C:40,C:85
 223 STORE C:115,C:160

  40 ARRAY B,I+P+1
  58 ARRAY A,I+Q+1
  76 ARRAY C,I
  85 OVER  C:58,C:76
 115 ARRAY A,I+Q+I
 133 ARRAY B,I+P+I
 151 ARRAY C,I+2
 160 TIMES C:133,C:151
```

```
*** TRACE 16 ***

39,55,65

OCTAL DUMP OF TABLE K

  1 000005000000 040000010004 000320000174 000000000002 000000000000
  6 040000010010 000367000270 000000000001 000001000000 040000010001
 11 000356000356 777777777777 000011000000 070000100001 000315000315
 16 400000000013 000000000001 000000000001 000000000015

DUMP OF WHOLE PROGRAM

   1 SEQNO 1
   4 ENTRY PAOPT
  10 SEQNO 13
 319 FORAL 75,I,C:304
 325 SEQNO 10001
 214 STORE C:40,C:85
 328 SEQNO 10002
 223 STORE C:115,C:160
 331 SEQNO 10003
 310 LABEL 75
 316 CONTI
 190 SEQNO 17
 193 RETUR
 196 SEQNO 18
 199 END

  40 ARRAY B,I+P+1
  58 ARRAY A,I+Q+1
  76 ARRAY C,I
  85 OVER  C:58,C:76
 115 ARRAY A,I+Q+I
 133 ARRAY B,I+P+I
 148 LE    I,N-1
 151 ARRAY C,I+2
 160 TIMES C:133,C:151
 304 SELEC I,K:13,C:148
      SUBROUTINE PAOPT
      IMPLICIT INTEGER(A-Z)
      DIMENSION A(15)[(1)],B(20)[(1)],C(15)[(1)]
      DO 75 FOR ALL (I)/[(1)/(1,2...13)]:I.LE.N-1]
      B(P+I+1)=A(Q+I+1)/C(I)
      A(Q+1+1)=B(P+I+1)*C(I+2)
  75  CONTINUE
      RETURN
      END

 STATISTICS OF INTEREST ---

  STRIP MINEABLE?   YES
  OLD FREQUENCY:   2*(N-1)
  NEW FREQUENCY:   2
*** TRACE 302 ***

 EXITING PERIL
```

4. Future Activities

Paralyzer activities for the remainder of 1972 will continue along present lines with no 'major' design or implementation efforts proposed until 1973. Activities which will be engaged in through December 1972 are described below.

4.1 User Programs

A collection of real user programs has been provided Massachusetts Computer Associates, Inc. from outside sources. These programs (at least sections of these programs) will be run through the Phase I Paralyzer. Results of these runs will be used to look for design deficiencies which may exist in Phase I so that appropriate enhancements may be incorporated into the Phase II (December 1972) version.

4.2 Improvements and Enhancements

In addition to the enhancements which may evolve from the activities described in 4.1, specific improvements of a 'minor' nature are planned for December 1972. These include:

a) improvements to the forward transfer elimination and scalar removal procedures pending completion of the flow analyzer program, which will soon be available to all phases of the IVTRAN compiler, .

b) some makeshift facility to select a 'final rewriting' of the loop after the exhaustive analysis cycle,

c) elimination of some present tight nesting deficiencies,

d) a possible alteration of the strategy of converging on DO nests for paralysis. (The current "outer-to-inner" and "sequential order of appearance" processing of DO loops is not expected to yield the best rewritings; it is designed to cover all the DO loops of a program relatively independently of one another. If the results of "exhaustive analysis" of actual programs, as described in Section 4.1, indicate some consistently better approach, this will be implemented.),

e) introduction of temporary arrays to resolve conflicts of allocation arising from the paralysis of disjoint loops, and

f) generation of OVERLAP specifications to minimize the total storage used by temporary arrays.

4.3 Documentation

Program unit documentation for the permanent sections of the Phase I Paralyzer will be completed. Most of these sections have been documented, but very little exists currently in a publishable form.

5. References

[1] Lamport, Leslie: The Detection of Parallelism in FORTRAN DO Loops. CAID-7110-0111, Applied Data Research, Inc., Wakefield, Mass., October 1971.

[2] Lamport, Leslie; Presberg, David: Concurrent Compiling, Volume II - The Parallel Execution of FORTRAN DO Loops. CADD-7112-2712, Applied Data Research, Inc., Wakefield, Mass., December 1971. RADC-TR-72-64, Volume II, Final Technical Report, March 1972.

[3] Lamport, Leslie: The Parallel Execution of FORTRAN DO Loops, CA-7202-2711, Applied Data Research, Inc., Wakefield, Mass., February 1972. (Submitted for publication to the CACM.)

[4] Third Semi-Annual Technical Report (13 January 1971 - 13 July 1971) for the Project Compiler Design for ILLIAC IV, CADD-7110-0511, Applied Data Research, Inc., Wakefield, Mass., October 1971, AD 893 168 L.

[5] Fourth Semi-Annual Technical Report (14 July 1971 - 13 January 1972) for the Project Compiler Design for ILLIAC IV, CADD-7207-1111, Volumes I and II, Applied Data Research, Inc., Wakefield, Mass., July 1972.