stichting
mathematisch
centrum

MC

AFDELING MATHEMATISCHE BESLISKUNDE BW 51/75 JUNE

JAC.M. ANTHONISSE & B.J. LAGEWEG

GRAPHLIB Ø:
PROCEDURES TO REPRESENT, GENERATE AND ANALYSE GRAPHS

2e boerhaavestraat 49 amsterdam

Printed at the Mathematical Centre, 49, 2e Boerhaavestraat, Amsterdam.

The Mathematical Centre, founded the 11-th of February 1946, is a non-profit institution aiming at the promotion of pure mathematics and its applications. It is sponsored by the Netherlands Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O), by the Municipality of Amsterdam, by the University of Amsterdam, by the Free University at Amsterdam, and by industries.

AMS(MOS) subject classification scheme (1970): 68K15,68U65,68U75,94A20

ACM -Computing Reviews- categories: 3.30, 3.35, 3.37, 4.22, 5.32

GRAPHLIB ∅: procedures to represent, generate and analyse graphs

by

Jac.M. Anthonisse & B.J. Lageweg

SUMMARY

The library GRAPHLIB ∅ of procedures to represent, generate and analyse graphs is based upon the graph defining language (report BW 30/73, Mathematisch Centrum, Amsterdam).

Both that language and the library have been designed for use by people with a very limited knowledge of computers and programming.

KEY WORDS & PHRASES: *Graphs, networks, social sciences, graph-representations, graph-generators, graph-analysis*

0. CONTENTS

The numbers in parentheses are the codenumbers of the procedures in the library GRAPHLIB0, as implemented in ALGOL 60 (version 3.0) on the CYBER 73-28 of the SARA Computing Centre at Amsterdam.

1. INTRODUCTION

The main part of this report is a printout (June 24, 1975) of the contents of a file serving as a manual for the use of the library GRAPHLIBO which contains a set of ALGOL 60 procedures to represent, generate and analyse graphs and networks.

The library is a growing one, due to new applications of graphs leading to new types of analyses, which are implemented either as new procedures or as extensions of existing procedures. Moreover, a number of procedures is being developed now and others are considered for implementation. Also, the efficiency of some of the existing procedures can and will be enhanced and some adjustments will improve the presentation of their output. Thus, for actual use of the library the up to date manual should be consulted.

The library is based upon the graph defining language [1]. Graphs and networks are required to be presented according to the rules of that language. Both the language and the library have been designed for use by people with a very limited knowledge of computers and programming. Experience has shown that they can indeed analyse their graphs effectively. Experience also has shown that a more general data description language could be very useful, and that the processing of very large graphs generates specific problems.

It follows from the above that this report should not be considered as a final description of the library. Such a final report can be produced if no new developments are expected, but then the report would be like an obituary, with a very limited use.

The first steps towards the development of the library were taken in 1970, when the first author became involved, as a consultant, in a study of interlocking directorates between Dutch industrial concerns, financial institutions, the Social Economic Council and the central government. This study was directed by prof.dr. R.J. Mokken and drs. F.N. Stokman of the Institute of Political Research of the University of Amsterdam. The programs for this study were written on an ad-hoc basis, and run on the Electrologica X8 computer of the Mathematical Center. When new studies using graphs were started and the CDC Cyber 73-28 computer became available we decided to develop a graph defining language and a coherent set of programs, resulting in the present library.

We gratefully acknowledge the collaboration and the many stimulating discussions with staff and students of the Institute of Political Research, especially with M. Boon, M. Fenneman, H.M. Helmers, R.J. Mokken, R.C. Plijter and F.N. Stokman, which had a non-trivial influence on the contents of the library.

*** 2. INPUT, KEYWORDS, OUTPUT ***

EACH OF THE PROCEDURES DESCRIBED IN THIS REPORT HAS A HEADING OF THE FOLLOWING FORM :

```
  "PROCEDURE" IDENTIFIER(GRAPHIN,JOBSIN);
  "VALUE" GRAPHIN,JOBSIN; "INTEGER" GRAPHIN,JOBSIN;
```

BOTH GRAPHIN AND JOBSIN ARE CHANNEL-NUMBERS, AND IDENTIFY AN INPUT-CHANNEL.

INPUT FOR EACH OF THE PROCEDURES CONSISTS OF TWO PARTS :
1. THE GRAPH OR NETWORK TO BE ANALYSED IS READ FROM CHANNEL GRAPHIN,
2. THE JOBS DEFINING THE ANALYSES ARE READ FROM CHANNEL JOBSIN.

TWO WAYS OF PRESENTING THE GRAPH OR NETWORK ARE DISTINGUISHED :
1.1. THE EXTERNAL REPRESENTATION,
      I.E. A SET OF PARAMETERS FOLLOWED BY A GRAPH, BOTH SATISFYING
           THE RULES OF THE GRAPH-DEFINING LANGUAGE AS DEFINED IN [1].

      THE EXTERNAL REPRESENTATION MAY BE DISTRIBUTED OVER SEVERAL INPUT-CHANNELS.   IF THE TEXT

```
      'CHANNEL' IN := <CHANNELNUMBER>;
```

      IS ENCOUNTERED THEN THE NEXT PART OF THE INPUT IS READ FROM THE CHANNEL IDENTIFIED BY <CHANNELNUMBER>.
      THAT CHANNEL MAY CONTAIN A SWITCH TO ANOTHER CHANNEL, ETC.

1.2. THE INTERNAL REPRESENTATION,
      I.E. AS THE OUTPUT OF AN EXECUTION OF THE PROCEDURE STOREGRAPH1.

ALL THE PROCEDURES DESCRIBED HERE, EXCEPT STOREGRAPH1, REQUIRE THE GRAPH TO BE PRESENTED IN ITS INTERNAL REPRESENTATION.   ONLY STOREGRAPH1 ACCEPTS BOTH THE EXTERNAL AND THE INTERNAL REPRESENTATION..

EACH JOB BEGINS WITH

```
'JOB'  <KEYWORDS> ;
```

IN MOST PROCEDURES THE KEYWORDS COMPLETELY DEFINE THE JOB, IN SEVERAL CASES, HOWEVER, FURTHER SPECIFICATIONS ARE REQUIRED. A JOB THEN CONSISTS OF

```
'JOB'  <KEYWORDS> ;
<FURTHER SPECIFICATIONS>
'JOBFIN'
```

EACH PROCEDURE HAS ITS OWN KEYWORDS AND ITS OWN REQUIREMENTS FOR FURTHER SPECIFICATIONS, TO BE DESCRIBED IN THE NEXT SECTIONS.

THE SEQUENCE OF JOBS SHOULD BE TERMINATED BY

```
'STOP'
```

THE KEYWORD

OUT = <CHANNELNUMBER>

CAN BE USED IN EACH JOB OF EACH PROCEDURE TO SPECIFY THE
OUTPUT CHANNEL FOR THE RESULTS OF THAT JOB.
DEFAULT: OUT = 64

APART FROM THE RESULTS OF THE ANALYSES, WHICH ARE SEND TO THE CHANNEL
IDENTIFIED BY THE VALUE OF THE KEYWORD OUT, EACH PROCEDURE REPRODUCES
(PART OF) ITS INPUT AS READ FROM THE CHANNELS GRAPHIN AND JOBSIN.
THIS PART OF THE OUTPUT IS SEND TO THE CHANNEL WITH CHANNELNUMBER = 64,
AND CONSISTS OF :
IF THE GRAPH IS IN EXTERNAL REPRESENTATION THEN A LITERAL REPRODUCTION
OF THIS REPRESENTATION, POSSIBLY INTERSPERSED WITH ERROR MESSAGES
AND WARNINGS, FOLLOWED BY A SURVEY OF ITS MINIMAL PARAMETERS.
IF THE GRAPH IS IN INTERNAL REPRESENTATION THEN ITS TITLE AND A SURVEY
OF THE VALUE OF THE PARAMETERS CORRESPONDING TO THE KEYWORDS OF
PROCEDURE STOREGRAPH1.
A LITERAL REPRODUCTION OF THE JOBS.

3. REPRESENTATIONS

Graphs can be represented in many different ways, the four procedures to be described now produce various representations, to be used by different readers.

STOREGRAPH1 produces a representation that is completely oriented to the library and the computer, this representation does not satisfy the graph-defining language.

PRINTGRAPH1 produces representations satisfying the graph-defining language, the representation is readable both for human beings and the computer.

PRINTGRAPH2 produces a fixed-format representation, not satisfying the graph-defining language, to be read by computers, but not by procedures of this library.

PRINTARCS1 produces a survey of the incidence and adjacency relations, not satisfying the graph-defining language, to be read by human beings.

*** 4. STOREGRAPH 1 ***

THE PURPOSE OF STOREGRAPH1 IS TO STORE THE INTERNAL REPRESENTATION OF A GRAPH ON A FILE, I.E. ON THE OUTPUT-CHANNEL.

THE KEYWORDS OF STOREGRAPH1 ARE :

IVM
  THE VERTEX-INFORMATIONS ARE NOT STORED

IVLM
  THE VERTEX-INFORMATION LIST-IDENTIFIERS ARE NOT STORED,
  THE VERTEX-INFORMATIONS, IF ANY, ARE STORED AS A SINGLE LIST,
  WITH 1 AS ITS IDENTIFIER

IAM
  THE ARC-INFORMATIONS ARE NOT STORED

IALM
  THE ARC-INFORMATION LIST-IDENTIFIERS ARE NOT STORED,
  THE ARC-INFORMATIONS, IF ANY, ARE STORED AS A SINGLE LIST,
  WITH 1 AS ITS IDENTIFIER

VM
  THE VERTICES AND THE ARCS ARE NOT STORED

VLM
  THE VERTEX LIST-IDENTIFIERS ARE NOT STORED,
  THE VERTICES, IF ANY, ARE STORED AS A SINGLE LIST,
  WITH 1 AS ITS IDENTIFIER

AM
  THE ARCS ARE NOT STORED

ALM
  THE ARC LIST-IDENTIFIERS ARE NOT STORED,
  THE ARCS, IF ANY, ARE STORED AS A SINGLE LIST,
  WITH 1 AS ITS IDENTIFIER

*** 5. PRINTGRAPH 1 ***

THE PURPOSE OF PRINTGRAPH1 IS TO OBTAIN READABLE PRESENTATIONS OF GRAPHS AND NETWORKS.

THE KEYWORDS OF PRINTGRAPH1 ARE:

COMPACT
  IF A JOB CONTAINS THIS KEYWORD THEN THE OTHER KEYWORDS ARE IGNORED. THE PROGRAM PRODUCES A REPRESENTATION IN WHICH VINFOS, AINFOS AND VERTICES ARE CODED. EACH LIST OF ENTITIES STARTS ON A NEW LINE.

CODEDVINF, CODEDAINF, CODEDVERT
  IF A JOB CONTAINS ANY OF THESE KEYWORDS THEN THE ENTITIES CORRESPONDING WITH THAT KEYWORD (VINFOS, AINFOS AND VERTICES RESPECTIVELY) ARE CODED.
  MOREOVER, IN THE CORRESPONDING LISTS OF ENTITIES EACH ENTITY IS PRINTED ON A NEW LINE AND PRECEDED BY ITS LABEL.
  THE LABEL IS INSERTED AS A COMMENT.

FORM =
  DEFAULT: FORM = 0
  THIS KEYWORD CONTROLS THE PRESENTATION OF THE LISTS OF ARCS.
  IF FORM=0 THEN EACH LIST OF ARCS IS PRINTED IN THE STRAIGHTFORWARD FORM, EACH ARC IS PRINTED ON A NEW LINE.
  IF FORM=1 (2) THEN EACH LIST OF ARCS IS PRINTED IN THE FIRST (SECOND) ABBREVIATED FORM.
  FORM=0 LEAVES THE ORDER OF THE ARCS UNCHANGED, FORM=1 AND FORM=2 WILL, IN GENERAL, REARRANGE THE ARCS.

TRANSPOSE
  BY THIS KEYWORD THE HEAD AND TAIL OF EACH ARC ARE INTERCHANGED.
  DEFAULT: THE DIRECTION OF THE ARCS REMAINS UNCHANGED.

*** 6. PRINTGRAPH 2 ***

THE PURPOSE OF PRINTGRAPH 2 IS TO PRODUCE FIXED-FORMAT REPRESENTATIONS OF GRAPHS.

THE KEYWORDS OF PRINTGRAPH 2 ARE:

CODEDVINF
THE VERTEX-INFORMATIONS ARE CODED.
DEFAULT: THE VINFOS ARE NOT CODED.

CODEDAINF
THE ARC-INFORMATIONS ARE CODED.
DEFAULT: THE AINFOS ARE NOT CODED.

CODEDVERT
THE VERTICES ARE CODED.
DEFAULT: THE VERTICES ARE NOT CODED.

FORWARD, BACKWARD
EACH OF THESE KEYWORDS LEADS TO A REPRESENTATION OF ALL ARCS. THE REPRESENTATION OF AN ARC DUE TO FORWARD HAS THE OPPOSITE DIRECTION OF THE REPRESENTATION OF THE SAME ARC DUE TO BACKWARD.
A JOB MAY CONTAIN BOTH KEYWORDS.
DEFAULT: ONLY FORWARD IS EXECUTED.

PAIRS
THIS KEYWORD MAY INTERCHANGE THE TAIL AND HEAD OF ARCS.
IF PAIRS THEN IN THE LISTS OF ARCS PRODUCED UNDER "FORWARD" HEAD AND TAIL ARE INTERCHANGED IF THE LABEL OF THE TAIL EXCEEDS THE LABEL OF THE HEAD.
IF PAIRS THEN IN THE LISTS OF ARCS PRODUCED UNDER "BACKWARD" HEAD AND TAIL ARE INTERCHANGED IF THE LABEL OF THE HEAD EXCEEDS THE LABEL OF THE TAIL.
DEFAULT: UNDER FORWARD: THE DIRECTION OF THE ARCS REMAINS UNCHANGED,
UNDER BACKWARD: THE DIRECTION OF EACH ARC IS CHANGED.

OUTPUT FROM PRINTGRAPH 2 CONSISTS OF A NUMBER OF BLOCKS.

BLOCK	CONTAINS
0	THE TITLE AND ALL LIST IDENTIFIERS
1	THE LISTS OF VERTEX-INFORMATIONS
2	THE LISTS OF ARC-INFORMATIONS
3	THE LISTS OF VERTICES
4	THE LISTS OF ARCS PRODUCED UNDER FORWARD
5	THE LISTS OF ARCS PRODUCED UNDER BACKWARD

THE FIRST POSITION OF EACH LINE OF OUTPUT CONTAINS THE INDEX OF ITS BLOCK.
THE NEXT LW POSITIONS OF EACH LINE CONTAIN AN INTEGER CORRESPONDING TO THE TITLE OF THE GRAPH(INTEGER=0) OR TO A LIST IDENTIFIER(INTEGER=1,2,...,LN).
IN BLOCK 0 THIS LIST-INTEGER IS FOLLOWED BY THE TITLE OR BY THE LIST-IDENTIFIER IT CORRESPOND TO(TW POSITIONS).
IN THE OTHER BLOCKS THIS LIST-INTEGER IS FOLLOWED BY ANOTHER INTEGER, THE SEQUENCE NUMBER OF THE INFORMATION, VERTEX OR ARC WITHIN THE LIST CORRESPONDING TO THE LIST-INTEGER. THE SEQUENCE NUMBER OCCUPIES SW POSITIONS.

IN BLOCKS 1 AND 2 THE SEQUENCE NUMBER IS FOLLOWED BY THE LABEL OF AN INFORMATION(LVIW RESP LAIW POSITIONS) AND THE INFORMATION ITSELF (VIW RESP AIW POSITIONS).

IN BLOCK 3 THE SEQUENCE NUMBER IS FOLLOWED BY THE LABEL OF THE VERTEX (LVW POSITIONS), THE IDENTIFIER OF THE VERTEX(VW POSITIONS) AND POSSIBLY (THE LABEL OF) ITS INFORMATION(LVIW OR VIW POSITIONS).

IN BLOCKS 4 AND 5 THE SEQUENCE NUMBER IS FOLLOWED BY AN 1 (OR A 0), TAKING 2 POSITIONS, IF THE HEAD AND THE TAIL OF THE ARC WERE (NOT) INTERCHANGED DUE TO PAIRS, THE (LABELS OF THE) ENDPOINTS OF THE ARC (TOGETHER ARCW POSITIONS) AND POSSIBLY (THE LABEL OF) ITS INFORMATION(LAIW OR AIW POSITIONS).

A TABLE OF THE VALUES OF THESE QUANTITIES FOR EACH JOB IS OUTPUT ON THE MAIN OUTPUT CHANNEL.

*** 7. PRINTARCS 1 ***

THE PURPOSE OF PRINTARCS1 IS TO PRINT, FOR EACH LIST OF ARCS, THE ADJACENCIES OF EACH VERTEX.

OUTPUT FROM PRINTARCS1 CONSISTS OF FOUR COLUMNS, COLUMN 1 AND 2 OR COLUMN 2 AND 3 CONTAIN THE TAIL AND HEAD OF AN ARC, COLUMN 4 CONTAINS THE ARC-INFORMATION.

EACH ARC IS PRINTED TWICE,
ONCE ITS TAIL IS IN COLUMN 1 AND ITS HEAD IS IN COLUMN 2,
ONCE ITS TAIL IS IN COLUMN 2 AND ITS HEAD IS IN COLUMN 3.
THE ARCS ARE SORTED, AS IS EXPLAINED BY THE EXAMPLE:

```
'GRAPH'  EXAMPLE;
'VERTICES'  := 1,2,3,4,5;
'ARCS'  := 1:2, 4:4, 2:3, 3:5, 5:3, 4:1, 3:4;
'FINI'
```

RESULTING IN:

```
     1   2
 4   1
 1   2
     2   3
 2   3
     3   4
 5   3
     3   5
     4   1
 3   4
 4   4
     4   4
 3   5
     5   3
```

8. GENERATORS

Each of the procedures to be described now is a graph-generator, they can generate new graphs if the data from which the graph is generated itself has the form of a graph, the representation of the new graphs satisfies the graph-defining language.

New graphs can be generated by:
sorting of elements: LEXICO1 and LEXICO2,
redefining the direction of arcs: TRANSPOSE1,
selecting elements: SELECT1, SELECT2, INTAR2, NEWINFO1 and NEWINFO2,
induction: INDUCE1,
aggregation of arcs: MULARC1.

Several procedures will be added, e.g. for the aggregation of vertices.

*** 9. LEXICO 1 ***

THE PURPOSE OF LEXICO1 IS TO SORT THE LISTS OF VINFOS, AINFOS, VERTICES AND ARCS AND TO REDEFINE THE DIRECTION OF THE ARCS. THE ENTITIES CAN BE PUT INTO ASCENDING OR DESCENDING ORDER.

THE KEYWORDS OF LEXICO1 ARE:

VINFOS AZ, VINFOS ZA
THE VERTEX-INFORMATIONS ARE PUT INTO ASCENDING (AZ) OR DESCENDING (Z*) ORDER RESPECTIVELY.
DEFAULT: THE ORDER REMAINS UNCHANGED.

AINFOS AZ, AINFOS ZA
THE ARC-INFORMATIONS ARE PUT INTO ASCENDING (AZ) OR DESCENDING (ZA) ORDER RESPECTIVELY.
DEFAULT: THE ORDER REMAINS UNCHANGED.

VERT AZ, VERT ZA
THE VERTICES ARE PUT INTO ASCENDING (AZ) OR DESCENDING (ZA) ORDER RESPECTIVELY.
DEFAULT: THE ORDER REMAINS UNCHANGED.

ARCS TH AZ , ARCS TH ZA , ARCS HT AZ , ARCS HT ZA
ARCS TAZ HZA, ARCS TZA HAZ, ARCS HAZ TZA, ARCS HZA TAZ
EACH KEYWORD DEFINES AN ARRANGEMENT OF THE ARCS.
DEFAULT: THE ORDER OF THE ARCS REMAINS UNCHANGED.
THE ARCS ARE SORTED IN TWO STEPS.
IF, WITHIN A KEYWORD, THE "T" PRECEDES THE "H" THEN THE ARCS ARE SORTED ACCORDING TO THE TAILS IN STEP 1 , ARCS WITH IDENTICAL TAILS ARE SORTED ACCORDING TO THE HEADS IN STEP 2 .
IF, WITHIN A KEYWORD, THE "H" PRECEDES THE "T" THEN STEP 1 CONSISTS OF SORTING THE ARCS ACCORDING TO THE HEADS, IN STEP 2 THE ARCS WITH IDENTICAL HEADS ARE SORTED ACCORDING TO THE TAILS.
TAILS AND HEADS ARE PUT INTO ASCENDING (AZ) OR DESCENDING (ZA) ORDER RESPECTIVELY.
THE FIRST FOUR KEYWORDS PUT BOTH TAILS AND HEADS INTO ASCENDING (OR DESCENDING) ORDER.
THE LAST FOUR KEYWORDS PUT THE TAILS AND HEADS INTO OPPOSITE ORDER.
THESE EIGHT KEYWORDS LEAVE THE DIRECTIONS OF THE ARCS UNCHANGED.

EP AZ, EP ZA
THESE KEYWORDS MAY MODIFY THE DIRECTION OF ARCS.
FOR EACH ARC THE TAIL AND HEAD ARE PUT INTO ASCENDING (AZ) OR DESCENDING (ZA) ORDER.
DEFAULT: THE DIRECTION REMAINS UNCHANGED.
IF BOTH ARCS AND ENDPOINTS ARE REARRANGED, THEN LEXICO1 FIRST SORTS THE ARCS, AND AFTERWARDS REARRANGES THE DIRECTION OF THE ARCS.

COMPACT
A REPRESENTATION OF THE SORTED GRAPH IS PRODUCED IN WHICH VINFOS, AINFOS AND VERTICES ARE CODED. EACH LIST STARTS ON A NEW LINE.
DEFAULT: A REPRESENTATION OF THE SORTED GRAPH IS PRODUCED IN WHICH EACH SORTED ENTITY OR ARC IS PRINTED ON A NEW LINE. VINFOS AND AINFOS ARE CODED. VERTICES ARE ALSO CODED, UNLESS ARCS OR ENDPOINTS WERE SORTED.
THE LISTS OF ARCS ARE PRESENTED IN THE STRAIGHTFORWARD FORM.

*** 10. LEXICO 2 ***

THE PURPOSE OF LEXICO 2 IS TO SORT THE LISTS OF VERTICES AND ARCS ACCORDING TO THE INFORMATION OF THESE ELEMENTS.
VERTICES AND ARCS CAN BE PUT INTO INCREASING OR DECREASING ORDER OF THEIR INFORMATIONS.

THE KEYWORDS OF LEXICO 2 ARE:

VERTAZ, VERTZA
IN EACH LIST OF VERTICES THE VERTICES ARE PUT INTO INCREASING (AZ) OR DECREASING (ZA) ORDER OF THEIR INFORMATIONS.
DEFAULT: THE ORDER REMAINS UNCHANGED.

ARCSAZ, ARCSZA
IN EACH LIST OF ARCS THE ARCS ARE PUT INTO INCREASING (AZ) OR DECREASING (ZA) ORDER OF THEIR INFORMATIONS.
DEFAULT: THE ORDER REMAINS UNCHANGED.

EPAZ, EPZA
THESE KEYWORDS MAY MODIFY THE DIRECTION OF ARCS.
FOR EACH ARC THE TAIL AND HEAD ARE PUT INTO INCREASING (AZ) OR DECREASING (ZA) ORDER OF THEIR VERTEX-INFORMATIONS.
DEFAULT: THE DIRECTION REMAINS UNCHANGED.

COMPACT
A REPRESENTATION OF THE SORTED GRAPH IS PRODUCED IN WHICH VINFOS, AINFOS AND VERTICES ARE CODED. EACH LIST STARTS ON A NEW LINE.
DEFAULT: A REPRESENTATION OF THE SORTED GRAPH IS PRODUCED IN WHICH EACH SORTED ELEMENT IS PRINTED ON A NEW LINE, WITHOUT CODING.
THE LISTS OF ARCS ARE PRESENTED IN THE STRAIGHTFORWARD FORM.

*** 11. TRANSPOSE 1 ***

THE PURPOSE OF TRANSPOSE1 IS TO CHANGE THE DIRECTION OF ARCS.

THE SEMICOLON CLOSING THE LIST OF KEYWORDS SHOULD BE FOLLOWED BY

```
'GRAPH'  <TITLE OF THE NEW GRAPH>;
ARCLISTS;  <ARC-LIST-IDENTIFIERS>;               (OPTIONAL)
INFLISTS;  <AINFOS-LIST-IDENTIFIERS>;            (OPTIONAL)
AINFOS;    <ARC-INFORMATIONS>;                   (OPTIONAL)
'JOBFIN'
```

IF THE IDENTIFIER OF A LIST OF ARCS IS MENTIONED IN
<ARC-LIST-IDENTIFIERS> THEN THAT LIST IS "ACTIVE".
IF   ARCLISTS; <ARC-LIST-IDENTIFIERS>;
IS MISSING THEN ALL LISTS OF ARCS ARE ACTIVE.
THE REDIRECTIONING OF ARCS IS RESTRICTED TO THE ACTIVE LISTS.

IF THE IDENTIFIER OF A LIST OF ARC-INFORMATIONS OCCURS IN
<AINFOS-LIST-IDENTIFIERS> THEN THE INFORMATIONS IN THAT LIST ARE ACTIVE
THE ARC-INFORMATIONS OCCURRING IN <ARC-INFORMATIONS> ALSO ARE ACTIVE.

```
IF BOTH      INFLISTS;  <AINFOS-LIST-IDENTIFIERS>;
   AND       AINFOS:    <ARC-INFORMATIONS>;
```

ARE MISSING THEN ALL ARCS IN THE ACTIVE LISTS ARE REDIRECTED,
OTHERWISE IT DEPENDS UPON THE INFORMATION AND THE KEYWORDS OF
TRANSPOSE1.

THE KEYWORDS OF TRANSPOSE1 ARE:

```
POSINF, NEGINF
    DEFAULT;   POSINF
    IF POSINF THEN THE DIRECTION OF THE ARCS, IN THE ACTIVE LISTS,
    HAVING AN ACTIVE INFORMATION IS CHANGED.
    IF NEGINF THEN THE DIRECTION OF THE ARCS, IN THE ACTIVE LISTS,
    NOT HAVING AN ACTIVE INFORMATION IS CHANGED.
```

*** 12. SELECT 1 ***

THE PURPOSE OF SELECT1 IS TO GENERATE NEW GRAPHS BY SELECTING ENTITIES FROM A GRAPH.

THE SEMICOLON CLOSING THE LIST OF KEYWORDS IS FOLLOWED BY:

```
'GRAPH' <TITLE OF THE NEW GRAPH>;
'VINFOS'    <LIST IDENTIFIERS>;         (OPTIONAL)
'AINFOS'    <LIST IDENTIFIERS>;         (OPTIONAL)
'VERTICES' <LIST IDENTIFIERS>;         (OPTIONAL)
'ARCS'      <LIST IDENTIFIERS>;         (OPTIONAL)
'JOBFIN'
```

<LIST IDENTIFIERS> IS A LIST OF LIST-IDENTIFIERS OF LISTS OF THE APPROPRIATE TYPE, SUBSEQUENT IDENTIFIERS ARE SEPARATED BY A COMMA. A LIST IS "ACTIVE" IF ITS IDENTIFIER IS MENTIONED IN <LIST IDENTIFIERS>.

THE KEYWORDS OF SELECT1 ARE:

DELETE, RETAIN
  DEFAULT: RETAIN.

ENTITIES, LISTS
  DEFAULT: LISTS.

SO THERE ARE FOUR POSSIBLE COMBINATIONS OF KEYWORDS:

RETAIN, LISTS: THE NEW GRAPH CONSISTS OF THE ACTIVE LISTS.

DELETE, LISTS: THE NEW GRAPH CONSISTS OF THE NON-ACTIVE LISTS.

RETAIN, ENTITIES: THE NEW GRAPH CONSISTS OF THE ENTITIES IN THE ACTIVE LISTS. ALL LIST-IDENTIFIERS ARE RETAINED BUT, IN GENERAL, THE CONTENTS OF THE NON-ACTIVE LISTS WILL BE MODIFIED.

DELETE, ENTITIES: THE ENTITIES IN THE ACTIVE LISTS ARE REMOVED FROM ALL LISTS. ONLY THE IDENTIFIERS OF THE NON-ACTIVE LISTS ARE RETAINED, THE CONTENTS OF THESE LISTS WILL, IN GENERAL, BE MODIFIED.

MULTIPLE ARCS OR EDGES ARE CONSIDERED TO BE DIFFERENT ARCS OR EDGES. THE REMOVAL OF A VERTEX FROM THE GRAPH IMPLIES THE REMOVAL OF ALL ARCS OR EDGES INCIDENT TO THAT VERTEX.

IN THE REPRESENTATION OF THE NEW GRAPH VINFOS, AINFOS AND VERTICES ARE CODED.

*** 13. SELECT 2 ***

THE PURPOSE OF SELECT2 IS TO GENERATE NEW GRAPHS BY SELECTING VERTICES AND ARCS ACCORDING TO THEIR INFORMATION.

THE SEMICOLON CLOSING THE LIST OF KEYWORDS IS FOLLOWED BY:

```
'GRAPH'   <TITLE OF THE NEW GRAPH>;
'VINFOS'   <LIST IDENTIFIERS>;          (OPTIONAL)
'AINFOS'   <LIST IDENTIFIERS>;          (OPTIONAL)
'JOBFIN'
```

<LIST IDENTIFIERS> IS A LIST OF LIST-IDENTIFIERS OF LISTS OF THE APPROPRIATE TYPE. SUBSEQUENT IDENTIFIERS ARE SEPARATED BY A COMMA.
A LIST IS "ACTIVE" IF ITS IDENTIFIER IS MENTIONED IN <LIST IDENTIFIERS>

THE KEYWORDS OF SELECT2 ARE:

POSVERT, NEGVERT
  DEFAULT: POSVERT.
  IF POSVERT THEN THE VERTICES HAVING AN INFORMATION IN AN ACTIVE LIST ARE "ACTIVE", IF NEGVERT THEN THE VERTICES WITHOUT AN ACTIVE INFORMATION ARE CONSIDERED AS "ACTIVE".

DELVERT, RETVERT
  DEFAULT: RETVERT
  IF DELVERT THEN THE ACTIVE VERTICES ARE REMOVED FROM ALL LISTS OF VERTICES.
  IF RETVERT THEN THE NON-ACTIVE VERTICES ARE DELETED AND THE NEW GRAPH CONTAINS ONLY THE ACTIVE VERTICES FROM THE ORIGINAL GRAPH.
  THE REMOVAL OF A VERTEX IMPLIES THE REMOVAL OF ALL ARCS INCIDENT TO THAT VERTEX.

POSARC, NEGARC
  DEFAULT: POSARC
  IF POSARC THEN THE ARCS HAVING AN INFORMATION IN AN ACTIVE LIST ARE "ACTIVE", IF NEGARC THEN THE ARCS WITHOUT AN ACTIVE INFORMATION ARE "ACTIVE".

DELARC, RETARC
  DEFAULT: RETARC.
  IF RETARC THEN THE ACTIVE ARCS ARE RETAINED, IF DELARC THEN THE ACTIVE ARCS ARE REMOVED FROM THE GRAPH.

IN THE NEW GRAPH THE VINFOS, AINFOS AND VERTICES ARE CODED.

*** 14. INDUCE 1 ***

THE PURPOSE OF INDUCE1 IS TO GENERATE NEW GRAPHS BY INDUCTION IN A GRAPH.

THE SEMICOLON CLOSING THE LIST OF KEYWORDS IS FOLLOWED BY:

```
'GRAPH' <TITLE OF THE NEW GRAPH>;
INDUCTEES; <LIST IDENTIFIERS>;
INDUCTORS: <LIST IDENTIFIERS>;
'JOBFIN'
```

<LIST IDENTIFIERS> IS A LIST OF LIST-IDENTIFIERS OF LISTS OF VERTICES, SUBSEQUENT IDENTIFIERS ARE SEPARATED BY A COMMA.

THE NEW GRAPH CONSISTS OF THE OPIGINAL VINFOS, THE LISTS OF VERTICES DEFINED TO BE INDUCTEES AND THE ARCS WHICH ARE GENERATED BY THE INDUCTION PROCESS.

THE VERTICES OCCURRING IN THE LISTS OF VERTICES DEFINED TO BE INDUCTORS ARE INDUCTORS. EACH SUCH VERTEX INDUCES ARCS BETWEEN INDUCTEES IT IS ADJACENT TO.

THE KEYWORDS OF INDUCE1 ARE:

```
TT, HT, TH, HH, TT2, HH2, DEF
  DEFAULT:  DEF.
  ANY COMBINATION OF THESE KEYWORDS MAY BE SPECIFIED, HOWEVER:
  IF DEF THEN THE OTHERS ARE IGNORED,
  IF TT2 THEN TT IS IGNORED,
  IF HH2 THEN HH IS IGNORED.
  EACH OF THESE KEYWORDS DEFINES A SPECIFIC TYPE OF INDUCTION.
  LET I DENOTE AN INDUCTOR, LET T DENOTE THE SET OF INDUCTEES WHICH ARE
  IN-ADJACENT TO I (I.E. AN INDUCTEE IS IN T IFF IT IS THE TAIL OF AN
  ARC HAVING I AS ITS HEAD), LET H DENOTE THE SET OF INDUCTEES WHICH
  ARE OUT-ADJACENT TO I (I.E. AN INDUCTEE IS IN H IFF IT IS THE HEAD
  OF AN ARC HAVING I AS ITS TAIL).
  TT INDUCES AN EDGE FOR EACH PAIR OF VERTICES IN T,
  HT INDUCES AN ARC FROM EACH VERTEX IN H TO EACH IN T,
  TH INDUCES AN ARC FROM EACH VERTEX IN T TO EACH IN H,
  HH INDUCES AN EDGE FOR EACH PAIR OF VERTICES IN H,
  TT2 INDUCES AN ARC FROM EACH VERTEX IN T TO EACH OTHER VERTEX IN T,
  HH2 INDUCES AN ARC FROM EACH VERTEX IN H TO EACH OTHER VERTEX IN H.
  DEF INDUCES AN EDGE FOR EACH PAIR OF VERTICES IN THE UNION OF T
  AND H.
  IT SHOULD BE NOTED THAT THE SETS T AND H ARE NOT NECESSARILY
  DISJOINT.
```

```
ADJ, MULT, WEIGHT, INFO
  DEFAULT:  ADJ.
  LET I DENOTE AN INDUCTOR, T1 AN INDUCTEE WITH W1 ARCS FROM T1 TO I,
  T2 AN INDUCTEE WITH W2 ARCS FROM T2 TO I.
  INDUCTION OF TYPE TT ON I RESULTS IN:
  IF ADJ    THEN ONE   EDGE  BETWEEN T1 AND T2.
  IF MULT   THEN W1*W2 EDGES BETWEEN T1 AND T2.
            IF INFO THEN THIS EDGE(ADJ) OR THESE EDGES (MULT)
            OBTAIN THE VERTEX-INFORMATION OF I AS EDGE-
            INFORMATION, OTHERWISE THE EDGES ARE WITHOUT INFORMATION.
  IF WEIGHT THEN ONE   EDGE  BETWEEN T1 AND T2, WITH W1*W2 AS
                 EDGE-INFORMATION.
  SIMILARLY FOR THE OTHER TYPES OF INDUCTION.
```

LISTS,VERTICES

DEFAULT: VERTICES.

IF VERTICES THEN THE INDUCTION PROCESSES ARE EXECUTED ONLY ONCE FOR EACH INDUCTOR. FOR EACH INDUCTOR THE NEW GRAPH CONTAINS A LIST OF ARCS, IDENTIFIED BY THE LABEL OF THE INDJCTOR, CONTAINING ALL ARCS DUE TO THAT INDUCTOR.

IF LISTS THEN THE INDUCTION PROCESSES ARE EXECUTED FOR EACH LIST OF INDUCTORS AND, WITHIN EACH LIST, FOR EACH INDUCTOR.

CONSEQUENTLY, AN INDUCTOR IS USED M TIMES IF IT OCCURS IN M LISTS OF INDUCTORS. FOR EACH LIST THE NEW GRAPH CONTAINS A LIST OF ARCS, IDENTIFIED BY THE IDENTIFIER OF THE LIST, CONTAINING ALL ARCS DUE TO THAT LIST.

FORMO

BY THIS KEYWORD THE LISTS OF ARCS ARE PRINTED IN THE STRAIGHTFORWARD FORM, EACH ARC IS PRINTED ON A NEW LINE.

IN THE REPRESENTATION OF THE NEW GRAPH THE VINFOS AND VERTICES ARE CODED. THE AINFOS, IF ANY, ARE NOT CODED.

*** 15. INTAR 2 ***

THE PURPOSE OF INTAR2 IS TO GENERATE GRAPHS DEFINED BY A SET OF VERTICES (X) AND THE ARCS HAVING BOTH ENDPOINTS IN X, OR BY TWO DISJOINT SETS OF VERTICES (X AND Y) AND THE ARCS HAVING ONE ENDPOINT IN X AND ONE ENDPOINT IN Y.

THE KEYWORDS OF INTAR2 ARE:

ALLINTRA
  FOR EACH LIST OF VERTICES THE GRAPH DEFINED BY THE VERTICES IN THAT LIST (= THE SET X) IS GENERATED.

ALLINTRACOMPL
  FOR EACH LIST OF VERTICES THE GRAPH DEFINED BY THE COMPLEMENT OF THE VERTICES IN THAT LIST IS GENERATED.

ALLINTER
  FOR EACH PAIR OF DISJOINT LISTS OF VERTICES THE BIPARTITE GRAPH DEFINED BY THE TWO LISTS (SET X AND SET Y) IS GENERATED.

ALLINTERCOMPL
  FOR EACH LIST OF VERTICES THE BIPARTITE GRAPH DEFINED BY THAT LIST (= SET X) AND ITS COMPLEMENT (= SET Y) IS GENERATED.

SPECINTRA
  THE GRAPH DEFINED BY THE VERTICES IN SETX IS GENERATED.

SPECINTRACOMPL
  THE GRAPH DEFINED BY THE COMPLEMENT OF SETX IS GENERATED.

SPECINTER
  THE BIPARTITE GRAPH DEFINED BY SETX AND SETY IS GENERATED, UNLESS THE TWO SETS ARE NOT DISJOINT.

SPECINTERCOMPL
  THE BIPARTITE GRAPH DEFINED BY SETX AND ITS COMPLEMENT IS GENERATED.

IN THE ABOVE SETX AND SETY ARE, IN GENERAL, THE UNION OF SEVERAL LISTS OF VERTICES. TO SPECIFY THESE SETS THE SEMICOLON CLOSING THE LIST OF KEYWORDS IS FOLLOWED BY:

```
'GRAPH' <TITLE>;
SETX = <VERTEX LIST IDENTIFIERS>;
SETY = <VERTEX LIST IDENTIFIERS>;      (FOR SPECINTER ONLY)
'JOBFIN'
```

THE KEYWORD SPECINTER SHOULD NOT OCCUR TOGETHER WITH ANY OF THE KEYWORDS SPECINTRA, SPECINTRACOMPL,SPECINTERCOMPL.

IN THE REPRESENTATION OF THE NEW GRAPH THE VINFOS, AINFOS AND VERTICES ARE CODED.

*** 16. MULARC 1 ***

THE PURPOSE OF MULARC 1 IS TO AGGREGATE MULTIPLE EDGES OR ARCS.
WITHIN EACH LIST OF ARCS EACH SET OF MULTIPLE EDGES OR ARCS IS REPLACED
BY A SINGLE ONE.
THE NEW EDGE OR ARC CAN BE PROVIDED WITH A NUMERICAL INFORMATION, I.E.
THE NUMBER OF EDGES OR ARCS IT REPLACES.

THE SEMICOLON CLOSING THE LIST OF KEYWORDS IS FOLLOWED BY:

'GRAPH' <TITLE OF THE NEW GRAPH>;
'JOBFIN'

THE KEYWORDS OF MULARC1 ARE:

DIRECTED
  THE GRAPH IS CONSIDERED TO BE A DIRECTED ONE.
  DEFAULT: A NON-DIRECTED ONE

MULUP, MULDOWN
  IF ANY OF THESE KEYWORDS IS SPECIFIED THEN THE ARCS OF THE NEW GRAPH
  ARE PROVIDED WITH INFORMATION, I.E. THE MULTIPLICITY OF THAT ARC IN
  THE ORIGINAL LIST OF ARCS.
  WITHIN EACH LIST THE ARCS ARE SORTED IN ASCENDING (MULUP) OR
  DESCENDING (MULDOWN) ORDER OF MULTIPLICITY.
  DEFAULT: THE ARCS ARE NOT PROVIDED WITH INFORMATION.

COMPACT
  IN THE REPRESENTATION OF THE NEW GRAPH THE VINFOS AND VERTICES ARE
  CODED, EACH LIST STARTS ON A NEW LINE.
  DEFAULT: THE VINFOS ARE CODED, THE VERTICES ARE NOT CODED,
           EACH ARC IS PRINTED ON A NEW LINE.

THE ORIGINAL ARC-INFORMATIONS, IF ANY, ARE DELETED FROM THE GRAPH.

*** 17. NEWINFO 1 ***

THE PURPOSE OF NEWINFO 1 IS TO CREATE NEW LISTS OF VINFOS AND AINFOS. ONLY NUMERICAL INFORMATIONS CAN BE PROCESSED.

A NUMERICAL INFORMATION IS A SEQUENCE OF NUMBERS, EACH NUMBER IS A "COMPONENT" OF THE INFORMATION.
FOR EACH PART OF A NEW LIST, A SET OF PERMITTED VALUES CAN BE DEFINED FOR EACH COMPONENT. ALL INFORMATIONS SUCH THAT THE VALUE OF EACH COMPONENT IS A PERMITTED ONE ARE INCLUDED IN THE NEW LIST.
THE PERMITTED VALUES ARE DEFINED IN THE FOLLOWING WAY:

```
        (A,(1,2,3),A,A,(-1,5,6),A)
```

IF A COMPONENT OF A SPECIFICATION OF PERMITTED VALUES CONSISTS OF THE LETTER A THEN THERE ARE NO RESTRICTIONS ON THE VALUE OF THE CORRESPONDING COMPONENT OF THE INFORMATIONS.
OTHERWISE THE COMPONENT OF A SPECIFICATION CONSISTS OF A LIST OF NUMBERS, ENCLOSED WITHIN PARENTHESES, WHICH IS THE SET OF PERMITTED VALUES FOR THE CORRESPONDING COMPONENT OF THE INFORMATIONS.

IF THE LENGTH OF A SPECIFICATION, I.E. ITS NUMBER OF COMPONENTS, EQUALS THE LENGTH OF AN INFORMATION THEN THE COMPARISON IS STRAIGHTFORWARD.
OTHERWISE THE JUSTIFICATION OF THE INFORMATIONS DEFINES THE CORRESPONDENCE BETWEEN THE COMPONENTS OF THE SPECIFICATION AND THE COMPONENTS OF THE INFORMATION.
IF THE INFORMATIONS ARE LEFT-JUSTIFIED THEN THE COMPARISON STARTS AT THE LEFT HAND SIDE OF THE SPECIFICATION AND THE INFORMATION.
IF THE INFORMATIONS ARE RIGHT-JUSTIFIED THEN THE COMPARISON STARTS AT THE RIGHT HAND SIDE OF THE SPECIFICATION AND THE INFORMATION.
IF THE NUMBER OF COMPONENTS OF THE INFORMATION EXCEEDS THAT OF THE SPECIFICATION, THE REDUNDANT COMPONENTS OF THE INFORMATION ARE CONSIDERED AS PERMITTED ONES.
THE LENGTH OF A SPECIFICATION SHOULD NOT EXCEED THE LENGTH OF THE LONGEST INFORMATION OF ITS TYPE.
THE LENGTH OF A NEWLISTID SHOULD NOT EXCEED THE LENGTH OF THE LONGEST LIST-IDENTIFIER OF ITS TYPE.

FOR A SINGLE NEW LIST SEVERAL SPECIFICATIONS OF PERMITTED VALUES CAN BE COMBINED:

```
        (A,(2),A), ((3),A,A), (A,A,(4,5))
```

THE NEW LIST CONTAINS ALL INFORMATIONS SATISFYING AT LEAST ONE OF THE SPECIFICATIONS.

THE SEMICOLON CLOSING THE LIST OF KEYWORDS SHOULD BE FOLLOWED BY:

```
'VINFOS' NEWLISTID = <SPECIFICATIONS OF PERMITTED VALUES>;
.
.
.
'VINFOS' NEWLISTID = <SPECIFICATIONS OF PERMITTED VALUES>;
'AINFOS' NEWLISTID = <SPECIFICATIONS OF PERMITTED VALUES>;
.
.
.
'AINFOS' NEWLISTID = <SPECIFICATIONS OF PERMITTED VALUES>;
'JOBFIN'
```

THE KEYWORDS OF NEWINFO 1 ARE:

LENGTHAINF, LENGTHVINF
IF LENGTH (AINF FOR AINFOS, VINF FOR VINFOS) THEN THE COMPARISON OF COMPONENTS IS PRECEDED BY A COMPARISON OF THE LENGTHS OF THE SPECIFICATION AND THE INFORMATION. IF THE LENGTHS ARE NOT EQUAL THEN THE INFORMATION DOES NOT SATISFY THE SPECIFICATION, OTHERWISE THE NORMAL COMPARISON OF THE COMPONENTS IS PERFORMED.

SPECONT =
DEFAULT: SPECONT = 500 ;
THE VALUE OF SPECONT IS AN UPPER BOUND FOR THE NUMBER OF PERMITTED VALUES IN ANY SPECIFICATION.

EPS =
DEFAULT: EPS = "-7 ;
TWO NUMERICAL VALUES ARE CONSIDERED EQUAL IF THE ABSOLUTE VALUE OF THEIR DIFFERENCE DOES NOT EXCEED EPS.

IN THE NEW GRAPH THE VINFOS, AINFOS AND VERTICES ARE CODED.

*** 18. NEWINFO 2 ***

THE PURPOSE OF NEWINFO2 IS TO CREATE NEW VINFOS AND AINFOS. ONLY NUMERICAL INFORMATIONS CAN BE PROCESSED.

A NUMERICAL INFORMATION IS A SEQUENCE OF NUMBERS, EACH NUMBER IS A "COMPONENT" OF THE INFORMATION.
NEWINFO2 CAN DELETE COMPONENTS FROM THE INFORMATIONS AND CAN RE-ARRANGE THE ORDER OF THE REMAINING COMPONENTS, THIS IS SPECIFIED IN THE FOLLOWING WAY:

```
        (0,3,1,0,2).
```

DUE TO THIS SPECIFICATION EACH INFO OF LENGTH 5, I.E. CONSISTING OF 5 COMPONENTS, IS REPLACED BY AN INFO OF LENGTH 3. THE NEWINFO CONTAINS, IN THIS ORDER, THE THIRD, FIFTH AND SECOND COMPONENT OF THE ORIGINAL INFO.
THE COMPONENTS ARE COUNTED FROM THE LEFT TO THE RIGHT.

THE SEMICOLON CLOSING THE LIST OF KEYWORDS SHOULD BE FOLLOWED BY:

```
'VINFOS'     <SPECIFICATIONS> ;     (OPTIONAL)
'AINFOS'     <SPECIFICATIONS> ;     (OPTIONAL)
'JOBFIN'
```

TWO SUBSEQUENT SPECIFICATIONS ARE SEPARATED BY A COMMA.

THE KEYWORD OF NEWINFO2 IS

CODE
  IN THE NEW GRAPH THE VINFOS, AINFOS AND VERTICES ARE CODED.
  DEFAULT: THE VINFOS, AINFOS AND VERTICES ARE NOT CODED.
    MOREOVER, EACH NEWINFO IS PRINTED ON A NEW LINE, IF THERE ARE NEW VINFOS EACH VERTEX IS PRINTED ON A NEW LINE, IF THERE ARE NEW AINFOS EACH ARC IS PRINTED ON A NEW LINE.

IMPORTANT: NEWINFO2 CAN ACCEPT ONLY ONE JOB FOR EACH GRAPH.

19. ELEMENTARY ANALYSES

Only six procedures for analysing graphs are available now, some of these however have several keywords corresponding to different types of analysis. Experience has shown that elementary analyses, if applied to suitably chosen subgraphs and induced graphs, is a powerful tool to obtain insight into the phenomenon that is studied.

Several procedures will be added, e.g. detection and analysis of cliques, blocks and clusters, and covering algorithms.

*** 20, SURVEYS 1 ***

THE PURPOSE OF SURVEYS1 IS TO PRODUCE FREQUENCY TABLES DESCRIBING THE MULTIPLICITY AND THE USE OR INCIDENCE OF VINFOS, AINFOS, VERTICES AND ARCS.

THE KEYWORDS OF SURVEYS1 ARE:

LISTSIZES
  FOR EACH TYPE OF LISTS THE DISTRIBUTION OF THE SIZES OF THE LISTS IS PRINTED. THE LIST-IDENTIFIERS ARE ALSO PRINTED IN ORDER OF SIZE OF THE LIST.

MULVIN
  A FREQUENCY TABLE OF THE MULTIPLICITY OF VINFOS IS PRINTED. A VINFO HAS MULTIPLICITY M IF IT OCCURS IN M LISTS OF VERTEX-INFORMATIONS,

ALLVINFOS
  A FREQUENCY TABLE OF THE USE OF VINFOS IS PRINTED.
  A VINFO IS USED N TIMES IF N VERTICES HAVE THAT INFORMATION,

VINFOLISTS
  FOR EACH LIST OF VINFOS A FREQUENCY TABLE OF THE USE OF THE VINFOS IN THAT LIST IS PRINTED.

MULAIN
  A FREQUENCY TABLE OF THE MULTIPLICITY OF AINFOS IS PRINTED.

ALLAINFOS
  A FREQUENCY TABLE OF THE USE OF AINFOS IS PRINTED.

AINFOLISTS
  FOR EACH LIST OF AINFOS A FREQUENCY TABLE OF THE USE OF THE AINFOS IN THAT LIST IS PRINTED.

MULVERT
  A FREQUENCY TABLE OF THE MULTIPLICITY OF VERTICES IS PRINTED,

ALLVERTICES
  A FREQUENCY TABLE OF THE INCIDENCE OF THE VERTICES IS PRINTED,
  A VERTEX HAS INCIDENCE I IF THAT VERTEX IS INCIDENT TO I ARCS,

VERTEXLISTS
  FOR EACH LIST OF VERTICES A FREQUENCY TABLE OF THE INCIDENCE OF THE VERTICES IN THAT LIST IS PRINTED.

ALLARCS
  A FREQUENCY TABLE OF THE MULTIPLICITY OF THE ARCS IS PRINTED, MOREOVER, THE NUMBER OF LOOPS AND NON-LOOPS AND THE COEFFICIENT OF ADJACENCY OF THE GRAPH IS PRINTED,

ARCLISTS
  FOR EACH LIST OF ARCS A FREQUENCY TABLE OF THE MULTIPLICITY OF THESE ARCS, THE NUMBER OF LOOPS AND NON-LOOPS AND THE COEFFICIENT OF ADJACENCY IS PRINTED,

*** 21. WEAKCOMPS 1 ***

THE PURPOSE OF WEAKCOMPS1 IS TO DETERMINE THE WEAKLY CONNECTED COMPONENTS OF A NON-DIRECTED GRAPH (OR A SERIES OF ITS PARTIAL GRAPHS) AND TO PERFORM A FIRST ANALYSIS ON THESE COMPONENTS.
THE ANALYSIS CONSISTS OF THE DETERMINATION OF THE DISTANCES BETWEEN THE VERTICES OF A COMPONENT AND THE IDENTIFICATION OF ITS CENTRAL AND PERIPHERAL VERTICES WITH RESPECT TO SEVERAL MEASURES OF CENTRALITY.

EACH COMPONENT IS IDENTIFIED BY THE IDENTIFIER OF ITS REPRESENTATIVE.
THE REPRESENTATIVE OF A COMPONENT IS THE VERTEX IN THE COMPONENT HAVING THE MINIMUM LABEL.

THE KEYWORDS OF WEAKCOMPS1 ARE:

SIZES
A FREQUENCY TABLE OF THE SIZES OF THE COMPONENTS IS PRINTED.
THE IDENTIFIERS AND SIZES OF THE COMPONENTS ARE PRINTED, BOTH IN ORDER OF THE IDENTIFIERS AND IN ORDER OF THE SIZES (ISOLATED VERTICES NOT INCLUDED).
THE COEFFICIENT OF CONNECTIVITY OF THE GRAPH IS PRINTED.

CROSSREFS
FOR EACH VERTEX THE IDENTIFIER OF ITS COMPONENT IS PRINTED.
FOR EACH COMPONENT ITS IDENTIFIER AND THE IDENTIFIER OF ITS VERTICES ARE PRINTED.

MINSIZE = , MAXSIZE =
DEFAULT: MINSIZE = 2, MAXSIZE = +INFINITY.
IF MINSIZE < 2 IS SPECIFIED THEN WEAKCOMPS REDEFINES MINSIZE = 2.
ONLY THE COMPONENTS SATISFYING MINSIZE <= SIZE <= MAXSIZE ARE ANALYSED.

NBHD =
DEFAULT: NBHD = 1.
THE NEIGHBOURHOOD OF A VERTEX CONSISTS OF THE VERTICES AT DISTANCE <= NBHD FROM THAT VERTEX.
FOR EACH VERTEX IN THE COMPONENT THE NUMBER OF VERTICES IN ITS NEIGHBOURHOOD IS PRINTED.
IF NBHD > 1 THEN THE VERTICES ARE SORTED IN DECREASING ORDER OF THE SIZE OF THEIR NEIGHBOURHOOD.
IF NBHD > 1 THEN, FOR EACH VERTEX, THE NUMBERS OF VERTICES AT DISTANCE <= I (I=1,...,NBHD) ARE PRINTED, BOTH IN ORDER OF THE VERTICES AND IN LEXICOGRAPHICALLY DECREASING ORDER OF THE NUMBERS OF VERTICES.

MINDIST = , MAXDIST =
DEFAULT: MINDIST = 1, MAXDIST = +INFINITY.
THE NUMBER OF VERTICES AT A DISTANCE SATISFYING MINDIST <= DISTANCE <= MAXDIST IS THE NUMBER OF CONTACTS OF A VERTEX.
FOR EACH VERTEX ITS NUMBER OF CONTACTS IS PRINTED, BOTH IN ORDER OF THE VERTICES AND IN DECREASING ORDER OF THE NUMBER OF CONTACTS.
IN THE DEFAULT CASE THE SECOND LIST IS NOT PRINTED.

RUSHADJ,RUSHMUL
FOR EACH VERTEX ITS RUSH IS PRINTED, BOTH IN ORDER OF THE VERTICES AND IN DECREASING ORDER OF THE RUSH.
THE RUSH IS COMPUTED IN THE FOLLOWING WAY:
EACH VERTEX SENDS ONE UNIT OF FLOW TO EACH OF ITS CONTACTS.
THE UNIT OF FLOW IS EQUALLY DISTRIBUTED OVER THE PATHS OF MINIMAL

LENGTH CONNECTING THE TWO VERTICES. THE RUSH OF A VERTEX IS THE FLOW THROUGH THE VERTEX, EXPRESSED AS A FRACTION OF ALL FLOW IN THE COMPONENT.
KEYWORD RUSHMUL TAKES THE MULTIPLICITY OF EDGES INTO ACCOUNT,
KEYWORD RUSHADJ IGNORES THE MULTIPLICITY OF EDGES.
A JOB MAY CONTAIN BOTH KEYWORDS.

ADJ, MEDIAN, SUM, MEAN, EXCEN, BAVELAS

A JOB MAY CONTAIN ANY COMBINATION OF THESE KEYWORDS.
EVEN IF NONE OF THESE KEYWORDS IS SPECIFIED, THE CORRESPONDING MEASURES OF CENTRALITY ARE PRINTED FOR EACH VERTEX.
IF A KEYWORD IS SPECIFIED THEN THE VERTICES ARE ALSO PRINTED IN THE DECREASING ORDER OF CENTRALITY.
ADJ CORRESPONDS TO THE ADJACENCY OF A VERTEX, I.E. THE NUMBER OF VERTICES AT DISTANCE = 1,
MEDIAN CORRESPONDS TO THE MEDIAN OF THE DISTANCES TO THE OTHER VERTICES IN THE COMPONENT.
SUM CORRESPONDS TO THE SUM OF THE DISTANCES TO THE OTHER VERTICES,
MEAN CORRESPONDS TO THE MEAN OF THE DISTANCES TO THE OTHER VERTICES,
EXCEN CORRESPONDS TO THE EXCENTRICITY OF A VERTEX, I.E. THE MAXIMUM OF THE DISTANCES TO THE OTHER VERTICES,
BAVELAS CORRESPONDS TO THE RATIO BETWEEN THE MEAN OF THE DISTANCES TO THE OTHER VERTICES AND THE MEAN OF ALL DISTANCES IN THE COMPONENT (THIS IS NOT THE SAME COEFFICIENT AS ORIGINALLY PROPOSED BY BAVELAS BUT YIELDS THE SAME ORDER OF THE VERTICES).
IT SHOULD BE NOTED THAT THE THREE MEASURES SUM, MEAN AND BAVELAS ALL YIELD THE SAME ORDER OF THE VERTICES.

ALL, PARTIAL

DEFAULT: ALL
IF ALL THEN ONLY THE COMPONENTS OF THE GRAPH ITSELF ARE ANALYSED.
IF PARTIAL THEN THE COMPONENTS OF A SERIES OF PARTIAL GRAPHS ARE ANALYSED.
IN THIS CASE THE SEMICOLON CLOSING THE LIST OF KEYWORDS SHOULD BE FOLLOWED BY:

```
'ARCS'      <ARCLIST-IDENTIFIERS> ;
.   .
.   .
.   .
'ARCS'      <ARCLIST-IDENTIFIERS> ;
'JOBFIN'
```

TWO SUBSEQUENT ARCLIST-IDENTIFIERS ARE SEPARATED BY A COMMA.
E.G.

```
'ARCS' A1,A2;
'ARCS' A3;
'JOBFIN'
```

IN THIS EXAMPLE TWO PARTIAL GRAPHS ARE ANALYSED.
THE FIRST ONE IS DEFINED BY THE ARCS IN LIST A1 AND A2,
THE SECOND ONE BY THE THREE LISTS OF ARCS A1, A2 AND A3.

FOR EACH COMPONENT THAT IS ANALYSED, WEAKCOMPS1 PRINTS A FREQUENCY TABLE OF THE DISTANCES IN THE COMPONENT.

WEAKCOMPS1 ALSO PRINTS A SURVEY, CONTAINING, FOR EACH COMPONENT THAT IS ANALYSED:

THE IDENTIFIER OF ITS REPRESENTATIVE,
THE SIZE OF THE COMPONENT,
THE NUMBER OF EDGES,
THE NUMBER OF LOOPS,
THE NUMBER OF REDUNDANT NON-LOOPS,
THE COEFFICIENT OF ADJACENCY,
THE RADIUS,
THE DIAMETER,
THE MEDIAN AND MEAN OF ALL DISTANCES,
THE TOTAL NUMBER OF CONTACTS.

*** 22, STRONGCOMPS 1 ***

THE PURPOSE OF STRONGCOMPS1 IS TO DETERMINE THE STRONGLY CONNECTED COMPONENTS OF A DIRECTED GRAPH (OR OF A SERIES OF ITS PARTIAL GRAPHS), AND TO PERFORM A FIRST ANALYSIS ON THESE COMPONENTS,

THE KEYWORDS OF STRONGCOMPS1 ARE

INDIST, OUTDIST
    DEFAULT :   OUTDIST

SIZES, CROSSREFS, MINSIZE, MAXSIZE
    AS DESCRIBED FOR WEAKCOMPS1

NBHD, MINDIST, MAXDIST, RUSHADJ, RUSHMUL, ADJ, MEDIAN, SUM, MEAN, BAVELAS
    AS DESCRIBED FOR WEAKCOMPS1, HOWEVER, THE COEFFICIENTS ARE BASED UPON THE OUT-DISTANCES (IF OUTDIST) OR THE IN-DISTANCES (IF INDIST) OF EACH VERTEX, I,E, UPON THE ROWS OR COLUMNS OF THE DISTANCE-MATRIX,
    THUS, IF INDIST AND NBHD = 1 , THEN THE NEIGHBOURHOOD OF VERTEX V CONSISTS OF ALL VERTICES X WHICH ARE IN-ADJACENT TO V; IF OUTDIST AND NBHD = 1 , THEN THE NEIGHBOURHOOD OF V CONSISTS OF ALL VERTICES WHICH ARE OUT-ADJACENT TO V,

ALL, PARTIAL
    AS DESCRIBED FOR WEAKCOMPS1

*** 23. RUSH 1 ***

THE PURPOSE OF RUSH1 IS TO COMPUTE THE RUSH IN A SET (M) OF VERTICES WHICH IS THE RESULT OF A FLOW FROM A SET (F) OF VERTICES TO A SET (T) OF VERTICES.

THE SEMICOLON CLOSING THE LIST OF KEYWORDS SHOULD BE FOLLOWED BY

```
FROM :      <VERTEX-LIST-IDENTIFIERS>;
TO :        <VERTEX-LIST-IDENTIFIERS>;
MEASURE :   <VERTEX-LIST-IDENTIFIERS>;
'JOBFIN'
```

<VERTEX-LIST-IDENTIFIERS> IS A LIST OF LIST-IDENTIFIERS OF LISTS OF VERTICES, SUBSEQUENT IDENTIFIERS ARE SEPARATED BY A COMMA.
A LIST OF VERTICES IS 'ACTIVE' IF ITS IDENTIFIER OCCURS IN ANY OF THE <VERTEX-LIST-IDENTIFIERS>.

THE SET OF VERTICES F CONSISTS OF THE VERTICES IN THE LISTS
FROM :   <VERTEX-LIST-IDENTIFIERS>.
THE SET OF VERTICES T CONSISTS OF THE VERTICES IN THE LISTS
TO :   <VERTEX-LIST-IDENTIFIERS>.
THE SET OF VERTICES M CONSISTS OF THE VERTICES IN THE LISTS
MEASURE :   <VERTEX-LIST-IDENTIFIERS>.

EACH VERTEX V IN F SENDS ONE UNIT OF FLOW TO EACH VERTEX W IN T THAT CAN BE REACHED FROM V, (I.E. MINDIST <= DISTANCE(V,W) <= MAXDIST). THE FLOW FROM V TO W IS EQUALLY DISTRIBUTED OVER ALL PATHS OF MINIMAL LENGTH FROM V TO W. THE RUSH IN VERTEX X IS THE TOTAL AMOUNT OF FLOW PASSING THROUGH X.

THE KEYWORDS OF RUSH1 ARE :

DIRECTED
THE GRAPH IS CONSIDERED TO BY A DIRECTED ONE.
DEFAULT: A NON-DIRECTED ONE

MINDIST, MAXDIST
DEFAULT : MINDIST = 1 , MAXDIST = + INFINITY

RUSHADJ, RUSHMUL
A JOB MAY CONTAIN BOTH KEYWORDS, IF NONE IS SPECIFIED THEN RUSHADJ IS ASSUMED.
RUSHADJ IGNORES THE MULTIPLICITY OF EDGES OR ARCS,
RUSHMUL TAKES THE MULTIPLICITY INTO ACCOUNT.

ALL, LISTS
A JOB MAY CONTAIN BOTH KEYWORDS, IF NONE IS SPECIFIED THEN ALL IS ASSUMED.
IF ALL THEN A LIST IS PRINTED CONTAINING A LINE FOR EACH VERTEX OF AN ACTIVE LIST.
IF LISTS THEN FOR EACH ACTIVE LIST OF VERTICES A LIST IS PRINTED CONTAINING A LINE FOR EACH VERTEX OF THE VERTEXLIST.
THE LINE OF A VERTEX CONTAINS :
ITS IDENTIFIER,
THE FLOW IT SENDS (IF THE VERTEX BELONGS TO F),
THE FLOW IT RECEIVES (IF THE VERTEX BELONGS TO T),
ITS RUSHADJ AND/OR ITS RUSHMUL (IF THE VERTEX BELONGS TO M).

*** 24. DISRUPT 1 ***

THE PURPOSE OF DISRUPT1 IS TO REDUCE A GRAPH TO THE NULL-GRAPH,
I.E. TO THE GRAPH WITHOUT ELEMENTS.
IN EACH STEP OF THE REDUCTION PROCESS A SET OF VERTICES (S) IS SELECTED AND DELETED FROM THE GRAPH. TO DETERMINE THIS SET A CRITERION IS COMPUTED FOR EACH VERTEX, VERTICES WITH MINIMUM (OR MAXIMUM) VALUE OF THE CRITERION CONSTITUTE THE SET M. THE VALUE OF THE CRITERION AND A FREQUENCY TABLE CAN BE PRINTED FOR ALL VERTICES AND FOR EACH LIST OF VERTICES. THE SET S EITHER IS M OR IS THE INTERSECTION OF M WITH THE UNION OF A NUMBER OF SPECIFIED LISTS OF VERTICES.
IN EACH STEP THE VERTICES IN S ARE PRINTED.

THE KEYWORDS OF DISRUPT1 ARE:

INCIDENCE, LOOPS, DEGREE, VALENCY, ADJACENCY, ININCI, OUTINCI,
INDEG, OUTDEG, INADJ, OUTADJ, RECI

DEFAULT: INCIDENCE
ONLY ONE OF THESE KEYWORDS MAY OCCUR IN THE LIST OF KEYWORDS, IT DETERMINES THE CRITERION TO BE USED. LET V DENOTE A VERTEX OF THE GRAPH, THEN

CRITERION	VALUE
INCIDENCE	THE NUMBER OF ARCS INCIDENT WITH V,
LOOPS	THE NUMBER OF LOOPS ON V,
DEGREE	THE NUMBER OF NON-LOOPS INCIDENT WITH V,
VALENCY	DEGREE + 2 * LOOPS,
ADJACENCY	THE NUMBER OF VERTICES ADJACENT WITH V,
ININCI	THE NUMBER OF ARCS IN-INCIDENT WITH V, I.E. THE NUMBER OF ARCS HAVING V AS HEAD,
OUTINCI	THE NUMBER OF ARCS OUT-INCIDENT WITH V,
INDEG	THE IN-DEGREE OF V, I.E. THE NUMBER OF NON-LOOPS IN-INCIDENT WITH V,
OUTDEG	THE OUT-DEGREE OF V,
INADJ	THE IN-ADJACENCY OF V, I.E. THE NUMBER OF VERTICES IN-ADJACENT WITH V, I.E. THE NUMBER OF VERTICES X (X¬=V) SUCH THAT (X,V) IS AN ARC,
OUTADJ	THE OUT-ADJACENCY OF V,
RECI	THE RECIPROCITY OF V, I.E. THE NUMBER OF VERTICES BOTH IN- AND OUT-ADJACENT TO V.

MIN, MAX

DEFAULT: MIN
THE MINIMUM (OR MAXIMUM) VALUE OF THE CRITERION DETERMINES THE SET S.

ITER, ORIG

DEFAULT: ITER
IF ITER THEN IN EACH STEP THE CRITERION IS COMPUTED, THUS IS BASED UPON THE REMAINING SUB-GRAPH.
IF ORIG THEN IN EACH STEP THE ORIGINAL VALUES OF THE CRITERION ARE USED.

LCRIT

IN EACH STEP, FOR EACH LIST OF VERTICES, THE CRITERION IS PRINTED FOR EACH VERTEX IN THAT LIST.

LFREQ

IN EACH STEP, FOR EACH LIST OF VERTICES, A FREQUENCY TABLE OF THE CRITERION OF THE VERTICES IN THAT LIST IS PRINTED.

VCRIT

IN EACH STEP THE CRITERION OF EACH VERTEX IS PRINTED.

VFREQ

IN EACH STEP A FREQUENCY TABLE OF THE CRITERION OF ALL VERTICES IS PRINTED.

ALLVERT, SPECVERT

DEFAULT: ALLVERT
IF ALLVERT THEN, IN EACH STEP THE SET S IS THE SET M.
IF SPECVERT THEN SPECIAL VERTICES ARE DISTINGUISHED AND S IS THE INTERSECTION OF M AND THE SPECIAL VERTICES.
IF SPECVERT THEN THE LIST OF KEYWORDS IS FOLLOWED BY:

```
VERTICES:  <VERTEX-LIST-IDENTIFIERS>;
'JOBFIN'
```

<VERTEX-LIST-IDENTIFIERS> IS A LIST OF IDENTIFIERS OF LISTS OF VERTICES, SUBSEQUENT IDENTIFIERS SEPARATED BY A COMMA. THE VERTICES IN THESE LISTS ARE THE SPECIAL VERTICES.

*** 25. INTAR 1 ***

THE PURPOSE OF INTAR1 IS TO ANALYSE RELATIONS WITHIN SETS (INTRA-ANALYSES) AND RELATIONS BETWEEN SETS (INTER-ANALYSES). THE KEYWORDS DEFINE THE SETS TO BE ANALYSED.

THE KEYWORDS OF INTAR1 ARE:

ALLINTRA
FOR EACH LIST OF VERTICES THE GRAPH DEFINED BY THE VERTICES IN THAT LIST (= THE SET X) IS ANALYSED.

ALLINTRACOMPL
FOR EACH LIST OF VERTICES THE GRAPH DEFINED BY THE COMPLEMENT OF THE VERTICES IN THAT LIST IS ANALYSED.

ALLINTER
FOR EACH PAIR OF DISJOINT LISTS OF VERTICES THE BIPARTITE GRAPH DEFINED BY THE TWO LISTS (SET X AND SET Y) IS ANALYSED.

ALLINTERCOMPL
FOR EACH LIST OF VERTICES THE BIPARTITE GRAPH DEFINED BY THAT LIST (= SET X) AND ITS COMPLEMENT (= SET Y) IS ANALYSED.

SPECINTRA
THE GRAPH DEFINED BY THE VERTICES IN SETX IS ANALYSED.

SPECINTRACOMPL
THE GRAPH DEFINED BY THE COMPLEMENT OF THE VERTICES IN SETX IS ANALYSED.

SPECINTER
THE BIPARTITE GRAPH DEFINED BY SETX AND SETY IS ANALYSED, UNLESS THE TWO SETS ARE NOT DISJOINT.

SPECINTERCOMPL
THE BIPARTITE GRAPH DEFINED BY SETX AND ITS COMPLEMENT IS ANALYSED.

IN THE ABOVE, SETX AND SETY ARE, IN GENERAL, THE UNION OF SEVERAL LISTS OF VERTICES. TO SPECIFY THESE SETS THE SEMICOLON CLOSING THE LIST OF KEYWORDS IS FOLLOWED BY:

```
'GRAPH' <TITLE>;
SETX = <VERTEX LIST IDENTIFIERS>;
SETY = <VERTEX LIST IDENTIFIERS>;     (FOR SPECINTER ONLY)
'JOBFIN'
```

THE KEYWORD SPECINTER SHOULD NOT OCCUR TOGETHER WITH ANY OF THE KEYWORDS SPECINTRA, SPECINTRACOMPL, SPECINTERCOMPL.

```
THE OUTPUT OF AN INTRA-ANALYSIS CONSISTS OF:
VERT    THE NUMBER OF VERTICES IN X,
ARCS    THE NUMBER OF NON-LOOPS HAVING BOTH ENDPOINTS IN X,
LOOPS   THE NUMBER OF LOOPS IN X,
EXTA    THE NUMBER OF ARCS HAVING ONE ENDPOINT IN X AND ONE ENDPOINT IN
        THE COMPLEMENT OF X,
EXTV    THE NUMBER OF VERTICES NOT IN X, ADJACENT TO AT LEAST ONE VERTEX
        IN X,
COMP    THE NUMBER OF COMPONENTS CONSTITUTING THE GRAPH X, INCLUDING
ISO     THE NUMBER OF ISOLATED VERTICES OF THE GRAPH X,
INCI    THE FRACTION OF ARCS HAVING AT LEAST ON ENDPOINT IN X,
ADJ     THE COEFFICIENT OF ADJACENCY OF THE GRAPH X,
CONN    THE COEFFICIENT OF CONNECTIVITY OF THE GRAPH X.

THE OUTPUT OF AN INTER-ANALYSIS CONSISTS OF:
VERTX   THE NUMBER OF VERTICES IN X,
VERTY   THE NUMBER OF VERTICES IN Y,
ARCS    THE NUMBER OF ARCS HAVING ONE ENDPOINT IN X AND ONE ENDPOINT IN
        Y,
LPS+    THE NUMBER OF ARCS HAVING BOTH ENDPOINTS IN X PLUS
        THE NUMBER OF ARCS HAVING BOTH ENDPOINTS IN Y,
EXTA    THE NUMBER OF ARCS HAVING ONE ENDPOINT IN THE UNION OF X AND Y
        AND ONE ENDPOINT OUTSIDE THAT UNION,
EXTV    THE NUMBER OF VERTICES NOT IN THE UNION OF X AND Y, ADJACENT TO
        AT LEAST ONE VERTEX IN THAT UNION,
COMP    THE NUMBER OF COMPONENTS CONSTITUTING THE BIPARTITE GRAPH,
        INCLUDING
ISOX    THE NUMBER OF ISOLATED VERTICES IN X AND
ISOY    THE NUMBER OF ISOLATED VERTICES IN Y,
INCI    THE FRACTION OF ARCS HAVING AT LEAST ONE ENDPOINT IN THE
        UNION OF X AND Y,
ADJXY   THE COEFFICIENT OF ADJACENCY OF THE BIPARTITE GRAPH,
CONNX   THE BIPARTITE COEFFICIENT OF CONNECTIVITY WITHIN X,
CONNY   THE BIPARTITE COEFFICIENT OF CONNECTIVITY WITHIN Y,
CONXY   THE BIPARTITE COEFFICIENT OF CONNECTIVITY BETWEEN X AND Y,
D2X     THE FRACTION OF PAIRS OF VERTICES IN X HAVING DISTANCE 2,
D2Y     THE FRACTION OF PAIRS OF VERTICES IN Y HAVING DISTANCE 2.
```

*** REFERENCES ***

1. JAC.M. ANTHONISSE
A GRAPH-DEFINING LANGUAGE
REPORT BW 30/73
MATHEMATISCH CENTRUM, AMSTERDAM