CURRICULUM ANALYSIS USING A GRAPHICAL TECHNIQUE

A Thesis

Presented to

the Faculty of the Department of

Chemical Engineering

The University of Houston

In Partial Fulfillment

of the Requirements for the Degree of

Master of Science in Chemical Engineering

---

by

Pablo Adolfo Longoria Treviño

December, 1975

ACKNOWLEDGMENT

I would like to express my gratitude to "CONACYT" (Consejo Nacional de Ciencia y Tecnología) for the economic support that made it possible for me to pursue graduate studies, also I wish to thank my advisor Dr. Ernest J. Henley for the constant guidance and encouragement he has provided over the past several months. A debt of gratitude is due to my parents, and last but not least I would like to acknowledge the love, understanding, encouragement and patience of my wife Guadalupe.

CURRICULUM ANALYSIS USING A GRAPHICAL TECHNIQUE

A Thesis

Presented to

the Faculty of the Department of

Chemical Engineering

The University of Houston

In Partial Fulfillment

of the Requirements for the Degree of

Master of Science in Chemical Engineering

---

by

Pablo Adolfo Longoria Treviño

December, 1975

ABSTRACT

This thesis develops an algorithm suitable for curriculum analysis. The program is based on a path finding algorithm developed by Caceres and Henley. The program is simple to use and requires only a module diagram for the system. Such a diagram is similar to a block diagram. It is easy to draw for a person specialized in the topic being analyzed, even if he is not familiar with flow graph analysis. The diagram shows direct prerequisite relations in a curriculum of a specified field, the elements (modules) of the diagram include an identification flag to differentiate between the "AND" and "OR" modules. The final output is a fault tree, with the concept to be studied as the top event.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

A block diagram is a graphic representation of a system showing all its elements and how they are interconnected. The diagram is composed of blocks and nodes and connecting directed arrows.

A fault tree is a model that graphically and logically represents the various combinations of possible events occuring in a system that lead to a "top" event. The fault tree is structured so that the main event appears as the top event in the tree, and the sequence of events that lead to it are shown below the top event and are logically linked to the top event by standard "OR" and "AND" logic gates.

The relation between the block diagrams and fault trees has been pointed out before (2,3), and a computer program for the construction of a fault tree from a block diagram has been developed (3). In the course of this research a computer program suitable for curriculum analysis was developed.

The program is based in the Caceres-Henley path finding algorithm but modified to include labeling of the modules to identify the corresponding logical gates. The program finds all the prerequisite modules for any given module of the system.

The input to the program is a list of modules with the corresponding type of gate as well as the input and output nodes specified. This information is obtained from a module diagram showing direct prerequisite relations for a curriculum of a specified field.

In chapter 2 theoretical background related to block diagram description, subsets and different types of sets is summarized, then chapter 3 includes a brief description of the Caceres-Henley path finding algorithm.

In chapter 4 the module diagram analysis is outlined and then applied to analyze an undergraduate chemical engineering course on Stoichiometry.

CHAPTER 2

BLOCK DIAGRAM CHARACTERIZATION

Block Diagram Characterization

A block diagram is a graphic representation of a system composed of blocks interconnected by directed arrows, figure 2.1 shows a typical block diagram.

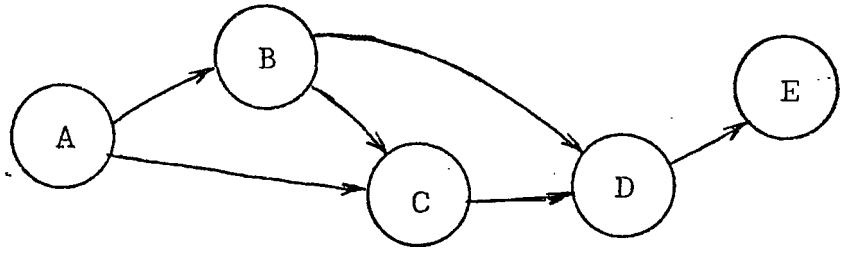


Figure 2.1

In order to represent the above system by means other than a graph some modifications must be made, for instance the introduction of nodes in the scheme so that each element in the system may be described by specifying the input node, the module name, and the output node. Thus the system in figure 2.1 may be represented as shown in figure 2.2

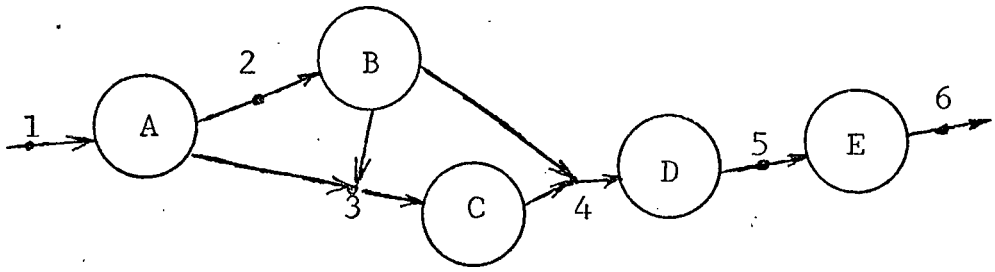


Figure 2.2

This diagram may now be represented by the following notation:

(2.1) System { (1,A,2),(1,A,3),(2,B,3),(2,B,4),
(3,C,4),(4,D,5),(5,E,6) }

In the above notation each element of the set represents a step in the diagram, figure 2.3 shows the part of a diagram equivalent to a "step".

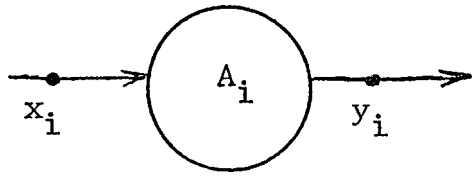


Figure 2.3

For any step "i", "A" represents the module, "x" the input node. and "y" the output node. It is easy to see that this type of notation is useful in characterizing systems for computer analysis.

Subsets

The following subsets will be used in subsequent sections therefore a brief description is required.

Class subsets

A class subset or simply class "C_i" is defined as the set of all the elements in a system that have node "i" as the output node, thus when all the elements of a diagram are classified according to their output node, there will be a class composed of one or more elements having the same output node associated with each node in the diagram, except for the initiating nodes (those that do not have predecessor or preceding nodes), since they do not appear as the output node of any elements. The following example will clarify the definition of a class. Consider the system given in figure 2.4. This system may also be represented as shown in equation (2.2)

(2.2) System: { (1,A,3),(2,B,3),(3,C,4),(4,D,5),(4,D,6),
(5,E,7),(6,F,8),(7,G,8),(8,H,9) }

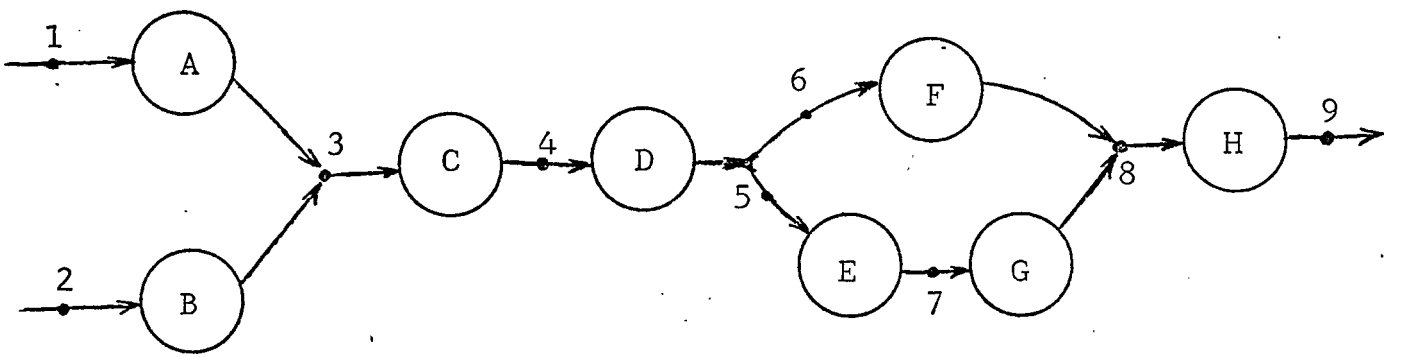


Figure 2.4

The classes associated with each node for the system of equation (2.2) are:

(2.3) $C_1 = \{ \emptyset \}$

(2.4) $C_2 = \{ \emptyset \}$

(2.5) $C_3 = \{ (1,A,3),(2,B,3) \}$

(2.6) $C_4 = \{ (3,C,4) \}$

(2.7) $C_5 = \{ (4,D,5) \}$

(2.8) $C_6 = \{ (4,D,6) \}$

(2.9) $C_7 = \{ (5,E,7) \}$

(2.10) $C_8 = \{ (6,F,8),(7,G,8) \}$

(2.11) $C_9 = \{ (8,H,9) \}$

Series sets:

A series set is defined as a set whose elements have a series configuration in which the output node of an element is equal to the input node of the next element in the set. Refering to figure 2.4 and corresponding equation (2.2), a series set for that system would be:

(2.12) Series set: $\{ (1,A,3),(3,C,4),(4,D,6),(6,F,8),$
$(8,H,9) \}$

Path sets:

A path set "P" is defined as a series set formed by all the elements leading from an initial input node to

the final output node of the system. For the example system given by equation (2.2), two path sets would be:

(2.13) $P_1 = \{(1,A,3),(3,C,4),(4,D,6),(6,F,8),(8,H9)\}$

(2.14) $P_2 = \{(2,B,3),(3,C,4),(4,D,5),(5,E,7),$
$(7,G,8),(8,H,9)\}$

There is another way to represent any path and that is by writing the path as a sequence of nodes and modules contained in it, thus the path given by equation (2.14) may be written as:

(2.15) $P_2 = [\ 2\text{-}B\text{-}3\text{-}C\text{-}4\text{-}D\text{-}5\text{-}E\text{-}7\text{-}G\text{-}8\text{-}H\text{-}9\ ]$

Furthermore, the path sets are classified into two types: (1) minimal path sets, and (2) non-minimal path sets.

A minimal path set is defined as a path set containing a minimum number of elements, that is, each element in the minimal path set belongs to a different class. On the other hand a non-minimal path set contains the elements required to go from an initial node to the final node but its elements do not necessarily belong to a different class, that is, a node may be used or traversed more than once within a path.

Consider the system of figure 2.5, for this system we may write one minimal path and one non-minimal path to illustrate the concepts.

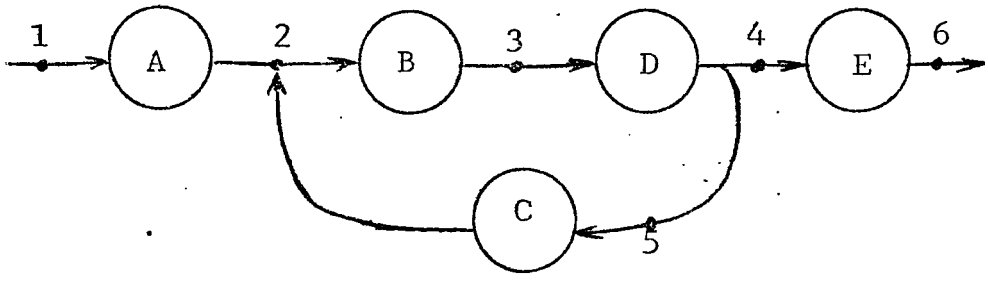


Figure 2.5

(2.16) $P_1 = \{(1,A,2),(2,B,3),(3,D,4),(4,E,6)\}$

(2.17) $P_2 = \{(1,A,2),(2,B,3),(3,D,5),(5,C,2),$
$(2,B,3),(3,D,4),(4,E,6)\}$

P_1 ia a minimal path while P_2 is a non-minimal path since nodes 2,3, and 4 are used more than once.

CHAPTER 3

MINIMAL PATH FINDING ALGORITHM

In this chapter an algorithm for finding all the minimal paths leading to the final output node from the basic or initiating nodes developed by Caceres and Henley (3) is outlined.

Description of algorithm

Starting in reverse order from output to input the first element in a path set will be an element of a class associated with the final node of the system. By definition a path set is a series set, therefore the next element in the minimal path will have as an output node the input node of the first element, thus the second element may be any of the elements belonging to the class associated with the input node of the first element. Following this reasoning subsequent, elements of the minimal path are added to the set until a basic (initiating) node is reached.

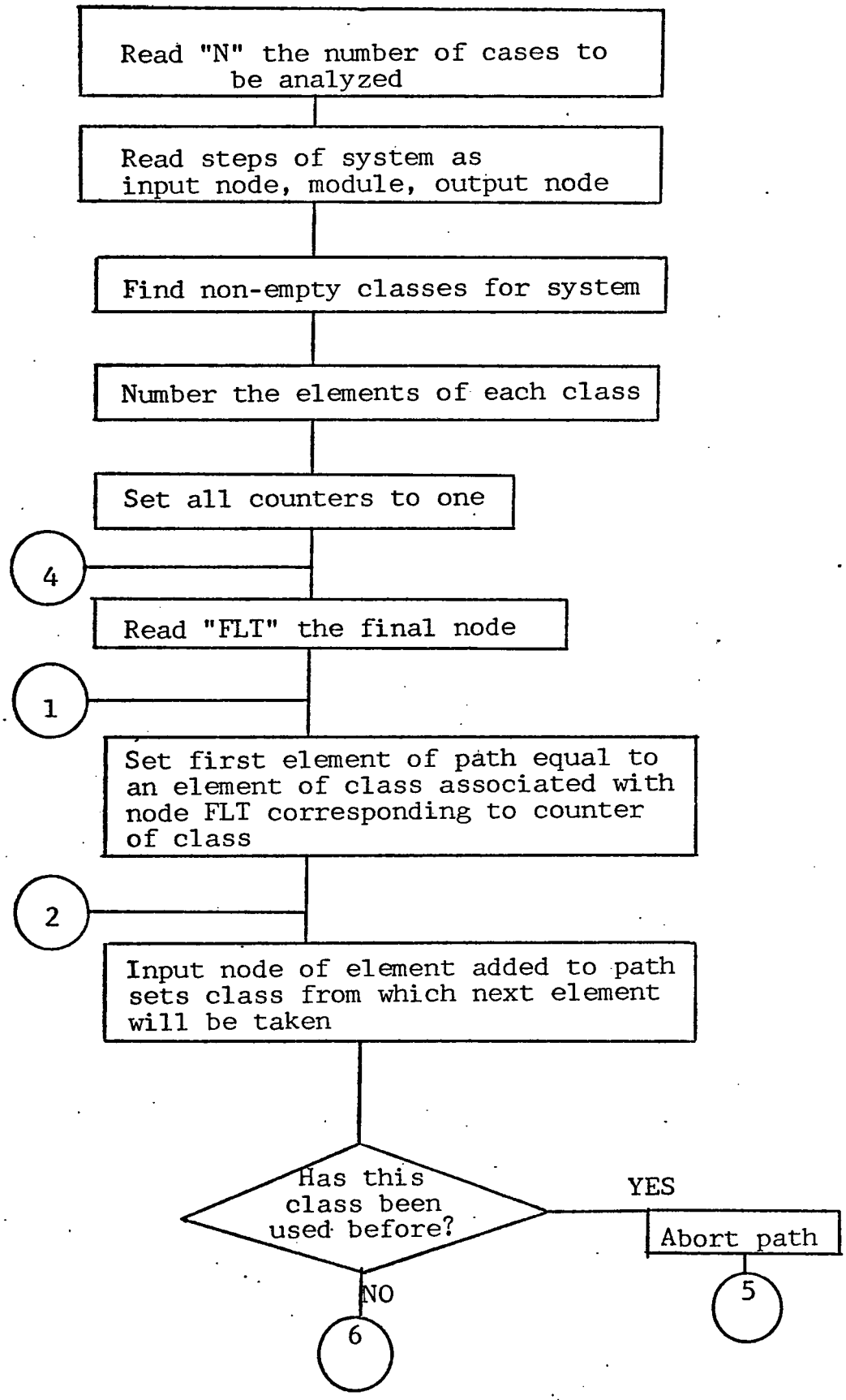
In order to obtain all the different paths a counter associated with each class is used to indicate which of the elements in any class should be added to the path in consideration. Initially all counters are set to 1 meaning that the first element of the class will be added whenever that class is encountered.

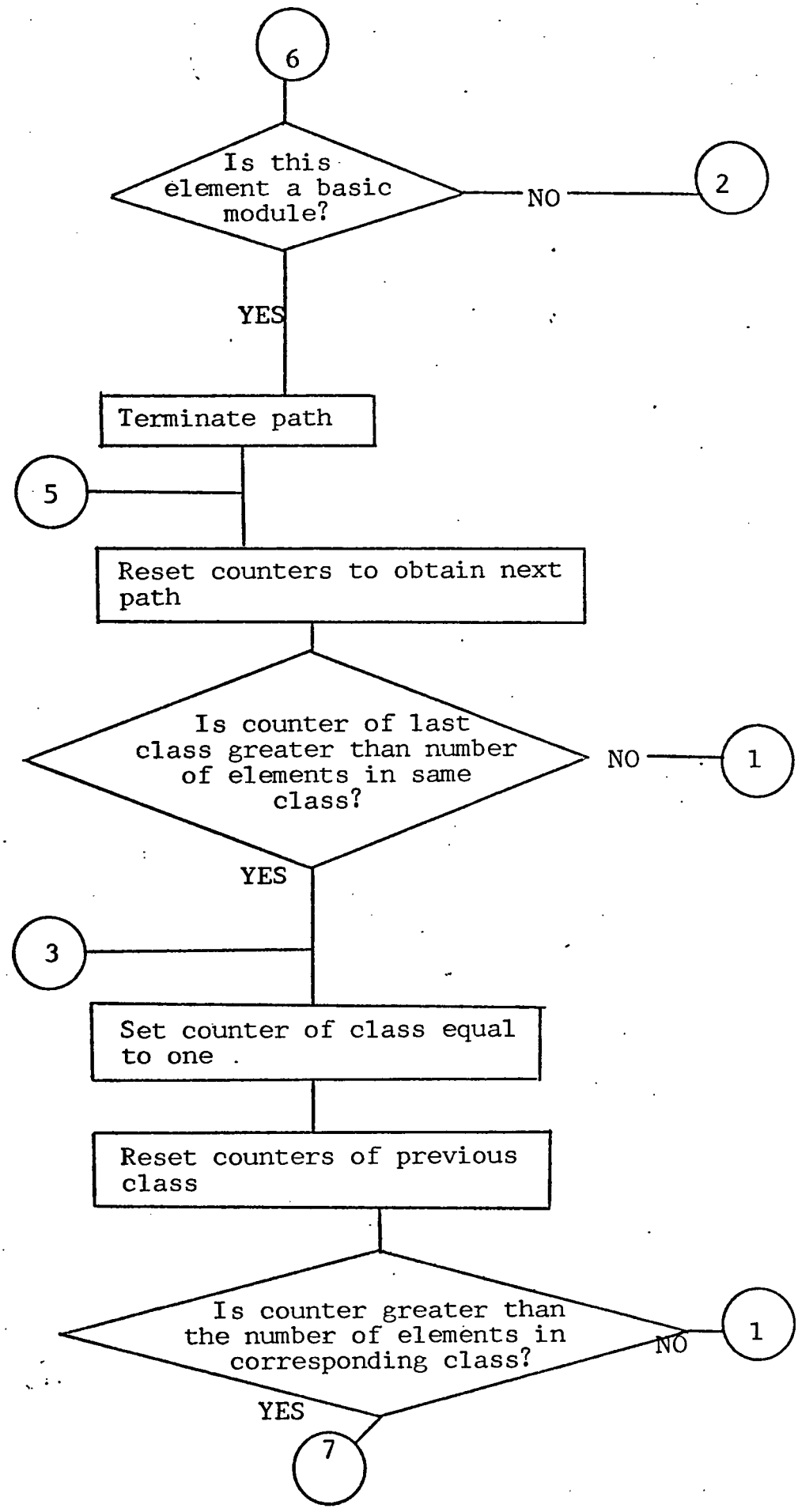
After a path is completed the counter associated with the class of the last element in the path is increased by one. If the counter does not exceed the number of elements in that class, the path finding procedure is repeated and new paths are obtained, on the other hand, if the counter becomes greater than the number of elements in its class, the counter is set equal to one and the counter adjustment procedure is repeated for the previous classes encountered in the path until one of the counters can be increased. The procedure is terminated when, after a cycle, all the counters are again equal to one.

Furthermore since this algorithm is searching for minimal paths, each element in the path must belong to a different class otherwise the path would not be minimal. Therefore, a path is terminated when the input node of the last element that has been added corresponds to a class previously encounteres in the path. When a path is terminated the counters are adjusted as indicated before, and, after deleting the non-minimal path the procedure is repeated to find a new path.

This procedure is illustrated by the computer flow diagram in next page.

Flow chart for algorithm:




6
Is this element a basic module?
NO
2
YES
Terminate path
5
Reset counters to obtain next path
Is counter of last class greater than number of elements in same class?
NO
1
YES
3
Set counter of class equal to one
Reset counters of previous class
Is counter greater than the number of elements in corresponding class?
NO
1
YES
7

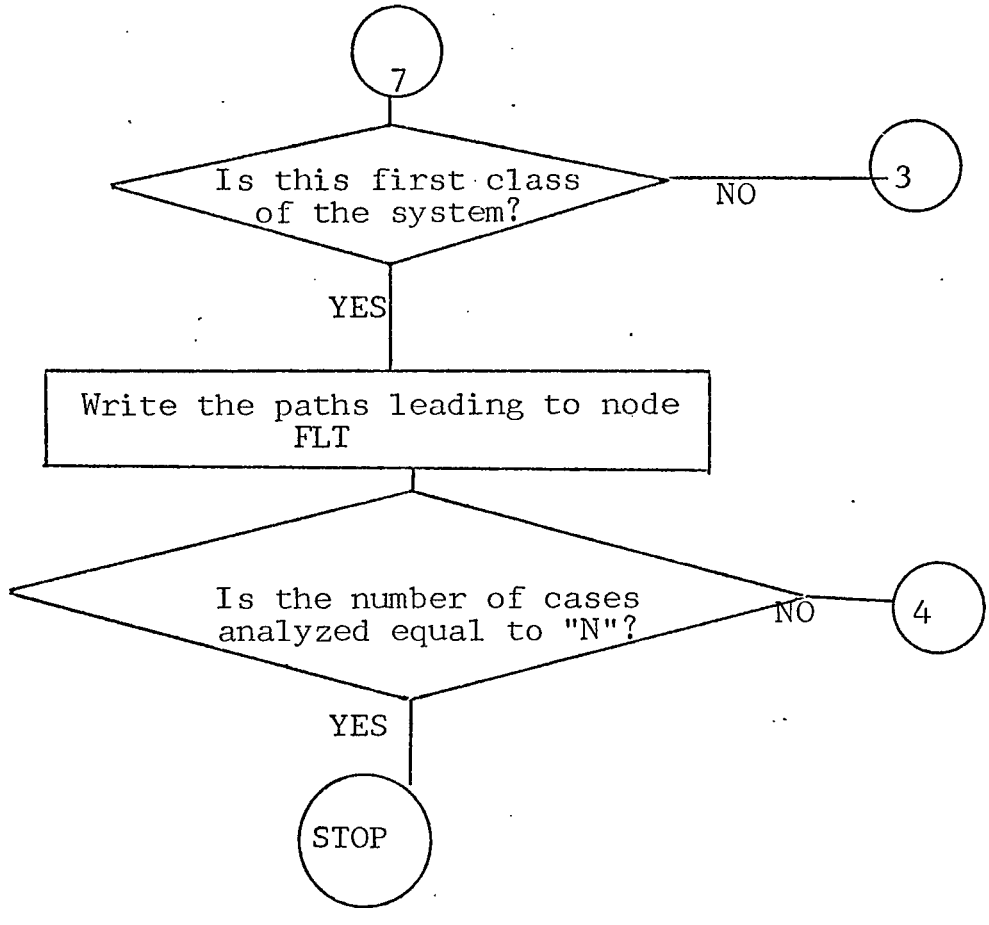


Figure 3.1

Caceres developed a program based on the algorithm of figure 3.1. This program finds all the minimal paths leading to a given final node in a system. Later in chapter 4 this algorithm is used as the base to develop a program suitable for curriculum analysis.

CHAPTER 4

CURRICULUM ANALYSIS BY MODULE DIAGRAM TECHNIQUE

Description of the module diagram:

This diagram was developed from the block diagram in order to use it for curriculum analusis. The diagram consists of modules, nodes, flag gates, and directed arrows. It is very easy to draw for a person who is specialized in the field being analyzed, since it shows only direct prerequisite relations between the modules of the system.

Elements of the module diagram:

Modules:

A module is the representation of the subjects that form the curriculum for a specific field. Each module is composed of a set of concepts closely related which may be taught as a unit. Units and dimensions, say, would be one module in a diagram for a Stoichiometry course, this module could include the following.

Module 1.- Units and dimensions

1.- Unit quantities

2.- Dimensions

3.- Equivalents

4.- Conversion of units

5.- Dimensionless variables

If a more detailed analysis is desired this module could be broken down into submodules and each element of the module would be decomposed into subelements and the same kind of approach could be used to analyze the module as a system to determine the relation between the concepts and the prerequisites needed for each of them, as well as, a logical sequence to teach them. A graphical representation of a hypothetical module "5" is given in figure 4.1

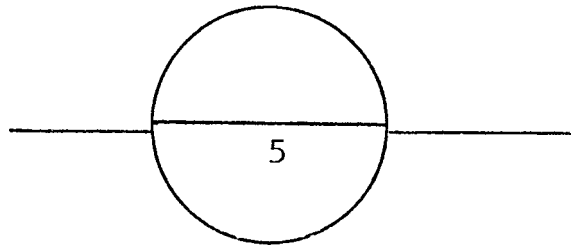


Figure 4.1

Flag gates:

Flag gates are used in the module diagram to identify whether the module is an "OR" or an "AND" module, by this is meant whether all prerequisite modules represent alternative paths so that any one of them is enough to satisfy the requirements, and the others are equivalent modules, or if all prerequisite modules are needed in order to satisfy the prerequisites.

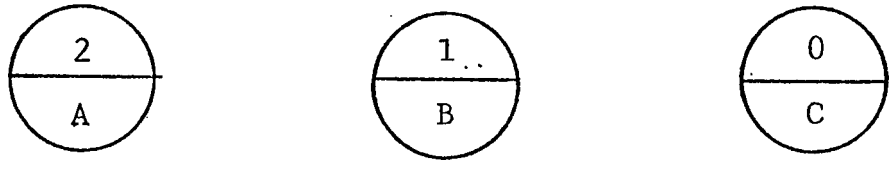
There are three types of flag gates namely: (1) Basic module gates, (2) "OR" module gates, and (3) "AND"

module gates. The basic module gates are used to identify the initiating or basic modules, that is, those modules that do not have any predecessors. This flag gate is represented in the module diagram by a "2".

The "OR" module flag gate identifify the OR modules, that is, those modules that can be reached by two or more different paths, or in other words those modules whose predecessors are equivalent modules so that the module in consideration may be approached through either one of its predecessors. This type of flag gate is represented in the module diagram by a "1".

Finally the "AND" flag gate identify the AND modules, that is, those modules which as prerequisite include all of the direct predecessors modules. This type of flag gate is represented by a "0" in the module diagram.

Figure 4.2 shows the graphical representation of all three types of flag gates:



a) Basic module A. b) "OR" module B. c) "AND" module C.

Figure 4.2

Nodes:

The nodes are used in the module diagram as in the case of block diagrams. The nodes, with respect to each module, are of two types: (1) Input nodes and (2) Output nodes. Nodes are represented by dots "." in the module diagram.

Directed arrows:

The directed arrows are used to indicate the direction of flow in the diagram.

A typical module diagram is shown in figure 4.3, the hypothetical system represented here consists of one OR module, two AND modules and two basic modules.

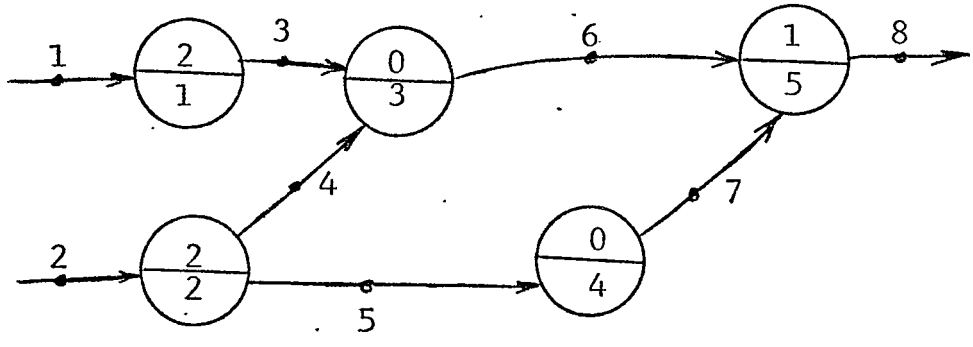


Figure 4.3

At this point it is worth pointing out that since the path finding algorithm uses only the output nodes of each module to form the classes, the diagram may be represented in more than one way as long as all the steps are included. Figure 4.4 shows another representation of the system of figure 4.3. Each one of these representations has its own advantages and disadvantages and the one to use depends on the size and complexity of the system as well as on the personal preference of the person writing the diagram.

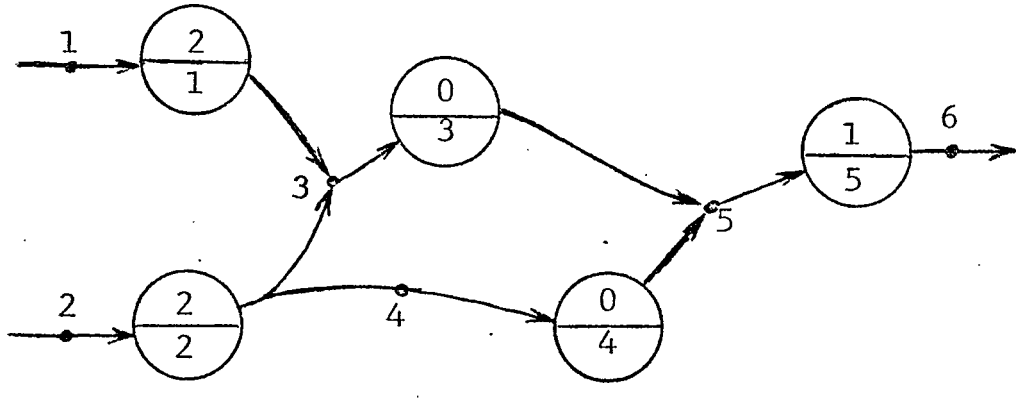


Figure 4.4

Restrictions of the diagram:

There is no way to represent a two way module, that is, a combination of AND and OR modules in the present model, therefore the situation in which the predecesor modules are of both types (AND and OR), must be avoided and one way to do it is by using dummy modules. Figure 4.5 shows a hypothetical case in which this problem occurs and how it could be avoided.

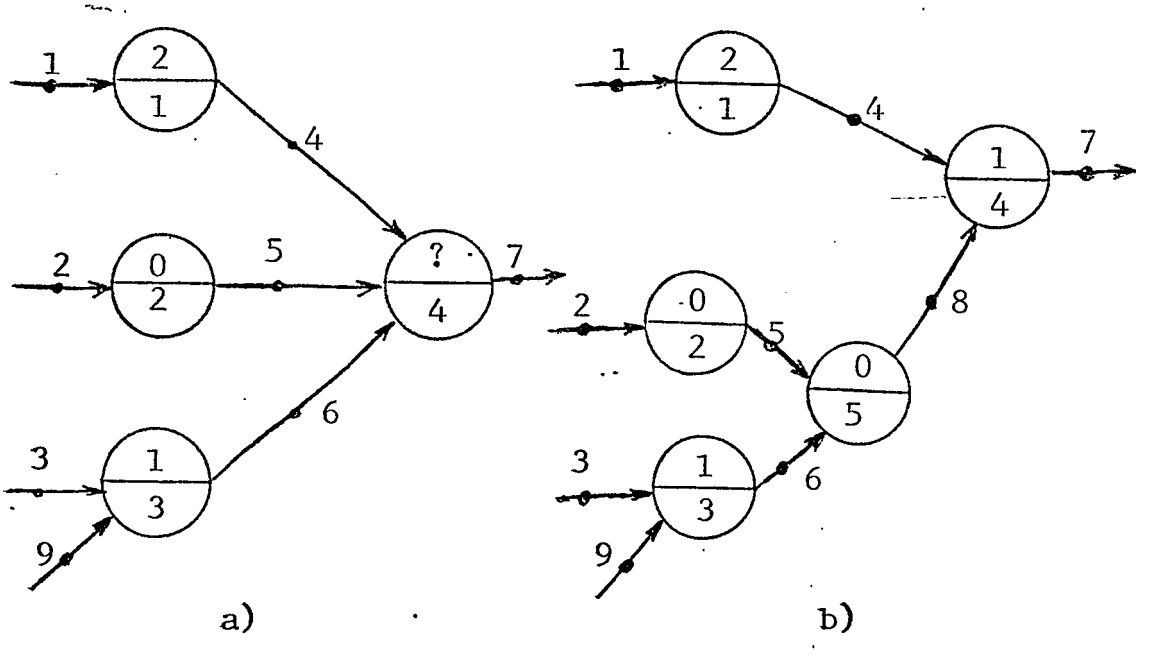


Figure 4.5

In diagram a) of figure 4.5 a hybrid AND/OR module is required to represent the true system, but since that type of module does not exist in the present model for module diagrams, by introducing a dummy module one is able to overcome the problem.

CHAPTER 5

CONSTRUCTION OF THE PREREQUISITE DIAGRAM

A prerequisite diagram is a graphical representation showing all the prerequisite modules for a given module in the system. This diagram shows the relationship between the different modules in a form similar to that of a fault tree (1); the module being analyzed corresponding to the top event of the tree. The flag gate functions as a logical gate of the fault tree, and each prerequisite module would correspond to an element of the tree.

The procedure for constructing the prerequisite diagram can be easily visualized in terms of an example. The hypothetical system shown in figure 4.3 will be used here to determine the prerequisite diagram for module 5.

First using the path finding algorithm of Caceres described in chapter 3, the minimal paths for the system are found. They are:

(5.1) $P_1 = \{(1,1,3),(3,3,6),(6,5,8)\}$

(5.2) $P_2 = \{(2,2,4),(4,3,6),(6,5,8)\}$

(5.3) $P_3 = \{(2,2,5),(5,4,7),(7,5,8)\}$

We can write the path matrix for the above equations

as follows: First, rewrite the paths to eliminate the connecting nodes;

(5.4) $P_1 = \{ 1\text{-}1\text{-}3\text{-}3\text{-}6\text{-}5\text{-}8 \}$

(5.5) $P_2 = \{ 2\text{-}2\text{-}4\text{-}3\text{-}6\text{-}5\text{-}8 \}$

(5.6) $P_3 = \{ 2\text{-}2\text{-}5\text{-}4\text{-}7\text{-}5\text{-}8 \}$

Second, invert the order of the elements in the paths so that the first element will be the output node corresponding to the module being analyzed;

(5.7) $P_1 = \{ 8\text{-}5\text{-}6\text{-}3\text{-}3\text{-}1\text{-}1 \}$

(5.8) $P_2 = \{8\text{-}5\text{-}6\text{-}3\text{-}4\text{-}2\text{-}2\}$

(5.9) $P_3 = \{ 8\text{-}5\text{-}7\text{-}4\text{-}5\text{-}2\text{-}2 \}$

Finally equations (5.7),(5.8), and (5.9) are used as the columns of the path matrix. Figure 5.1 shows the resulting path matrix.

8	8	8
5	5	5
6	6	7
3	3	4
3	4	5
1	2	2
1	2	2

Figure 5.1 Path matrix.

In the path matrix of figure 5.1, odd rows represent nodes and even rows modules of the system.

All the information required to construct the prerequisite diagram is contained in the path matrix. In order to be able to draw this diagram a series of modifications to the original path matrix are required.

First modification:

The first element of all the columns except the first are eliminated, then the modules on all the other columns are checked against the modules of the first column to determine the portions of each path that appear more than once in consecutive columns. MOD1 is the subroutine used to preform this modification. A listing of MOD1 is shown in the appendix 3. Figure 5.2 shows the putput of this subroutine.

8	0	0
5	0	0
6	0	0
3	0	4
3	0	5
1	2	2
1	2	2

Figure 5.2 Path matrix after first modification.

Second Modification:

In this modification the nodes that identify the beginning of the branches of the prerequisite diagram are determined and marked with a negative sign. Appendix 3 shows a listing of MOD2, the subroutine used for this modification. Figure 5.3 shows the path matrix after the second modification.

-8	0	0
5	0	0
-6	0	0
3	0	4
-3	0	5
1	2	2
1	2	2

Figure 5.3 Path matrix after second modification.

Third modification;

This modification determines to which module in a previous column, the top module of each column is equivalent to. This information is saved in two new rows at the end of each column. The first entry is the module corresponding to the equivalent of the top module of the column, and the second entry is the column in which the

module is located.

A listing of MOD3, the subroutine used for this modification is given in appendix 3. Figure 5.4 shows the path matrix after the third modification.

-8	0	0
5	0	0
-6	0	0
3	0	4
-3	0	5
1	2	2
1	2	2
0	1	3
0	1	1

Figure 5.4 Path matrix after third modification.

Fourth modification:

With this modification the unnecessary nodes in the path matrix are eliminated to reduce it to contain only key nodes and the modules of the system.

The nodes of the matrix resulting after the third modification are checked, if they are zero or negative numbers they are left unchanged, but if they are positive numbers the corresponding modules that have those nodes as input nodes are saved and the nodes are eliminated. The modules saved are stored at the top of each column starting

on the second row.

The listing of MOD4, the subroutine used for this modification is given in appendix 3 and figure 5.5 shows the form of the path matrix after the fourth modification.

-8	0	0
5	2	4
-6	0	2
3	0	0
-3	0	0
1	0	0
0	0	0

Figure 5.5 Path matrix after fourth modification.

Fifth modification:

The modules saved with the previous modification are now moved to the proper place using the information stored at the end of each column by subroutine MOD3. The subroutine used to preform this modification is MOD5. A listing of MOD5 is given in appendix 3. Figure 5.6 shows the path matrix after the fifth modification.

-8	0	0
5	0	0
-6	0	0
3	0	4
-3	0	2
1	2	0
0	0	0

Figure 5.6 Path matrix after fifth modification.

Sixth modification:

In this modification the elements of the path matrix representing a segment of line in a branch of the prerequisite diagram are identified as -99. The listing of the subroutine to do this, MOD6, is given in the appendix 3. Figure 5.7 shows the final form of the path matrix.

-8	0	0
5	0	0
-6	-99	-99
3	0	4
-3	-99	2
1	2	0
0	0	0

Figure 5.7 Path matrix after sixth modification.

Figure 5.7 shows the path matrix in its final form, the next step is to classify each element in the matrix so that the prerequisite diagram may be drawn, this is done in the ANALY subroutine. The listing of this subroutine is shown in appendix 3. In this subroutine each element of the path matrix is checked to determine the kind of module it represents, from this information a new matrix is formed which is the prerequisite diagram.

There are four different types of elements in the final form of the path matrix, they are:

(1)	$P(J,I) = -99$	Represents a segment of line connecting the modules in the diagram.
(2)	$-99 < P(J,I) < 0$	Represents the begining of a connecting line in the prerequisite diagram.
(3)	$P(J,I) = 0$	Represents an empty element.
(4)	$P(J,I) > 0$	Represents a module in the prerequisite diagram.

Furthermore each element $P(J,I)$ of types (1),(2), and (4) is subclassified depending on its location in the matrix, as follows;

Type (1) $P(J,I) = -99$

1a.- $P(J,I) = -99$ and $P(J,I+1) = 0$

1b.- $P(J,I) = -99$ and $P(J+1,I) = -99$

1c.- $P(J,I) = -99$ and $P(J,I+1) \neq 0$ and $P(J+1,I) \neq -99$

Type (2) $-99 < P(J,I) < 0$

2a.- $P(J,I) < 0$ and $P(J,I) \neq -99$ and $J = 1$, $I = 1$

2b.- $-99 < P(J,I) < 0$ except $P(1,1)$

Type (4) $P(J,I) > 0$

4a.- $P(J,I) > 0$ and $P(J,I+1) \neq 0$

4b.- $P(J,I) > 0$ and $P(J,I+1) = 0$

Figure 5.8 shows the representation in the prerequisite diagram of each type of element.

Type of element	Representation
1a	-------
1b	--┬----- ┆
1c	--- ┬ ┆
2a	Blank space
2b	┆ ┆ ┆ ├---- ┆
3	Blank space
4a	** * * ** ┆ AND ┆ or ---- ** * * ** ┆ OR ┆
4b	** * * **

Figure 5.8

Figure 5.9 shows the prerequisite diagram for module "5" of the system in figure 4.3

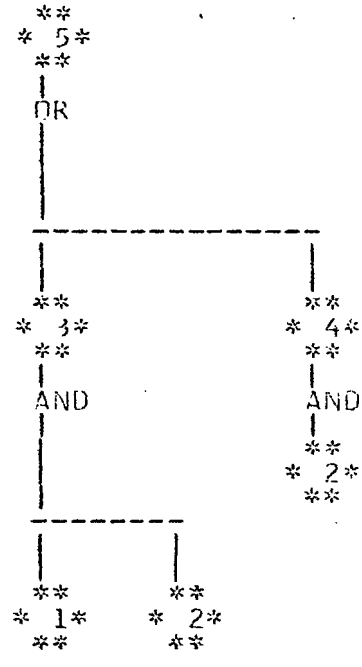


Figure 5.9

The way to interpret the diagram of figure 5.9 is as follows: When one wants to learn module 5 since this is an OR module, one can approach it by using either module 3 or module 4. If one chooses to take module 4, one needs to learn module 2 and since module 2, is a basic module this completes the path, thus one way to get to module 5 is by: 2-and-4-or-5 or simply:

$$P_1 = 2\text{-}4\text{-}5$$

On the other hand if one selects to take the module 3 one needs to learn first both modules 1 and 2 because module 3 ia and AND module, therefore another way to get to module 5 is by: $\frac{1}{2}$>and-3-or-5 or simply:

$$P_2 = \frac{1}{2} >\text{-}3\text{-}5$$

CHAPTER 6

ANALYSIS OF A CHEMICAL ENGINEERING CALCULATIONS COURSE USING MODULE DIAGRAM TECHNIQUE

In this chapter the concepts outlined in the previous chapters are used to analyze an undergraduate chemical engineering course on chemical engineering calculations. A module diagram for this system was developed following the outline of Himmelblau's book (10).

A list of the modules included in the system is given in table 6.1. A brief description of the concepts involved in each module is shown in table 6.2.

The module diagram used for this system is given in figure 6.1. This diagram contains all the information one needs to analyze any module in the system, there is no need to draw a different diagram to analyze each module.

To ilustrate the procedure five key modules were selected and used as top events of the system to determine the prerequisite modules leading to them. The computer program shown in appendix 3 was used to systematically determine all the prerequisites required to teach each one of those key modules.

The input data drawn from the module diagram is shown in table 6.3. This data represents the module diagram of figure 6.1.

The five modules selected are: (1) Mechanical energy balance, (2) Simultaneous material and energy balances for

steady state systems, (3) Enthalpy-concentration charts, (4) Humidity charts, and (5) Unsteady state material and energy balances.

The final form of the path matrix and the prerequisite diagram for each of the above modules are shown in figures 6.2 to 6.6 respectively.

The module numbers shown in the prerequisite diagrams refer to the numbers given in table 6.1.

Unfortunately none of the modules in the system of figure 6.1 is of the OR type, therefore, in order to show an example including all three types of modules, the control theory course example is included in appendix 1.

TABLE 6.1

Module diagram analysis for Basic Principles and Calculations in Chemical Engineering.

List of modules:

1.- Units and dimensions.

2.- Systems of units and conversions.

3.- Temperature and pressure.

4.- Concentration and density.

5.- Stoichiometry.

6.- Simple material balance.

7.- Processes and processing units.

8.- Material balance using algebraic techniques and tie components.

9.- Ideal gas law and mixtures of ideal gases.

10.- Real gas relationships.

11.- Vapor pressure.

12.- Saturation.

13.- Material balance involving condensation and vaporization.

14.- Phase phenomena.

15.- Energy and energy transformations.

16.- Heat capacity.

17.- Enthalpy changes with and without change of phase.

18.- General energy balance.

19.- Mechanical energy balance.

20.- Heats of reaction, combustion, formation, solution, and mixing.

21.- Simultaneous material and energy balances for steady state systems.

22.- Enthalpy-concentration charts.

23.- Humidity charts.

24.- Material and energy balance for complex systems.

25.- Unsteady state material and energy balance.

TABLE 6.2

Description of the modules:

Module 1: Units and dimensions.

1.- Unit quantities.

2.- Dimensions.

3.- Equivalents.

4.- Conversion of units.

5.- Dimensionless variables.

Module 2: Systems of units and conversions.

1.- Basic units.

2.- Derived units.

3.- CGS system.

4.- SI system.

5.- MKS system.

6.- English system.

7.- g_c constant.

Module 3: Temperature and pressure.

1.- Measurement of temperature.

2.- Temperature scales.

3.- Absolute temperature.

4.- Conversion of temperature to different scales.

5.- Measurement of pressure.

6.- Units of pressure.

7.- Conversion of pressure units to different systems of units.

8.- Gauge pressure.

9.- Vacuum pressure.

10.- Absolute pressure.

Module 4: Concentration and density.

1.- Mass.

2.- Volume.

3.- Density.

4.- Specific gravity.

5.- Measurement of density and specific gravity.

6.- Special gravity scales.

7.- Conversion between gravity scales.

8.- Molecular weight.

9.- Mole.

10.- Weight fraction.

11.- Mole fraction.

12.- Component ratio.

13.- Molarity.

14.- Molality.

15.- Volume percent.

16.- Mole percent.

Module 5: Stoichiometry.

1.- Chemical equation.

2.- Concept of basis.

3.- Stoichiometry definitions,

a).- Limiting reactant.

b).- Excess reactant.

c).- % excess reactant.

d).- Excess air.

e).- Completely burn.

f).- Required amount of reactant.

g).- Conversion.

h).- Degree of completion.

i).- Selectivity.

j).- Yield.

k).- Incomplete reaction.

Module 6: Simple material balance.

1.- Material balance principle.

2.- System.

3.- Material balance problems with direct solution;

a).- Excess air problem.

b).- Drying problem.

c).- Distillation problem.

d).- Combustion problem.

e).- Crystallization problem.

Module 7: Process and processing units.

1.- Process

a).- Continuous.

b).- Batch.

c).- Semi-batch.

2.- Material flow diagrams;

a).- Systems.

b).- Streams:

i).- Recycle.
ii).- Purge.
iii).- Bypass.

3.- Processing units.

a).- Mix units.

b).- Split units.

c).- Reaction units.

Module 8: Material balance using algebraic techniques and tie components.

1.- Examples showing use of algebraic techniques;

a).- Mixing example.

b).- Distillation example.

c).- Simple process flowsheet example.

d).- Countercurrent stagewise mass transfer example.

2.- Tie components examples;

a).- Drying example.

b).- Dilution example.

c).- Combustion example.

d).- Distillation example.

3.- Problems involving recycle, bypass and purge;

a).- Recycle without chemical reaction example.

b).- Recycle with chemical reaction example.

c).- Bypass calculation example.

Module 9: Ideal gas laws and mixtures of ideal gases.

1.- Ideal gas laws;

a).- Charles' law.

b).- Boyle's law.

c).- Gay-Lussac's law.

d).- Dalton's law.

e).- Amagat's law.

f).- Concept of ideal gas.

2.- Standard conditions.

3.- Mathematical representation of ideal gas law.

4.- Universal gas constant "R".

5.- Gas density and specific gravity.

6.- Ideal gas mixtures;

a).- Partial pressure.

b).- Partial volume.

Module 10: Real gas relationships.

1.- Equations of state;

a).- Van der Waal's equation.

b).- Redlich-Kwong

c).- Beattie-Bridgeman

d).- Benedict-Webb-Rubin

e).- Other equations of state.

2.- Compressibility factor;

a).- Critical state.

b).- Reduced conditions.

c).- Use of compressibility factor.

3.- Gaseous mixtures;

a).- Equations of state.

b).- Average constants.

c).- Mean compressibility factor.

d).- Pseudo-critical properties.

e).- P-V-T relations for gas mixtures.

Module 11: Vapor pressure.

1.- Concept of vapor pressure.

2.- Boiling point.

3.- Dew point.

4.- Bubble point.

5.- Change of vapor pressure with temperature,

6.- Change of vapor pressure with pressure.

7.- Estimation of vapor pressure.

Module 12: Saturation.

1.- Saturated liquid.

2.- Saturated gas.

3.- Saturation problems.

4.- Partial saturation.

5.- Molal saturation.

6.- Absolute saturation.

7.- Humidity.

Module 13: Material balance involving condensation and vaporization.

1.- Dehydration example.

2.- Humidification example.

3.- Drying example.

Module 14: Phase phenomena.

1.- Phase rule.

2.- Phase phenomena of pure components;

a).- P-V-T diagrams for pure components.

3.- Phase phenomena of mixtures;

a).- Critical and pseudo-critical points for a mixture.

b).- Cricondentherm and cricondenbar points.

c).- Retrograde condensation.

Module 15: Energy and energy transformations.

1.- Concepts and units;

a).- Calorie and BTU.

b).- System;

i).- Closed.
ii).- Open.

c).- Surroundings

d).- Property;

i).- Intensive.
ii).- Extensive.

e).- State.

2.- Energy forms;

a).- Heat.

b).- Work.

c).- Kinetic energy.

d).- Potential energy.

e).- Internal energy.

f).- Enthalpy.

3.- Point functions.

Module 16: Heat capacity.

1.- Definition of C_p and C_v.

2.- Heat capacity and phase transitions.

3.- Specific heat.

4.- Heat capacity equation.

5.- Heat capacity for ideal gas.

6.- Estimation of heat capacities;

a).- Solids

b).- Liquids;

i).- Aqueous solutions.
ii).- Hydrocarbons.
iii).- Organic liquids.

c).- Gases and vapors;

i).- Petroleum vapors.
ii).- Kothari plotting.

Module 17: Enthalpy changes with and without change of phase.

1.- Enthalpy changes without change of phase.

2.- ΔH using heat capacity equations.

3.- ΔH from tabulated enthalpy values.

4.- ΔH for transitions;

a).- Heat of fusion.

b).- Heat of vaporization;

i).- Kistyakowsky equation.

ii).- Clausius-Clapeyron plots;

*).- Dühring plot.
**).- Othmer plot.

iv).- Watson correlation.

Module 18: General energy balance.

1.- Concept of macroscopic energy balance.

2.- General energy balance equation.

3.- Special cases from general energy balance;

a).- First law of thermodynamics for closed systems.

b).- No accumulation.

c).- Enthalpy balance.

4.- Special process associated with energy balance;

a).- Isothermal.

b).- Adiabatic.

c).- Isobaric.

d).- Isometric or Isocoric.

5.- Applications of energy balance.

Module 19: Mechanical energy balance.

1.- Reversible processes and mechanical energy balance.

2.- Applications of mechanical energy balance.

Module 20: Heats of reaction, combustion, formation, solution and mixing.

1.- Heat of reaction.

2.- Heat of formation.

3.- Standard heat of combustion.

4.- Heat of reaction at constant pressure or at constant volume.

5.- Heat of reaction for incomplete reactions.

6.- Heat of reaction at non-standard conditions.

7.- Temperature of reaction.

8.- Heat of solution.

9.- Heat of mixing.

Module 21: Simultaneous material and energy balances for steady state systems.

1.- Simultaneous material and energy balances;

a).- Distillation column example.

2.- Degrees of freedom in a process unit.

3.- Degrees of freedom in combined units.

Module 22: Enthalpy-concentration charts.

1.- Construction of an enthalpy-concentration chart.

2.- Applications of the H-x chart.

3.- Graphical solution of material balance using H-x charts.

Module 23: Humidity charts.

1.- Definitions;

a).- Humid heat.

b).- Humid voulme.

c).- Dry bulb temperature.

d).- Wet bulb temperature.

2.- Psychometric chart.

3.- Use of humidity chart;

a).- Properties of moist air from humidity charts.

b).- Heating at constant humidity.

c).- Cooling and humidity.

d).- Combined material and energy balances for cooling towers.

Module 24: Material and energy balance for complex systems.

1.- Hand solution of material and energy balance problems.

2.- Use of special purpose digital computer programs.

Module 25: Unsteady state material and energy balance.

1.- Unsteady state macroscopic balance.

2.- Example cases;

a).- Material balance in batch distillation.

b).- Unsteady state chemical reaction.

c).- Overall unsteady state balance for a simple process.

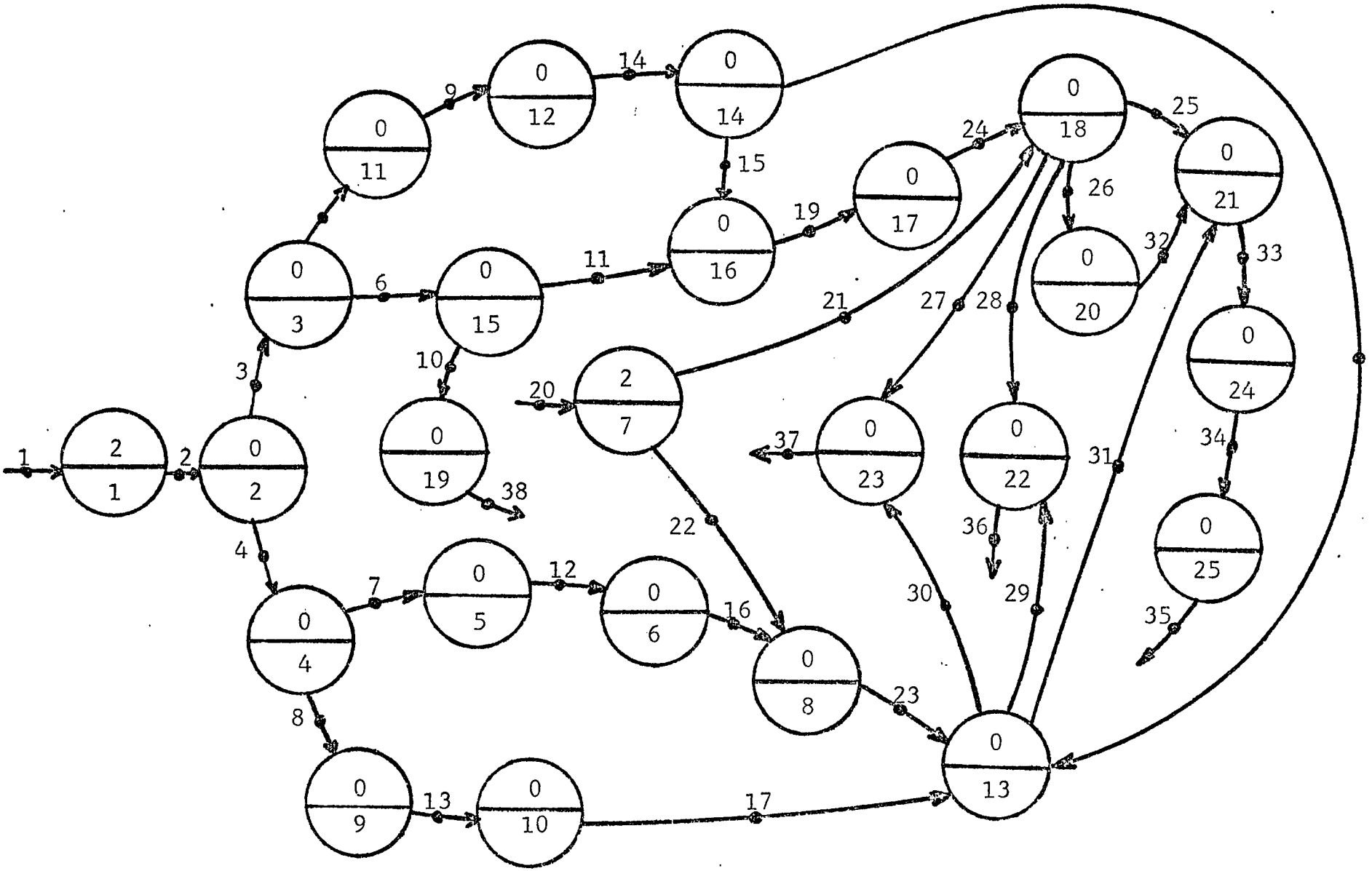


Figure 6.1 Module diagram for Stoichiometry course.

TABLE 6.3

STEP NUMBER	NODE IN	MODULE	NODE OUT	GATE TYPE
1	1	1	2	2
2	2	2	3	0
3	2	2	4	0
4	3	3	5	0
5	3	3	6	0
6	4	4	7	0
7	4	4	8	0
8	5	11	9	0
9	6	15	10	0
10	6	15	11	0
11	7	5	12	0
12	8	9	13	0
13	9	12	14	0
14	10	19	38	0
15	11	16	19	0
16	12	6	16	0
17	13	10	17	0
18	14	14	18	0
19	14	14	15	0
20	15	16	19	0
21	16	8	23	0
22	17	13	29	0
23	17	13	30	0
24	17	13	31	0
25	18	13	29	0
26	18	13	30	0
27	18	13	31	0
28	19	17	24	0

29	20	7	21	2
30	20	7	22	2
31	21	18	25	0
32	21	18	26	0
33	21	18	27	0
34	21	18	28	0
35	22	8	23	0
36	23	13	29	0
37	23	13	30	0
38	23	13	31	0
39	24	18	25	0
40	24	18	26	0
41	24	18	27	0
42	24	18	28	0
43	25	21	33	0
44	26	20	32	0
45	27	23	37	0
46	28	22	36	0
47	29	22	36	0
48	30	23	37	0
49	31	21	33	0
50	32	21	33	0
51	33	24	34	0
52	34	25	35	0

FINAL FORM OF PATH MATRIX:

```
-38
 19
 15
  3
  2
  1
  0
  0
  0
  0
  0
```

a)

Figure 6.2

THE FOLLOWING DIAGRAM SHOWS THE PREREQUISITE MODULES LEADING TO MODULE 19

```
 **
*19*
 **
 |
 AND
 |
 **
*15*
 **
 |
 AND
 |
 **
* 3*
 **
 |
 AND
 |
 **
* 2*
 **
 |
 AND
 |
 **
* 1*
 **
```

b)

Figure 6.2

```
FINAL FORM OF PATH MATRIX:


-33  0  0  0  0  0  0  0  0  0
 21  0  0  0  0  0  0  0  0  0
-25-99-99-99-99-99-99-99  0  0
 18  0  0 13  0  0  0 20  0  0
-21-99  0-17-99-99  0 18  0  0
  7 17  0 10 14  8  0-21-99  0
  0 16  0  9 12-16-99  7 17  0
  0-11-99  4 11  6  7  0 16  0
  0 15 14  2  3  5  0  0-11-99
  0  3 12  1  2  4  0  0 15 14
  0  2 11  0  1  2  0  0  3 12
  0  1  3  0  0  1  0  0  2 11
  0  0  2  0  0  0  0  0  1  3
  0  0  1  0  0  0  0  0  0  2
  0  0  0  0  0  0  0  0  0  1
  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0
```

a)

Figure 6.3

HE FOLLOWING DIAGRAM SHOWS THE PREREQUISITE MODULES LEADING TO MODULE 21

```
**
21*
**
|
AND
|
|
|
|
-----------------------------------------------------
|                       |                             |
|                       |                             |
**                      **                            **
18*                     *13*                          *20*
**                      **                            **
|                       |                             |
AND                     AND                           AND
|                       |                             |
|                       |                             **
|                       |                            *18*
|                       |                             **
--------                ---------------               |
|       |               |       |       |             AND
|       |               |       |       |             |
**      **              **      **      **            |
 7*    *17*             *10*    *14*    * 8*          |
**      **              **      **      **            |
        |               |       |       |             --------
        AND             AND     AND     AND           |       |
        |               |       |       |             |       |
        **              **      **      |             **      **
       *16*             * 9*    *12*    |            * 7*    *17*
        **              **      **      |             **      **
        |               |       |       --------              |
        AND             AND     AND     |       |             AND
        |               |       |       |       |             |
        |               **      **      **      **            **
        |              * 4*    *11*    * 6*    * 7*          *16*
        |               **      **      **      **            **
        --------        |       |       |                     |
        |       |       AND     AND     AND                   AND
        |       |       |       |       |                     |
        **      **      **      **      **                    |
       *15*    *14*    * 2*    * 3*    * 5*                  |
        **      **      **      **      **                    |
        |       |       |       |       |                     --------
        AND     AND     AND     AND     AND                   |       |
        |       |       |       |       |                     |       |
        **      **      **      **      **                    **      **
       * 3*    *12*    * 1*    * 2*    * 4*                 *15*    *14*
        **      **      **      **      **                    **      **
        |       |               |       |                     |       |
        AND     AND             AND     AND                   AND     AND
```

```
 |        |          |        |          |        |
 **       **         **       **         **       **
* 2*     *11*       * 1*     * 2*       * 3*     *12*
 **       **         **       **         **       **
 |        |                  |          |        |
AND      AND                 AND        AND      AND
 |        |                  |          |        |
 **       **                  **         **       **
* 1*     * 3*                * 1*       * 2*     *11*
 **       **                  **         **       **
          |                                |        |
         AND                               AND      AND
          |                                |        |
          **                               **       **
         * 2*                             * 1*     * 3*
          **                               **       **
          |                                |        |
         AND                               AND      AND
          |                                |        |
          **                               **       **
         * 1*                             * 2*     * 2*
          **                               **       **
                                           |        |
                                          AND      AND
                                           |        |
                                           **       **
                                          * 1*     * 1*
                                           **       **
```

b)

Figure 6.3

FINAL FORM OF PATH MATRIX:

-36	0	0	0	0	0	0
22	0	0	0	0	0	0
-28	-99	-99	-99	0	0	0
18	0	0	13	0	0	0
-21	-99	0	-17	-99	-99	0
7	17	0	10	14	8	0
0	16	0	9	12	-16	-99
0	-11	-99	4	11	6	7
0	15	14	2	3	5	0
0	3	12	1	2	4	0
0	2	11	0	1	2	0
0	1	3	0	0	1	0
0	0	2	0	0	0	0
0	0	1	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

a)

Figure 6.4

HE FOLLOWING DIAGRAM SHOWS THE PREREQUISITE MODULES LEADING TO MODULE 22

```
**
22*
**
|
AND
|
|
|
|
|
---------------------
|                     |
|                     |
**                    **
18*                  *13*
**                    **
|                     |
AND                   AND
|                     |
|                     |
|                     |
|                     |
--------              ---------------
|      |              |      |      |
|      |              |      |      |
**     **             **     **     **
 7*   *17*            *10*   *14*   * 8*
**     **             **     **     **
       |              |      |      |
       AND            AND    AND    AND
       |              |      |      |
       **             **     **     |
      *16*           * 9*   *12*    |
       **             **     **     |
       |              |      |      ---------
       AND            AND    AND    |       |
       |              |      |      |       |
       |              **     **     **     **
       |             * 4*   *11*   * 6*   * 7*
       |              **     **     **     **
       --------       |      |      |
       |      |       AND    AND    AND
       |      |       |      |      |
       **     **      **     **     **
      *15*   *14*    * 2*   * 3*   * 5*
       **     **      **     **     **
       |      |       |      |      |
       AND    AND     AND    AND    AND
       |      |       |      |      |
       **     **      **     **     **
      * 3*   *12*    * 1*   * 2*   * 4*
       **     **      **     **     **
       |      |              |      |
       AND    AND            AND    AND
```

```
 |        |              |        |
 **       **             **       **
* 2*     *11*           * 1*     * 2*
 **       **             **       **
 |        |                       |
 AND      AND                     AND
 |        |                       |
 **       **                      **
* 1*     * 3*                    * 1*
 **       **                      **
          |
          AND
          |
          **
         * 2*
          **
          |
          AND
          |
          **
         * 1*
          **
```

b)

Figure 6.4

FINAL FORM OF PATH MATRIX:

-37	0	0	0	0	0	0
23	0	0	0	0	0	0
-27	-99	-99	-99	0	0	0
18	0	0	13	0	0	0
-21	-99	0	-17	-99	-99	0
7	17	0	10	14	8	0
0	16	0	9	12	-16	-99
0	-11	-99	4	11	6	7
0	15	14	2	3	5	0
0	3	12	1	2	4	0
0	2	11	0	1	2	0
0	1	3	0	0	1	0
0	0	2	0	0	0	0
0	0	1	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

a)

Figure 6.5

HE FOLLOWING DIAGRAM SHOWS THE PREREQUISITE MODULES LEADING TO MODULE 23

```
**
23*
**
|
AND
|
|
|
|
---------------------
|                     |
|                     |
**                    **
18*                  *13*
**                    **
|                     |
AND                   AND
|                     |
|                     |
|                     |
|                     |
-------               ---------------
|      |              |      |      |
|      |              |      |      |
**     **             **     **     **
 7*   *17*            *10*   *14*   * 8*
**     **             **     **     **
       |              |      |      |
       AND            AND    AND    AND
       |              |      |      |
       **             **     **     |
      *16*            * 9*   *12*   |
       **             **     **     |
       |              |      |      --------
       AND            AND    AND    |      |
       |              |      |      |      |
       |              **     **     **     **
       |             * 4*   *11*   * 6*   * 7*
       |              **     **     **     **
       --------       |      |      |
       |      |       AND    AND    AND
       |      |       |      |      |
       **     **      **     **     **
      *15*   *14*    * 2*   * 3*   * 5*
       **     **      **     **     **
       |      |       |      |      |
       AND    AND     AND    AND    AND
       |      |       |      |      |
       **     **      **     **     **
      * 3*   *12*    * 1*   * 2*   * 4*
       **     **      **     **     **
       |      |              |      |
       AND    AND            AND    AND
```

```
 |        |                 |        |
 **       **                **       **
* 2*     *11*             * 1*     * 2*
 **       **                **       **
 |        |                          |
 AND      AND                       AND
 |        |                          |
 **       **                        **
* 1*     * 3*                     * 1*
 **       **                        **
          |
          AND
          |
          **
         * 2*
          **
          |
          AND
          |
          **
         * 1*
          **
```

b)

Figure 6.5

```
FINAL FORM OF PATH MATRIX:


-35  0  0  0  0  0  0  0  0  0
 25  0  0  0  0  0  0  0  0  0
 24  0  0  0  0  0  0  0  0  0
 21  0  0  0  0  0  0  0  0  0
-25-99-99-99-99-99-99-99  0  0
 18  0  0 13  0  0  0 20  0  0
-21-99  0-17-99-99  0 18  0  0
  7 17  0 10 14  8  0-21-99  0
  0 16  0  9 12-16-99  7 17  0
  0-11-99  4 11  6  7  0 16  0
  0 15 14  2  3  5  0  0-11-99
  0  3 12  1  2  4  0  0 15 14
  0  2 11  0  1  2  0  0  3 12
  0  1  3  0  0  1  0  0  2 11
  0  0  2  0  0  0  0  0  1  3
  0  0 -2  0  0  0  0  0  7  2
  0  0  1  0  0  0  0  0  0  1
  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0
```

a)

Figure 6.6

HE FOLLOWING DIAGRAM SHOWS THE PREREQUISITE MODULES LEADING TO MODULE 25

```
**
25*
**
|
AND
|
**
24*
**
|
AND
|
**
21*
**
|
AND
|
|
|
|
-----------------------------------------------
|                     |                         |
|                     |                         |
**                    **                        **
18*                  *13*                      *20*
**                    **                        **
|                     |                         |
AND                   AND                       AND
|                     |                         |
|                     |                         **
|                     |                        *18*
|                     |                         **
--------              ---------------           |
|      |              |      |      |           AND
|      |              |      |      |           |
**     **             **     **     **          |
 7*   *17*            *10*   *14*   * 8*        |
**     **             **     **     **          |
       |              |      |      |           --------
       AND            AND    AND    AND         |      |
       |              |      |      |           |      |
       **             **     **     |           **     **
      *16*            * 9*   *12*   |          * 7*   *17*
       **             **     **     |           **     **
       |              |      |      --------            |
       AND            AND    AND    |      |           AND
       |              |      |      |      |           |
       |              **     **     **     **          **
       |             * 4*   *11*   * 6*   * 7*        *16*
       |              **     **     **     **          **
       --------       |      |      |                 |
       |      |       AND    AND    AND               AND
```

```
 |       |       |       |       |                       |
 **      **      **      **      **                      |
*15*    *14*    * 2*    * 3*    * 5*                    |
 **      **      **      **      **                      |
 |       |       |       |       |                       --------
 AND     AND     AND     AND     AND                     |       |
 |       |       |       |       |                       |       |
 **      **      **      **      **                      **      **
* 3*    *12*    * 1*    * 2*    * 4*                   *15*    *14*
 **      **      **      **      **                      **      **
 |       |               |       |                       |       |
 AND     AND             AND     AND                     AND     AND
 |       |               |       |                       |       |
 **      **              **      **                      **      **
* 2*    *11*            * 1*    * 2*                    * 3*    *12*
 **      **              **      **                      **      **
 |       |                       |                       |       |
 AND     AND                     AND                     AND     AND
 |       |                       |                       |       |
 **      **                      **                      **      **
* 1*    * 3*                    * 1*                    * 2*    *11*
 **      **                      **                      **      **
         |                                               |       |
         AND                                             AND     AND
         |                                               |       |
         **                                              **      **
        * 2*                                            * 1*    * 3*
         **                                              **      **
         |                                               |       |
         AND                                             ----    AND
         |                                               |       |
         |                                               **      **
         |                                              * 7*    * 2*
         |                                               **      **
         |                                                       |
         |                                                       AND
         |                                                       |
         **                                                      **
        * 1*                                                    * 1*
         **                                                      **
```

b)

Figure 6.6

CHAPTER 7

CONCLUSIONS AND RECOMENDATIONS

The concepts and algorithm described in the previous chapters show an appropiate way for the curriculum analysis. The described module diagram is quite easy to draw and has the advantages that it is sequence oriented and easy to analyze. Logical sequences to teach the different subjects are implicit in the prerequisite diagram, all that is required to determine such sequences is to find out what the prerequisite modules are and the order in which they appear in the prerequisite diagram.

This type of analysis may be applied to all the courses in the curriculum of certain field of studies, this will provide the relationship between each topic, making possible to determine the concepts involved in teaching any given subject and if necessary to modify the order in which the concepts are taught so that no overlaping or out of sequence teaching occurs, therefore this curriculum analysis technique may help to redesign or improve an existing curriculum.

By making a module diagram analysis of a course, it is possible to design a set of units that will make this particular course easy to teach and very appropiate for a sel-paced or self-taught kind of program since the student will know in advance what are the basic concepts he needs to

understand in order to be able to learn one particular subject. This kind of module diagram would be prepared by an expert or a team of experts in the field since it is required to have a thorough knowledge of the topics involved to be able to successfully determine the relationships among the different concepts.

A further application of this analysis will include the use of estimated time required to teach each subject, this will help to determine how much time should be spent in each subject so that all the topics will be considered within the expected time. This will help to avoid the situation where all attention and effort is concentrated on just part of the material that was supposed to be covered in a fraction of the time actually used.

BIBLIOGRAPHY

BIBLIOGRAPHY

1. Bass L., Wynholds H.W., and Porterfield W.R.,"Fault Tree Graphics", Presented at the conference on reliability and fault tree analysis, University of California, Berkeley, Sept. 1974.

2. Caceres S.,"Fault Trees for Block Diagram Analysis", M.S. Thesis, University of Houston, Houston 1974.

3. Caceres S., and Henley E.J.,"Process Reliability Analysis by Block Diagram", Ind. Eng. Chem., to be published.

4. Fussel J.B., "Fault Tree Analysis - Concepts and Techniques", in Generic Techniques in Reliability Assessment, E.J. Henley, and J. Lynn Eds., Nordhoff Publishing Co., 1974.

5. Fussel J.B., "Synthetic Tree Model, a Formal Methodology for Fault Tree Construction", Presented at the Conference on Reliability and Fault Tree Analysis, University of California, Berkeley, Sept. 1974.

6. Genna P.L., and Motard R.L.,"Optimal Decomposition of Process Networks", AICHE J., Vol 21, No.4,656, (July 1975).

7. Henley E.J., and Gandhi S.L.,"Process Reliability Analysis", AICHE J.,Vol.21,No.6,677,(July 1975).

8. Henley E.J., and Rosen E.M., Material and Energy Balance Computations, John Willey and Sons Inc., New York 1969.

9. Henley E.J.,and Williams R.A., Graph Theory in Modern Engineering, Academic Press, New York, 1973.

10. Himmelblau D.M., Basic Principles and Calculations in Chemical Engineering., Prentice-Hall Inc., New Jersey, Third Edition, 1974.

11. Inoue K., Personal communication.

12. Powers G.J.,and Tompkins F.L. Jr.,"Fault Tree Synthesis for Chemical Processes",AICHE J.,Vol.20,376, (1974).

13. Powers G.J., Cummings D.L., and Rathore R.N.S.,"Synthesis of Fault Tolerant Reactions Paths",AICHE J., Vol.21,No.1,90,(Jan 1975).

14. Rodrigo F.R., and Seader J.D.,"Synthesis of Separation Sequences by Ordered Branch Search", AICHE J., Vol.21, No.5, 885, (Sept. 1975).

15. Tou J.T., Modern Control Theory, McGraw Hill, New York, 1964.

16. Vesely W.E., "PREP and KITT, Computer Codes for the Automatic Evauation of a Fault Tree", IN-1340, Aug. 1970.

17. Williams R.A., "Systematic Flow Graph Analysis ans Applications", Ph.D. Thesis, University of Houston, Houston, 1971.

APPENDICES

APPENDIX 1

MODULE DIAGRAM ANALYSIS APPLIED TO A CONTROL THEORY COURSE

APPENDIX 1

Module diagram analysis applied to a control theory course

The module diagram for this system is based on a diagram developed by Inoue (11) for the analysis of a Modern Control Theory course using a book by Tou (15) as textbook. A list of the modules used in the diagram is given in table A1.1. A module diagram for the system is shown in figure A1.1

The module diagram technique described in chapters 4 and 5 is used to analyze five of the key modules of the system. Table A1.2 shows the input data for the computer program.

The modules analyzed are: (1) Controlability and Observability, (2) Dead Beat Preformance System Design, (3) Minimal Integral Control, (4) Time Optimal Control, and (5) Terminal Control. The final form of the path matrix and the prerequiisite diagram for each of these modules are shown in figures A1.2 to A1.6 respectively.

TABLE A1.1

Modules for Modern Control Theory

1.- Set Theory.

2.- Eigenvalue and Quadratic forms.

3.- Finite Dimension Vector Space.

4.- Matrix Theory.

5.- Vector-Matrix Differential Equations.

6.- Calculus of Variations.

7.- Generalized Coordinate.

8.- Probability Theory.

9.- Lagrange Multiplier Method.

10.- State Space.

11.- Optimization by Variational Calculus.

12.- State Transition Method.

13.- Maximum Principle.

14.- Dynamic Programing.

15.- Controlability and Observability.

16.- Dead Beat Preformance System Design.

17.- Dummy module.

18.- Stochastic and Adaptive Control.

19.- Minimal Integral Control.

20.- Time Optimal Control.

21.- Terminal Control.

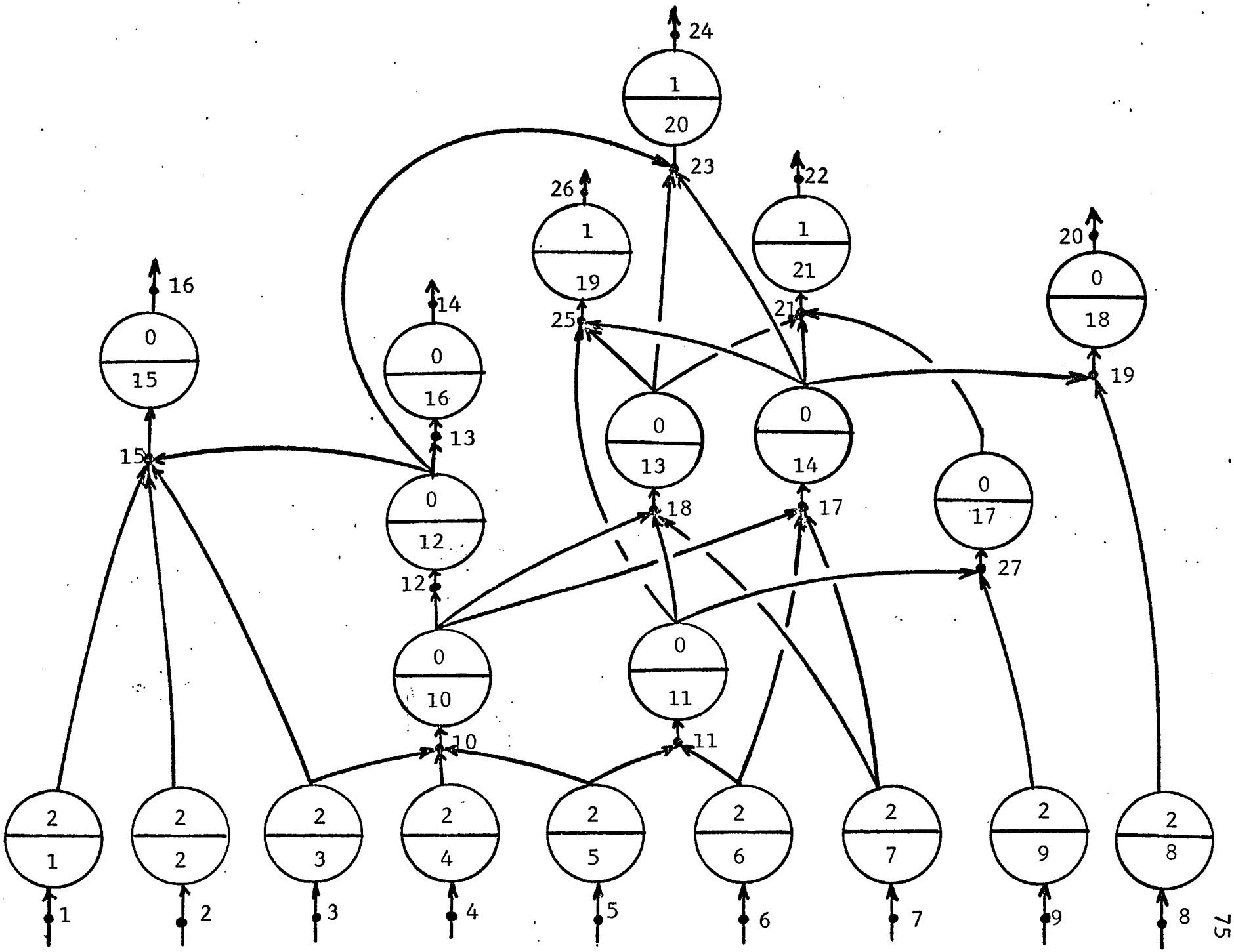


Figure A1.1 Module diagram for Modern Control Theory course.

TABLE A1.2

STEP NUMBER	NODE IN	MODULE	NODE OUT	GATE TYPE
1	1	1	15	2
2	2	2	15	2
3	3	3	15	2
4	3	3	10	2
5	4	4	10	2
6	5	5	10	2
7	5	5	11	2
8	6	6	11	2
9	6	6	17	2
10	7	7	17	2
11	7	7	18	2
12	8	8	19	2
13	9	9	27	2
14	10	10	12	0
15	10	10	18	0
16	10	10	17	0
17	11	11	25	0
18	11	11	27	0
19	11	11	18	0
20	18	13	21	0
21	27	17	21	0
22	12	12	15	0
23	12	12	13	0
24	12	12	23	0
25	13	16	14	0
26	15	15	16	0
27	17	14	25	0
28	17	14	23	0
29	17	14	21	0
30	17	14	19	0
31	18	13	25	0
32	18	13	23	0
33	19	18	20	0
34	21	21	22	1
35	23	20	24	1
36	25	19	26	1

Table A1.2 Input data for Computer program

FINAL FORM OF PATH MATRIX:

-16	0	0	0	0	0
15	0	0	0	0	0
-15	-99	-99	-99	0	0
1	2	3	12	0	0
0	0	0	10	0	0
0	0	0	-10	-99	-99
0	0	0	3	4	5
0	0	0	0	0	0
0	0	0	0	0	0

THE FOLLOWING DIAGRAM SHOWS THE PREREQUISITE MODULES LEADING TO MODULE 15

```
 **
*15*
 **
 |
 AND
 |
 |
 |
 ----------------------
 |       |       |       |
 **      **      **      **
* 1*    * 2*    * 3*    *12*
 **      **      **      **
                         |
                         AND
                         |
                         **
                        *10*
                         **
                         |
                         AND
                         |
                         |
                         |
                         ---------------
                         |       |       |
                         **      **      **
                        * 3*    * 4*    * 5*
                         **      **      **
```

Figure A1.2

FINAL FORM OF PATH MATRIX:

-14	0	0
16	0	0
12	0	0
10	0	0
-10	-99	-99
3	4	5
0	0	0
0	0	0
0	0	0

THE FOLLOWING DIAGRAM SHOWS THE PREREQUISITE MODULES LEADING TO MODULE 16

```
 **
*16*
 **
 |
 AND
 |
 **
*12*
 **
 |
 AND
 |
 **
*10*
 **
 |
 AND
 |
 |
 |
 ---------------
 |       |       |
 |       |       |
 **      **      **
* 3*    * 4*    * 5*
 **      **      **
```

Figure A1.3

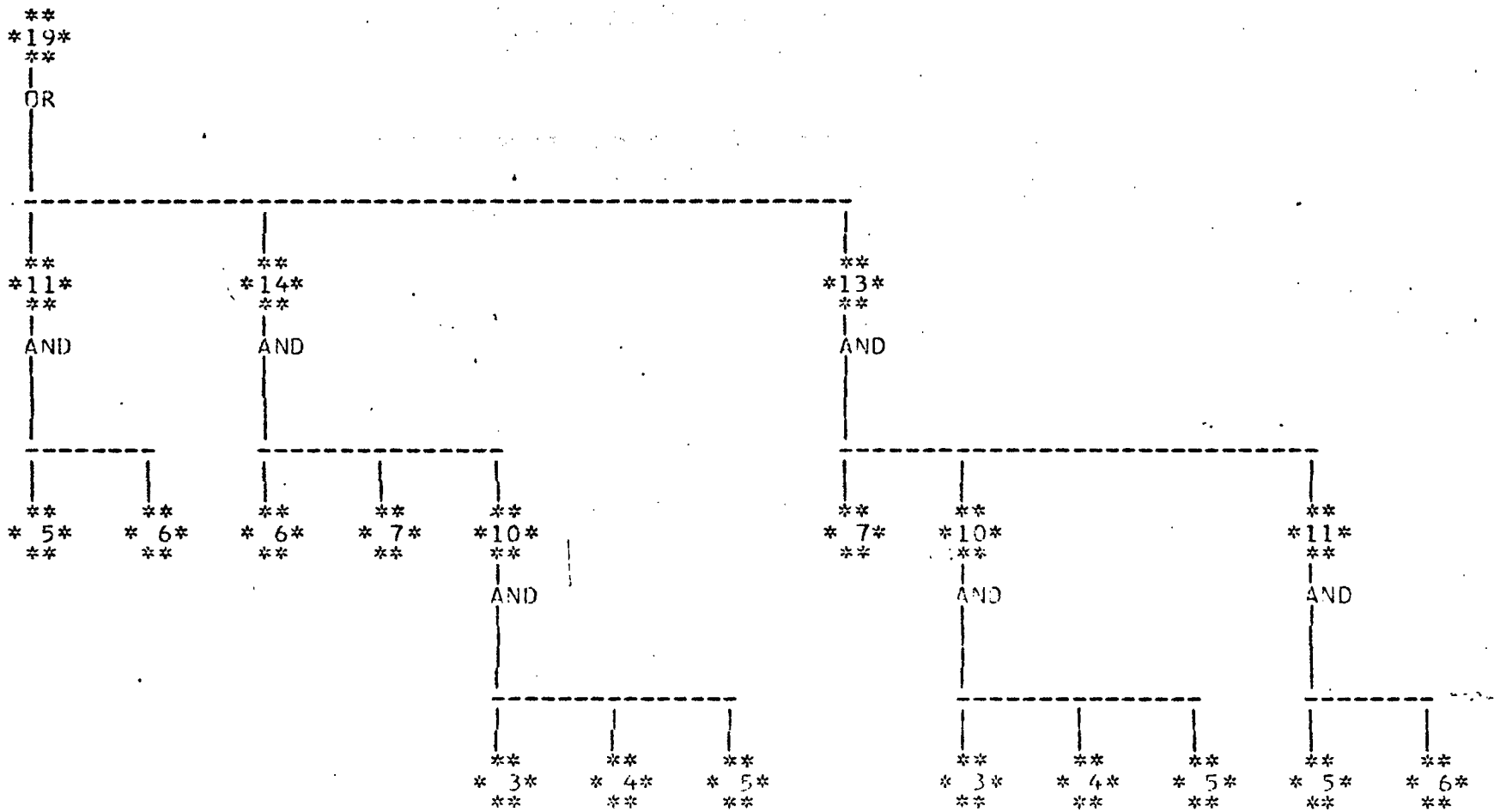
FINAL FORM OF PATH MATRIX:

-26	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0
-25	-99	-99	-99	-99	-99	-99	-99	0	0	0	0	0
11	0	14	0	0	0	0	13	0	0	0	0	0
-11	-99	-17	-99	-99	0	0	-18	-99	-99	-99	-99	0
5	6	6	7	10	0	0	7	10	0	0	11	0
0	0	0	0	-10	-99	-99	0	-10	-99	-99	-11	-99
0	0	0	0	3	4	5	0	3	4	5	5	6
0	0	0	0	0	0	0	0	0	0	0	0	0

Figure A1.4

Figure A1.4

THE FOLLOWING DIAGRAM SHOWS THE PREREQUISITE MODULES LEADING TO MODULE 19



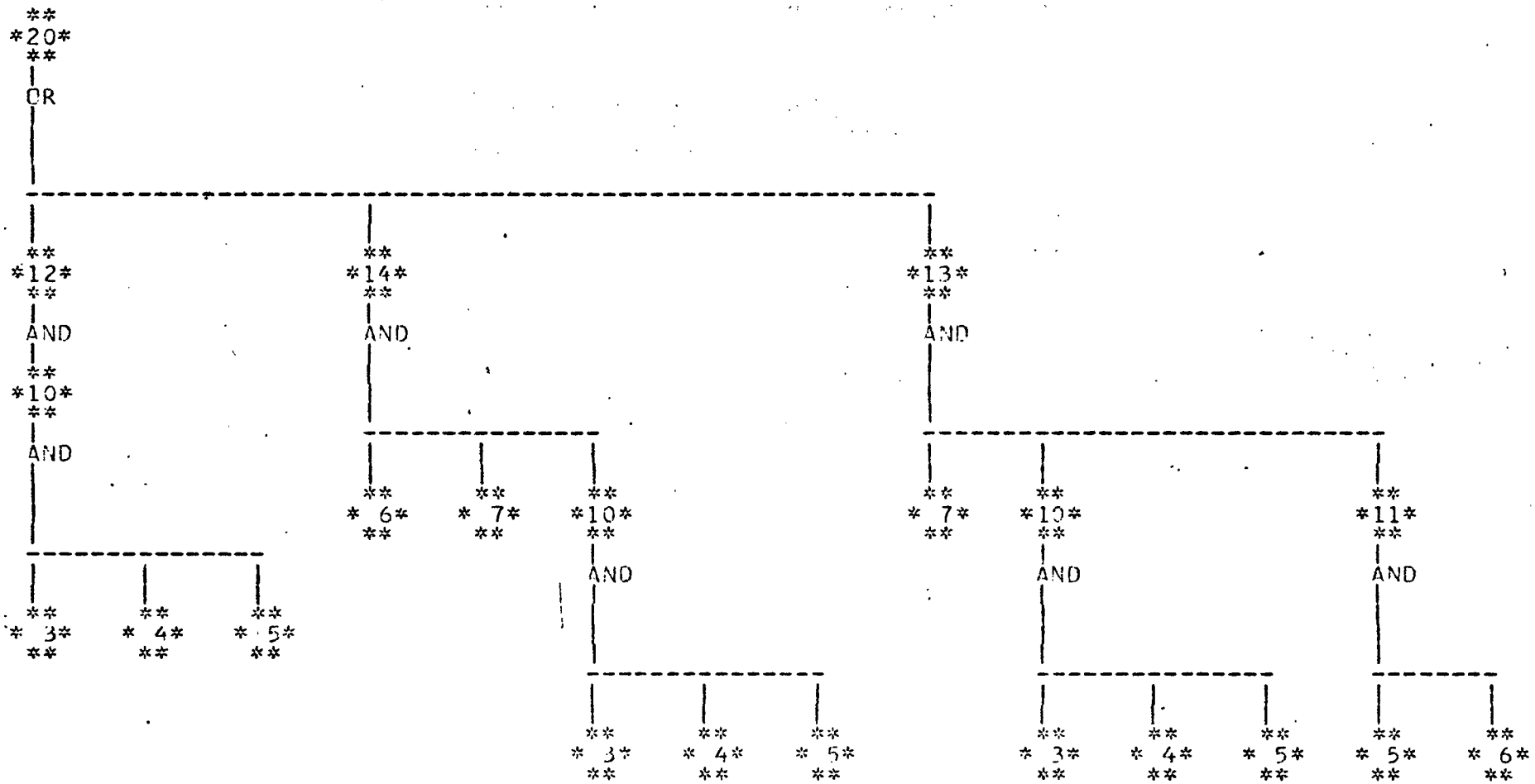
FINAL FORM OF PATH MATRIX:

-24	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0
-23	-99	-99	-99	-99	-99	-99	-99	-99	0	0	0	0	0
12	0	0	14	0	0	0	0	13	0	0	0	0	0
10	0	0	-17	-99	-99	0	0	-18	-99	-99	-99	-99	0
-10	-99	-99	6	7	10	0	0	7	10	0	0	11	0
3	4	5	0	0	-10	-99	-99	0	-10	-99	-99	-11	-99
0	0	0	0	0	3	4	5	0	3	4	5	5	6
0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure A1.5

Figure A1.5

THE FOLLOWING DIAGRAM SHOWS THE PREREQUISITE MODULES LEADING TO MODULE 20



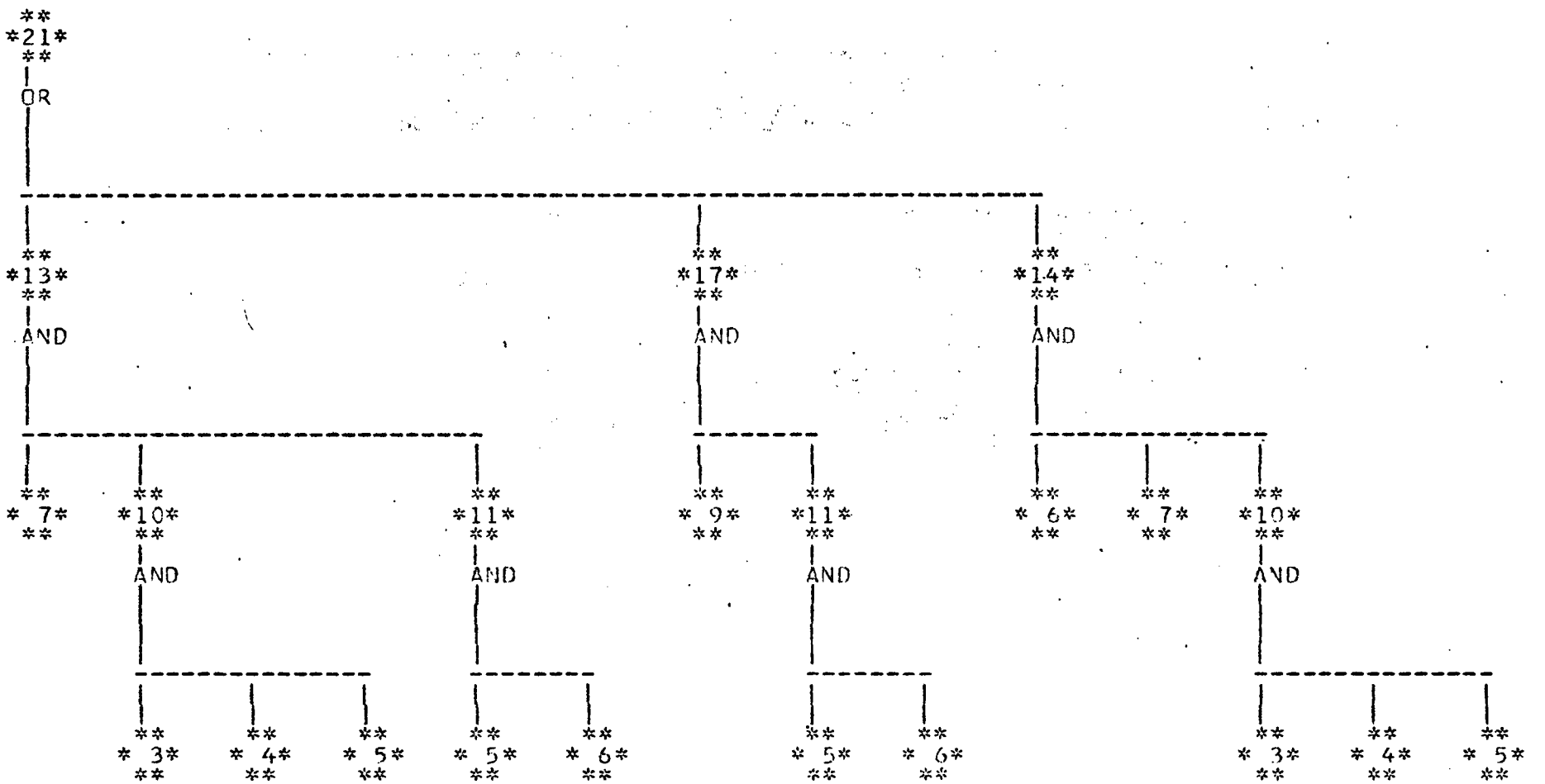
FINAL FORM OF PATH MATRIX:

-22	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0
-21	-99	-99	-99	-99	-99	-99	-99	-99	-99	0	0	0	0
13	0	0	0	0	0	17	0	0	14	0	0	0	0
-18	-99	-99	-99	-99	0	-27	-99	0	-17	-99	-99	0	0
7	10	0	0	11	0	9	11	0	6	7	10	0	0
0	-10	-99	-99	-11	-99	0	-11	-99	0	0	-10	-99	-99
0	3	4	5	5	6	0	5	6	0	0	3	4	5
0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure A1.6

Figure A1.6

THE FOLLOWING DIAGRAM SHOWS THE PREREQUISITE MODULES LEADING TO MODULE 21



APPENDIX 2

INPUT FORMS FOR PROGRAM

APPENDIX 2

INPUT FORMS FOR COMPUTER PROGRAM

The input data for the computer program is obtained from the module diagram. Figure A2.1 shows the input data required in a FORTRAN coding sheet.

FORTRAN Coding Form

PROGRAMMER

PROGRAM

DATE

JOB NO.

PAGE ____ OF ____

STATEMENT NUMBER | CONT. | FORTRAN STATEMENT | IDENTIFICATION SEQUENCE

```
      NNODE - NO. OF NODES
      NELE - NO. OF STEPS

      N - NO. OF NODES TO BE ANALYZED

      FLT - FIRST NODE TO BE ANALYZED

      GATE - FLAG GATE
      NDOUT - OUTPUT NODE                ONE CARD FOR EACH STEP
      BLOCK(I) - MODULE NO.                   (NELE CARDS)
      NDIN - INPUT NODE

      FLT        SECOND TO NTH NODE TO BE ANALYZED
                      (N-1 CARDS)
```

Figure A2.1

APPENDIX 3

LISTING OF COMPUTER PROGRAM

```
C          ***** MAIN PROGRAM *******                                    MAIN  10
C      PURPOSE:                                                          MAIN  20
C                                                                        MAIN  30
C          FIND THE PREREQUISITE MODULES LEADING TO A GIVEN MODULE       MAIN  40
C      IN A SYSTEM REPRESENTED BY A MODULE DIAGRAM.                      MAIN  50
C                                                                        MAIN  60
C      SUBROUTINES USED:                                                 MAIN  70
C                                                                        MAIN  80
C          PATHS                                                         MAIN  90
C          PRPTH                                                         MAIN 100
C          MOD1                                                          MAIN 110
C          MOD2                                                          MAIN 120
C          MOD3                                                          MAIN 130
C          MOD4                                                          MAIN 140
C          MOD5                                                          MAIN 150
C          MOD6                                                          MAIN 160
C          ANALY                                                         MAIN 170
C                                                                        MAIN 180
       IMPLICIT INTEGER*4(A-Z)                                           MAIN 190
       DIMENSION PATH(100,50),BLOCK(60)                                  MAIN 200
       DIMENSION GATE(60)                                                MAIN 210
C      INITIALIZE PATH MATRIX                                            MAIN 220
C                                                                        MAIN 230
       DATA PATH / 5000*0 /                                              MAIN 240
C      READ IN                                                           MAIN 250
C          NELE - NO. OF STEPS                                           MAIN 260
C          NNODE - NO. OF NODES IN MODULE DIAGRAM                        MAIN 270
C                                                                        MAIN 280
       READ(5,1)NELE,NNODE                                               MAIN 290
C      READ IN                                                           MAIN 300
C          N - NO. OF NODES TO BE ANALYZED                               MAIN 310
       READ(5,2)N                                                        MAIN 320
       CHECK=0                                                           MAIN 330
C                                                                        MAIN 340
       DO 10 I=1,N                                                       MAIN 350
```

```
C       READ IN                                                         MAIN 360
C           FLT - NODE TO BE ANALYZED                                   MAIN 370
C                                                                       MAIN 380
        READ(5,11)FLT                                                   MAIN 390
        CALL PATHS(NELE,NNODE,FLT,PATH,BLOCK,NP,LGTH,GATE,CHECK)        MAIN 400
        IF(NP.LT.100.AND.LGTH.LE.48)GO TO 4                             MAIN 410
        WRITE(6,3) NP,LGTH                                              MAIN 420
        GO TO 8                                                         MAIN 430
4       WRITE(6,5)                                                      MAIN 440
5       FORMAT(///,10X,'PATH MATRIX:')                                  MAIN 450
        CALL PRPTH(PATH,NP,LGTH)                                        MAIN 460
        CALL MOD1(PATH,NP,LGTH)                                         MAIN 470
        CALL MOD2(PATH,NP,LGTH)                                         MAIN 480
        CALL MOD3(PATH,NP,LGTH)                                         MAIN 490
        CALL MOD4(PATH,NP,LGTH)                                         MAIN 500
        CALL MOD5(PATH,NP,LGTH)                                         MAIN 510
        CALL MOD6(PATH,NP,LGTH)                                         MAIN 520
        WRITE(6,6)                                                      MAIN 530
        CALL PRPTH(PATH,NP,LGTH)                                        MAIN 540
        WRITE(6,7)                                                      MAIN 550
        CALL ANALY(NELE,PATH,BLOCK,NP,LGTH,GATE)                        MAIN 560
10      CONTINUE                                                        MAIN 570
C           FORMATS                                                     MAIN 580
1       FORMAT(2I5)                                                     MAIN 590
2       FORMAT(I5)                                                      MAIN 600
3       FORMAT(1H1,//,3X,'WARNING: THE PATH MATRIX',                    MAIN 610
       1 ' NEEDS TO BE DIMENSIONED(',I4,',',I4,')')                     MAIN 620
6       FORMAT(///,10X,'FINAL FORM OF PATH MATRIX:')                    MAIN 630
7       FORMAT(1H1)                                                     MAIN 640
11      FORMAT(15I5)                                                    MAIN 650
8       STOP                                                            MAIN 660
        END                                                             MAIN 670
```

```
      SUBROUTINE PATHS(NELE,NNODE,FLT,PATH,BLOCK,NP,LGTH,GATE,CHECK)
C     USAGE:
C
C         CALL PATHS(NELE,NNODE,FLT,PATH,BLOCK,NP,LGTH,GATE)
C
C     PARAMETERS:
C
C         NELE   -   NO. OF STEPS
C         NNODE  -   NO. OF NODES IN MODULE DIAGRAM
C         FLT    -   NODE GEING ANALYZED
C         PATH   -   MINIMAL PATHS MATRIX
C         BLOCK  -   MODULES ARRAY
C         NP     -   NO. OF MINIMAL PATHS ( NO. COLUMNS OF PATH MATRIX)
C         LGTH   -   NO. OF ROWS OF PATH MATRIX
C         GATE   -   FLAG GATE ARRAY
C         CHECK  -   IDENTIFICATION FLAG
C
C     PURPOSE:
C
C         TO DETERMINE THE MINIMAL PATHS LEADING TO NODE "FLT" USING
C     CACARES ALGORITHM.
C
C     SUBROUTINES USED:
C
C     NONE
C
      IMPLICIT INTEGER*4 (A-Z)
      DIMENSION PATH(100,50),CLASS(50,50)
      DIMENSION KTER(50),BLOCK(60)
      DIMENSION GATE(60)
      IF(CHECK .EQ. 1) GO TO 41
      WRITE(6,5)
C     INITIALIZE
C
      DO 10 I = 1, NNODE
```

```
      CLASS(I,1) = 0                                                    PATH 360
      KTER(I) = 1                                                       PATH 370
10    CONTINUE                                                          PATH 380
C     READ FOR EACH MODULE;                                             PATH 390
C         NDIN   -   INPUT NODE                                         PATH 400
C         BLOCK  -   MODULE NUMBER                                      PATH 410
C         NDOUT  -   OUTPUT NODE                                        PATH 420
C         GATE   -   FLAG GATE TO IDENTIFY MODULE TYPE                  PATH 430
C                                                                       PATH 440
      DO 30 I = 1,NELE                                                  PATH 450
      READ(5,20)NDIN,BLOCK(I),NDOUT,GATE(I)                             PATH 460
      WRITE(6,25)I,NDIN,BLOCK(I),NDOUT,GATE(I)                          PATH 470
C     FORM THE NON-EMPTY CLASSES OF THE SYSTEM                          PATH 480
      CLASS(NDOUT,1) = CLASS(NDOUT,1) + 1                               PATH 490
      A = 2*CLASS(NDOUT,1)                                              PATH 500
      CLASS(NDOUT,A) = BLOCK(I)                                         PATH 510
      CLASS(NDOUT,A+1) =  NDIN                                          PATH 520
30    CONTINUE                                                          PATH 530
C     START PATH FINDING ALGORITHM                                      PATH 540
C                                                                       PATH 550
41    J = 0                                                             PATH 560
      LGTH = 0                                                          PATH 570
50    J = J + 1                                                         PATH 580
60    PATH(J,1) = FLT                                                   PATH 590
      CHECK = 1                                                         PATH 600
      I = 2                                                             PATH 610
      L = PATH(J,1)                                                     PATH 620
70    K = 2* KTER(L)                                                    PATH 630
      IF(CLASS(L,1).EQ.0 ) FLAG = 0                                     PATH 640
      IF(CLASS(L,1).EQ.0 ) GO TO 90                                     PATH 650
      PATH(J,I) = CLASS(L,K)                                            PATH 660
      I = I + 1                                                         PATH 670
      PATH(J,I) = CLASS (L,K+1)                                         PATH 680
      L = PATH(J,I)                                                     PATH 690
      IND = I - 2                                                       PATH 700
      I = I + 1                                                         PATH 710
```

```
      DO 80 A = 1,IND,2                                          PATH 720
      LL = PATH(J,A)                                             PATH 730
      IF(L.EQ.LL) FLAG = 1                                       PATH 740
      IF(L.EQ.LL) GO TO 90                                       PATH 750
80    CONTINUE                                                   PATH 760
      GO TO 70                                                   PATH 770
90    IND = IND + 1                                              PATH 780
      DO 110 A = 1,IND,2                                         PATH 790
      L = PATH(J,IND-A)                                          PATH 800
      LASS = CLASS(L,1)                                          PATH 810
      IF(KTER(L) + 1 .LE.  LASS) GO TO 100                       PATH 820
      KTER(L) = 1                                                PATH 830
      GO TO 110                                                  PATH 840
100   KTER(L) = KTER(L) + 1                                      PATH 850
      GO TO 120                                                  PATH 860
110      CONTINUE                                                PATH 870
120   DO 130 A = 1,NNODE                                         PATH 880
      IF(KTER(A) .NE.1) GO TO 140                                PATH 890
130   CONTINUE                                                   PATH 900
      IF(FLAG.EQ.0.AND.I-1.GT.LGTH) LGTH = I-1                   PATH 910
      NP = J                                                     PATH 920
      IF(FLAG.EQ. 1) NP = J-1                                    PATH 930
      GO TO 150                                                  PATH 940
140   IF(FLAG.EQ.0.AND.I-1.GT.LGTH) LGTH = I-1                   PATH 950
      IF (FLAG.EQ.0) GO TO 50                                    PATH 960
      IND = I - 1                                                PATH 970
      DO 145 A = 1,IND                                           PATH 980
      PATH(J,A) = 0                                              PATH 990
145   CONTINUE                                                   PATH1000
      GO TO 60                                                   PATH1010
5     FORMAT(1H1,//,10X,'   STEP NUMBER',10X,'NODE IN',          PATH1020
     1 3X,'MODULE',3X,'NODE OUT',3X,'GATE TYPE',//)              PATH1030
20    FORMAT(3X,I2,3X,I2,3X,I2,4X,I1)                            PATH1040
25    FORMAT(18X,I4,15X,I2,7X,I2,7X,I2,10X,I1,/)                 PATH1050
150   RETURN                                                     PATH1060
      END                                                        PATH1070
```

```
      SUBROUTINE PRPTH(P,NC,NL)                                         PRPT  10
C     USAGE:                                                            PRPT  20
C                                                                       PRPT  30
C         CALL PRPTH(P,NC,NL)                                           PRPT  40
C                                                                       PRPT  50
C     PARAMETERS:                                                       PRPT  60
C                                                                       PRPT  70
C         P       -   MINIMAL PATH MATRIX                               PRPT  80
C         NC      -   NO. OF COLUMNS OF PATH MATRIX                     PRPT  90
C         NL      -   NO. OF ROWS OF PATH MATRIX                        PRPT 100
C                                                                       PRPT 110
C     PURPOSE:                                                          PRPT 120
C                                                                       PRPT 130
C         TO PRINT PATH MATRIX                                          PRPT 140
C                                                                       PRPT 150
C     SUBROUTINES USED:                                                 PRPT 160
C                                                                       PRPT 170
C         NONE                                                          PRPT 180
C                                                                       PRPT 190
      IMPLICIT INTEGER*4 (A-Z)                                          PRPT 200
      DIMENSION P(100,50)                                               PRPT 210
      WRITE(6,10)                                                       PRPT 220
      DO 30 I = 1,NL                                                    PRPT 230
      WRITE(6,20)(P(J,I),J=1,NC)                                        PRPT 240
30    CONTINUE                                                          PRPT 250
C         FORMATS                                                       PRPT 260
C                                                                       PRPT 270
10    FORMAT(///)                                                       PRPT 280
20    FORMAT(1X,43I3,/)                                                 PRPT 290
      RETURN                                                            PRPT 300
      END                                                               PRPT 310
```

```
      SUBROUTINE MOD1(PATH,NP,LGTH)                                     MOD1  10.
C     USAGE:                                                            MOD1  20
C                                                                       MOD1  30
C         CALL MOD1(PATH,NP,LGTH)                                       MOD1  40
C                                                                       MOD1  50
C     PARAMETERS:                                                       MOD1  60
C                                                                       MOD1  70
C         PATH    -   PATH MATRIX                                       MOD1  80
C         NP      -   NO. OF COLUMNS                                    MOD1  90
C         LGTH    -   NO. OF ROWS                                       MOD1 100
C                                                                       MOD1 110
C     PURPOSE:                                                          MOD1 120
C                                                                       MOD1 130
C         FIRST MODIFICATION OF PATH MATRIX                             MOD1 140
C     SUBROUTINES USED:                                                 MOD1 150
C                                                                       MOD1 160
C         NONE                                                          MOD1 170
C                                                                       MOD1 180
      IMPLICIT INTEGER*4 (A-Z)                                          MOD1 190
      DIMENSION PATH(100,50)                                            MOD1 200
      DO 10 J =2,NP                                                     MOD1 210
      PATH(J,1) = 0                                                     MOD1 220
10    CONTINUE                                                          MOD1 230
      DO 50 I = 2,LGTH,2                                                MOD1 240
      A = PATH(1,I)                                                     MOD1 250
      DO 40 J =2,NP                                                     MOD1 260
      IF(PATH(J,I).EQ.A) GO TO 30                                       MOD1 270
20    A = PATH(J,I)                                                     MOD1 280
      GO TO 40                                                          MOD1 290
30    IF(PATH(J,I-1).NE. 0) GO TO 20                                    MOD1 300
      PATH(J,I) = 0                                                     MOD1 310
      PATH(J,I+1) = 0                                                   MOD1 320
40    CONTINUE                                                          MOD1 330
50    CONTINUE                                                          MOD1 340
      RETURN                                                            MOD1 350
```

```
36:         END
```

```
      SUBROUTINE MOD2(PATH,NP,LGTH)                                     MOD2  10
C     USAGE:                                                            MOD2  20
C                                                                       MOD2  30
C         CALL MOD2(PATH,NP,LGTH)                                       MOD2  40
C                                                                       MOD2  50
C     PARAMETERS:                                                       MOD2  60
C                                                                       MOD2  70
C         PATH   -   PATH MATRIX                                        MOD2  80
C         NP     -   NO. OF COLUMNS                                     MOD2  90
C         LGTH   -   NO. OF ROWS                                        MOD2 100
C                                                                       MOD2 110
C     PURPOSE:                                                          MOD2 120
C         SECOND MODIFICATION OF PATH MATRIX                            MOD2 130
C                                                                       MOD2 140
C     SUBROUTINES USED:                                                 MOD2 150
C                                                                       MOD2 160
C         NONE                                                          MOD2 170
C                                                                       MOD2 180
      IMPLICIT INTEGER*4 (A-Z)                                          MOD2 190
      DIMENSION PATH(100,50)                                            MOD2 200
      PATH(1,1) = -PATH(1,1)                                            MOD2 210
      N = NP - 1                                                        MOD2 220
      IF(NP .EQ. 1)  N = NP                                             MOD2 230
      L = LGTH - 2                                                      MOD2 240
      DO 20 I = 3,L,2                                                   MOD2 250
      DO 10 J = 1,N                                                     MOD2 260
      IF(PATH(J,I).EQ.0) GO TO 10                                       MOD2 270
      IF(PATH(J+1,I).NE.0) GO TO 10                                     MOD2 280
      A = J + 1                                                         MOD2 290
      DO 2 K = A,NP                                                     MOD2 300
      IF(PATH(K,I).EQ.0) GO TO 2                                        MOD2 310
      ST = K - 1                                                        MOD2 320
      GO TO 4                                                           MOD2 330
2     CONTINUE                                                          MOD2 340
      ST= NP                                                            MOD2 350
```

```
4       DO 6 K = A,ST                                   MOD2 360
        IF(PATH(K,I+1).EQ.0) GO TO 6                    MOD2 370
        PATH(J,I) = -PATH(J,I)                          MOD2 380
        GO TO 10                                        MOD2 390
6       CONTINUE                                        MOD2 400
10      CONTINUE                                        MOD2 410
  20      CONTINUE                                      MOD2 420
        RETURN                                          MOD2 430
        END                                             MOD2 440
```

```
      SUBROUTINE MOD3(PATH,NP,LGTH)                                     MOD3  10.
C     USAGE:                                                            MOD3  20
C                                                                       MOD3  30
C         CALL MOD3(PATH,NP,LGTH)                                       MOD3  40
C                                                                       MOD3  50
C     PARAMETERS:                                                       MOD3  60
C                                                                       MOD3  70
C         PATH    -   PATH MATRIX                                       MOD3  80
C         NP      -   NO. OF COLUMNS                                    MOD3  90
C         LGTH    -   NO. OF ROWS                                       MOD3 100
C                                                                       MOD3 110
C     PURPOSE:                                                          MOD3 120
C                                                                       MOD3 130
C         THIRD MODIFICATION OF PATH MATRIX                             MOD3 140
C                                                                       MOD3 150
C     SUBROUTINES USED:                                                 MOD3 160
C                                                                       MOD3 170
C         NONE                                                          MOD3 180
C                                                                       MOD3 190
      IMPLICIT INTEGER*4 (A-Z)                                          MOD3 200
      DIMENSION PATH(100,50)                                            MOD3 210
      DO 30 J =2,NP                                                     MOD3 220
      DO 20 I =2,LGTH,2                                                 MOD3 230
      IF(PATH(J,I).EQ.0) GO TO 20                                       MOD3 240
      DO 10 K = 1,J                                                     MOD3 250
      L= J-K                                                            MOD3 260
      IF(PATH(L,I).EQ.0) GO TO 10                                       MOD3 270
      PATH(J,LGTH+1) = PATH(L,I)                                        MOD3 280
      PATH(J,LGTH+2) = L                                                MOD3 290
      GO TO 30                                                          MOD3 300
   10 CONTINUE                                                          MOD3 310
   20 CONTINUE                                                          MOD3 320
   30 CONTINUE                                                          MOD3 330
      RETURN                                                            MOD3 340
      END                                                               MOD3 350
```

```
      SUBROUTINE MOD4(PATH,NP,LGTH)                                     MOD4  10
C     USAGE:                                                            MOD4  20
C                                                                       MOD4  30
C         CALL MOD4(PATH,NP,LGTH)                                       MOD4  40
C                                                                       MOD4  50
C     PARAMETERS:                                                       MOD4  60
C                                                                       MOD4  70
C         PATH    -    PATH MATRIX                                      MOD4  80
C         NP      -    NO. OF COLUMNS                                   MOD4  90
C         LGTH    -    NO. OF ROWS                                      MOD4 100
C                                                                       MOD4 110
C     PURPOSE:                                                          MOD4 120
C                                                                       MOD4 130
C         FOURTH MODIFICATION OF PATH MATRIX                            MOD4 140
C                                                                       MOD4 150
C     SUBROUTINES USED:                                                 MOD4 160
C                                                                       MOD4 170
C         NONE                                                          MOD4 180
C                                                                       MOD4 190
      IMPLICIT INTEGER*4 (A-Z)                                          MOD4 200
      DIMENSION PATH(100,50)                                            MOD4 210
      DO 70 J =1,NP                                                     MOD4 220
      K =2                                                              MOD4 230
      FLAG = 0                                                          MOD4 240
      DO 40 I = 3,LGTH,2                                                MOD4 250
      IF(PATH(J,I))10,20,30                                             MOD4 260
10    PATH(J,K) = PATH(J,I-1)                                           MOD4 270
      PATH(J,K+1) = PATH(J,I)                                           MOD4 280
      K = K + 2                                                         MOD4 290
      FLAG = 1                                                          MOD4 300
      GO TO 40                                                          MOD4 310
20    IF(FLAG.EQ.0) GO TO 40                                            MOD4 320
      IF(FLAG.EQ.1) GO TO 50                                            MOD4 330
30    PATH(J,K) = PATH(J,I-1)                                           MOD4 340
      K = K + 1                                                         MOD4 350
```

```
      FLAG = 1                                                          MOD4 360
40    CONTINUE                                                          MOD4 370
50    DO 60 L = K,LGTH                                                  MOD4 380
      PATH(J,L) = 0                                                     MOD4 390
60    CONTINUE                                                          MOD4 400
70    CONTINUE                                                          MOD4 410
      RETURN                                                            MOD4 420
      END                                                               MOD4 430
```

```
      SUBROUTINE MOD5(PATH,NP,LGTH)                                     MOD5  10
C     USAGE:                                                            MOD5  20
C                                                                       MOD5  30
C         CALL MOD5(PATH,NP,LGTH)                                       MOD5  40
C                                                                       MOD5  50
C     PARAMETERS:                                                       MOD5  60
C                                                                       MOD5  70
C         PATH    -    PATH MATRIX                                      MOD5  80
C         NP      -    NO. OF COLUMNS                                   MOD5  90
C         LGTH    -    NO. OF ROWS                                      MOD5 100
C                                                                       MOD5 110
C     PURPOSE:                                                          MOD5 120
C                                                                       MOD5 130
C         FIFTH MODIFICATION OF PATH MATRIX                             MOD5 140
C                                                                       MOD5 150
C     SUBROUTINES USED:                                                 MOD5 160
C                                                                       MOD5 170
C         NONE                                                          MOD5 180
C                                                                       MOD5 190
      IMPLICIT INTEGER*4 (A-Z)                                          MOD5 200
      DIMENSION PATH(100,50),COL(50)                                    MOD5 210
      DO 60 J = 2,NP                                                    MOD5 220
      K = 0                                                             MOD5 230
      DO 10 I = 2,LGTH                                                  MOD5 240
      IF(PATH(J,I).EQ.0) GO TO 20                                       MOD5 250
      K = K + 1                                                         MOD5 260
      COL(K) = PATH(J,I)                                                MOD5 270
      PATH(J,I) = 0                                                     MOD5 280
10    CONTINUE                                                          MOD5 290
20    A = PATH(J,LGTH+1)                                                MOD5 300
      B = PATH(J,LGTH+2)                                                MOD5 310
      DO 30 I = 1,LGTH                                                  MOD5 320
      IF(PATH(B,I).EQ.A) ST = I                                         MOD5 330
      IF(PATH(B,I).EQ.A) GO TO 40                                       MOD5 340
30    CONTINUE                                                          MOD5 350
```

```
36:  40     IF(K .EQ. 0) K = 1                          MOD5 360
37:         DO  50 I = 1,K                              MOD5 370
38:         A = I - 1                                   MOD5 380
39:         PATH(J,ST+A) = COL(I)                       MOD5 390
40:  50     CONTINUE                                    MOD5 400
41:  60     CONTINUE                                    MOD5 410
42:         RETURN                                      MOD5 420
43:         END                                         MOD5 430
```

```
      SUBROUTINE MOD6(PATH,NP,LGTH)                                     MOD6  10
C     USAGE:                                                            MOD6  20
C                                                                       MOD6  30
C         CALL MOD6(PATH,NP,LGTH)                                       MOD6  40
C                                                                       MOD6  50
C     PARAMETERS:                                                       MOD6  60
C                                                                       MOD6  70
C         PATH    -    PATH MATRIX                                      MOD6  80
C         NP      -    NO. OF COLUMNS                                   MOD6  90
C         LGTH    -    NO. OF ROWS                                      MOD6 100
C                                                                       MOD6 110
C     PURPOSE:                                                          MOD6 120
C                                                                       MOD6 130
C         FINAL MODIFICATION OF PATH MATRIX                             MOD6 140
C                                                                       MOD6 150
C     SUBROUTINES USED:                                                 MOD6 160
C                                                                       MOD6 170
C         NONE                                                          MOD6 180
C                                                                       MOD6 190
      IMPLICIT INTEGER*4 (A-Z)                                          MOD6 200
      DIMENSION PATH(100,50)                                            MOD6 210
      DO 40 J = 2,NP                                                    MOD6 220
      K = NP + 2 - J                                                    MOD6 230
      FLAG = 0                                                          MOD6 240
      DO 10 I = 1,LGTH                                                  MOD6 250
      L = LGTH +1-I                                                     MOD6 260
      IF(PATH(K,L).NE.0) FLAG = 1                                       MOD6 270
      IF(PATH(K,L).EQ.-99) GO TO 40                                     MOD6 280
      IF(PATH(K,L).EQ.0.AND.FLAG.EQ.1) GO TO 20                         MOD6 290
10    CONTINUE                                                          MOD6 300
20    KKK = K - 1                                                       MOD6 310
      IF(KKK .EQ. 0) KKK = 1                                            MOD6 320
      DO 30 JJ = 1,KKK                                                  MOD6 330
      KK = K + 1 - JJ                                                   MOD6 340
      IF(PATH(KK,L).NE.0) GO TO 40                                      MOD6 350
```

```
36:          IF(PATH(KK,L).EQ.0) PATH(KK,L) = -99          MOD6 360
37:   30     CONTINUE                                      MOD6 370
38:   40     CONTINUE                                      MOD6 380
39:          RETURN                                        MOD6 390
40:          END                                           MOD6 400
```

```
      SUBROUTINE ANALY(NELE,PATH,BLOCK,NP,LGTH,GATE)                    ANAL  10
C     USAGE                                                             ANAL  20
C                                                                       ANAL  30
C         CALL ANALY(NELE,PATH,BLOCK,NP,LGTH,GATE)                      ANAL  40
C                                                                       ANAL  50
C     PARAMETERS:                                                       ANAL  60
C                                                                       ANAL  70
C         NELE   -   NO. OF STEPS                                       ANAL  80
C         PATH   -   PATHS MATRIX                                       ANAL  90
C         BLOCK  -   MODULES ARRAY                                      ANAL 100
C         NP     -   NO. OF COLUMNS OF PATH MATRIX                      ANAL 110
C         LGTH   -   NO. OF ROWS OF PATH MATRIX                         ANAL 120
C         GATE   -   FLAG GATE ARRAY                                    ANAL 130
C                                                                       ANAL 140
C     PURPOSE:                                                          ANAL 150
C                                                                       ANAL 160
C         TO CLASSIFY EACH ELEMENT OF THE PATH MATRIX IN ITS FINAL FORM ANAL 170
C     TO DRAW THE PREREQUISITE DIAGRAM                                  ANAL 180
C                                                                       ANAL 190
C     SUBROUTINES USED:                                                 ANAL 200
C                                                                       ANAL 210
C                                                                       ANAL 220
C         INIT                                                          ANAL 230
C         FINAL                                                         ANAL 240
C         BLOO                                                          ANAL 250
C         LIN1                                                          ANAL 260
C         LINE                                                          ANAL 270
C         TITLE                                                         ANAL 280
C                                                                       ANAL 290
      IMPLICIT INTEGER*4 (A-Z)                                          ANAL 300
      DIMENSION PATH(100,50),BLOCK(60),M(6,1000)                        ANAL 310
      DIMENSION GATE(60)                                                ANAL 320
      K = 4*NP                                                          ANAL 330
      DO 40 I=1,LGTH                                                    ANAL 340
      CALL INIT(M,NP)                                                   ANAL 350
```

```
      FLAG=0                                                    ANAL 360
      DO 10 J=1,NP                                              ANAL 370
      IF(PATH(J,I).EQ.0) GO TO 10                               ANAL 380
      FLAG =1                                                   ANAL 390
      IF(PATH(J,I).GT.0.AND.PATH(J,I+1).EQ.0)                   ANAL 400
     1   CALL FINAL(PATH,M,I,J)                                 ANAL 410
      IF(PATH(J,I).GT.0.AND.PATH(J,I+1).NE.0)                   ANAL 420
     1   CALL BLOQ(NELE,PATH,M,I,J,GATE,BLOCK)                  ANAL 430
      IF(PATH(J,I).LT.0.AND.PATH(J,I).NE.-99)                   ANAL 440
     1   CALL LIN1(PATH,M,I,J)                                  ANAL 450
      IF(PATH(J,I).LT.0.AND.PATH(J,I).EQ.-99)                   ANAL 460
     1   CALL LINE(PATH,M,I,J)                                  ANAL 470
      IF(I .EQ.1 .AND. J .EQ. 1) CALL FIRST(PATH,M,I,J)         ANAL 480
      IF(J .EQ. 1 .AND. I .EQ. 1) CALL TITLE(PATH,I,J)          ANAL 490
10    CONTINUE                                                  ANAL 500
      IF(FLAG.EQ.0) GO TO 50                                    ANAL 510
C     WRITE RESULTS                                             ANAL 520
      DO 30 L=1,6                                               ANAL 530
      WRITE(6,20)(M(L,N),N=1,K)                                 ANAL 540
30    CONTINUE                                                  ANAL 550
40    CONTINUE                                                  ANAL 560
C         FORMATS                                               ANAL 570
20    FORMAT(1X,18(2A2,A1,A2))                                  ANAL 580
50    RETURN                                                    ANAL 590
      END                                                       ANAL 600
```

```
      SUBROUTINE INIT(M,NP)                                             INIT  10
C     USAGE:                                                            INIT  20
C                                                                       INIT  30
C         CALL INIT(M,NP)                                               INIT  40
C                                                                       INIT  50
C     PARAMETERS:                                                       INIT  60
C                                                                       INIT  70
C         M       -   PREREQUISITE DIAGRAM MATRIX                       INIT  80
C         NP      -   NO. OF COLUMNS OF PATH MATRIX                     INIT  90
C     PURPOSE:                                                          INIT 100
C                                                                       INIT 110
C         INITIALIZE "M" MATRIX                                         INIT 120
C                                                                       INIT 130
C     SUBROUTINES USED:                                                 INIT 140
C                                                                       INIT 150
C         NONE                                                          INIT 160
C                                                                       INIT 170
      IMPLICIT INTEGER*4 (A-Z)                                          INIT 180
      DIMENSION M(6,1000)                                               INIT 190
      DATA B,BB / ' ','  ' /                                            INIT 200
      K = 4*NP                                                          INIT 210
      DO 30 I = 1,6                                                     INIT 220
      DO 10 J =3,K,3                                                    INIT 230
      M(I,J) = B                                                        INIT 240
10    CONTINUE                                                          INIT 250
      DO 20 J = 1,K,3                                                   INIT 260
      M(I,J) = BB                                                       INIT 270
      M(I,J+1) = BB                                                     INIT 280
20    CONTINUE                                                          INIT 290
30    CONTINUE                                                          INIT 300
      RETURN                                                            INIT 310
      END                                                               INIT 320
```

```
      SUBROUTINE TITLE(PATH,I,J)                                        TITL  10
C     USAGE:                                                            TITL  20
C                                                                       TITL  30
C         CALL TITLE(PATH,I,J)                                          TITL  40
C       PARAMETERS:                                                     TITL  50
C         PATH  -   PATH MATRIX                                         TITL  60
C         I     -   ROW OF PATH MATRIX                                  TITL  70
C         J     -   COLUMN OF PATH MATRIX                               TITL  80
C                                                                       TITL  90
C     PURPOSE:                                                          TITL 100
C                                                                       TITL 110
C         PRINT TITLE OF PREREQUISITE DIAGRAM                           TITL 120
C     SUBROUTINE USED:                                                  TITL 130
C                                                                       TITL 140
C         NONE                                                          TITL 150
C                                                                       TITL 160
      IMPLICIT INTEGER*4 (A-Z)                                          TITL 170
      DIMENSION  NODE(60),PATH(100,50)                                  TITL 180
      DATA NODE / ' 1',' 2',' 3',' 4',' 5',' 6',' 7',' 8',' 9','10',    TITL 190
     1'11','12','13','14','15','16','17','18','19','20',                TITL 200
     2 '21','22','23','24','25','26','27','28','29','30',               TITL 210
     3'31','32','33','34','35','36','37','38','39','40',                TITL 220
     4'41','42','43','44','45','46','47','48','49','50',                TITL 230
     5'51','52','53','54','55','56','57','58','59','60' /               TITL 240
C                                                                       TITL 250
      TOP = PATH(J,I+1)                                                 TITL 260
      WRITE(6,15) NODE(TOP)                                             TITL 270
C                                                                       TITL 280
15    FORMAT(1H1,' THE FOLLOWING DIAGRAM SHOWS THE PREREQUISITE MODULES TITL 290
     1LEADING TO MODULE',2X,A2)                                         TITL 300
      RETURN                                                            TITL 310
      END                                                               TITL 320
```

```
      SUBROUTINE FIRST(PATH,M,I,J)                                      FIRS  10
C     USAGE:                                                            FIRS  20
C                                                                       FIRS  30
C         CALL FIRST(PATH,M,I,J)                                        FIRS  40
C                                                                       FIRS  50
C     PARAMETERS:                                                       FIRS  60
C                                                                       FIRS  70
C        PATH   -   PATH MATRIX                                         FIRS  80
C         M      -   PREREQUISITE DIAGRAM MATRIX                       FIRS  90
C         I      -   ROW OF PATH MATRIX                                FIRS 100
C         J      -   COLUMN OF PATH MATRIX                             FIRS 110
C                                                                       FIRS 120
C     PURPOSE:                                                          FIRS 130
C                                                                       FIRS 140
C         TRANSFORM THE FIRST ELEMENT OF PATH MATRIX INTO               FIRS 150
C     CORRESPONDING ELEMENTS OF "M" MATRIX                              FIRS 160
C                                                                       FIRS 170
C     SUBROUTINES USED:                                                 FIRS 180
C                                                                       FIRS 190
C         NONE                                                          FIRS 200
C                                                                       FIRS 210
      IMPLICIT INTEGER*4 (A-Z)                                          FIRS 220
      DIMENSION PATH(100,50),M(6,76)                                    FIRS 230
      DATA BB / '  ' /                                                  FIRS 240
      K = 4*(J-1) + 1                                                   FIRS 250
      M(1,K+1)=BB                                                       FIRS 260
      M(2,K+1)=BB                                                       FIRS 270
      M(3,K+1)=BB                                                       FIRS 280
      M(4,K+1)=BB                                                       FIRS 290
      M(5,K+1)=BB                                                       FIRS 300
      M(6,K+1)=BB                                                       FIRS 310
      RETURN                                                            FIRS 320
      END                                                               FIRS 330
```

```
C     USAGE:                                                            FINA  10
C                                                                       FINA  20
C         CALL FINAL(PATH,M,I,J)                                        FINA  30
C                                                                       FINA  40
C     PARAMETERS:                                                       FINA  50
C                                                                       FINA  60
C        PATH   -   PATH MATRIX                                         FINA  70
C         M      -   PREREQUISITE DIAGRAM MATRIX                       FINA  80
C         I      -   ROW OF PATH MATRIX                                FINA  90
C         J      -   COLUMN OF PATH MATRIX                             FINA 100
C                                                                       FINA 110
C     PURPOSE:                                                          FINA 120
C                                                                       FINA 130
C         CLASSIFY AND TRANSFORM AN ELEMENT OF PATH MATRIX              FINA 140
C     REPRESENTING A BASIC MODULE.                                      FINA 150
C                                                                       FINA 160
C     SUBROUTINES USED:                                                 FINA 170
C                                                                       FINA 180
C         NONE                                                          FINA 190
C                                                                       FINA 200
      SUBROUTINE FINAL(PATH,M,I,J)                                      FINA 210
      IMPLICIT INTEGER*4 (A-Z)                                          FINA 220
      DIMENSION PATH(100,50),MODE(60),M(6,76)                           FINA 230
      DATA MODE / ' 1',' 2',' 3',' 4',' 5',' 6',' 7',' 8',' 9','10',    FINA 240
     1'11','12','13','14','15','16','17','18','19','20',                 FINA 250
     2 '21','22','23','24','25','26','27','28','29','30',                FINA 260
     3'31','32','33','34','35','36','37','38','39','40',                 FINA 270
     4'41','42','43','44','45','46','47','48','49','50',                 FINA 280
     5'51','52','53','54','55','56','57','58','59','60' /                FINA 290
      DATA A,AA,BA / '*','**',' *' /                                    FINA 300
      K = 4*(J-1) + 1                                                   FINA 310
      LET = PATH(J,I)                                                   FINA 320
      M(2,K) = BA                                                       FINA 330
      M(1,K+1)=AA                                                       FINA 340
      M(2,K+1)=MODE(LET)                                                FINA 350
```

```
36:          M(3,K+1)=AA                                  FINA 360
37:          M(2,K+2)=A                                   FINA 370
38:   10     RETURN                                       FINA 380
39:          END                                          FINA 390
```

```
      SUBROUTINE BLOQ(NELE,PATH,M,I,J,GATE,BLOCK)                       BLOQ  10
C     USAGE:                                                            BLOQ  20
C                                                                       BLOQ  30
C         CALL BLOQ(NELE,PATH,M,I,J,GATE,BLOCK)                         BLOQ  40
C                                                                       BLOQ  50
C     PARAMETERS:                                                       BLOQ  60
C                                                                       BLOQ  70
C         NELE   -   NO. OF STEPS                                       BLOQ  80
C         PATH   -   MINIMAL PATHS MATRIX                               BLOQ  90
C         M      -   PREREQUISITE DIAGRAM MATRIX                        BLOQ 100
C         I      -   ROW OF PATH MATRIX                                 BLOQ 110
C         J      -   COLUMN OF PATH MATRIX                              BLOQ 120
C         GATE   -   FLAG GATE ARRAY                                    BLOQ 130
C         BLOCK  -   MODULES ARRAY                                      BLOQ 140
C                                                                       BLOQ 150
C     PURPOSE:                                                          BLOQ 160
C                                                                       BLOQ 170
C         CLASSIFY AND TRANSFORM AN ELEMENT OF PATH MATRIX              BLOQ 180
C     REPRESENTING A MODULE (AND / OR)                                  BLOQ 190
C                                                                       BLOQ 200
C     SUBROUTINES USED:                                                 BLOQ 210
C                                                                       BLOQ 220
C         NONE                                                          BLOQ 230
C                                                                       BLOQ 240
      IMPLICIT INTEGER*4 (A-Z)                                          BLOQ 250
      DIMENSION PATH(100,50),MODE(60),M(6,76),GATE(60),YO(4),BLOCK(60)  BLOQ 260
      DATA MODE / ' 1',' 2',' 3',' 4',' 5',' 6',' 7',' 8',' 9','10',    BLOQ 270
     1'11','12','13','14','15','16','17','18','19','20',                 BLOQ 280
     2 '21','22','23','24','25','26','27','28','29','30',               BLOQ 290
     3'31','32','33','34','35','36','37','38','39','40',                BLOQ 300
     4'41','42','43','44','45','46','47','48','49','50',                BLOQ 310
     5'51','52','53','54','55','56','57','58','59','60' /               BLOQ 320
      DATA YO / 'OR','  ','AN','D ' /                                   BLOQ 330
      DATA BB,BA,AA,VB,B,A,HB,HH,H /                                    BLOQ 340
     1  '  ',' *','**','| ',' ','*','- ','--','-' /                      BLOQ 350
```

```
      K = 4*(J-1) + 1                                   BLOQ 360
      LET=PATH(J,I)                                     BLOQ 370
      M(1,K) = BB                                       BLOQ 380
      M(2,K) = BA                                       BLOQ 390
      M(3,K) = BB                                       BLOQ 400
      M(1,K+1)=AA                                       BLOQ 410
      M(2,K+1)=MODE(LET)                                BLOQ 420
      M(3,K+1)=AA                                       BLOQ 430
      M(4,K+1)=VB                                       BLOQ 440
      DO 40 L =1,NELE                                   BLOQ 450
      IF(BLOCK(L) .NE. LET) GO TO 40                    BLOQ 460
      PUERTA = GATE(L)                                  BLOQ 470
40    CONTINUE                                          BLOQ 480
      IF(PUERTA .EQ. 0 ) GO TO 1                        BLOQ 490
      M(5,K+1)=YO(PUERTA)                               BLOQ 500
      GO TO 2                                           BLOQ 510
1     M(5,K+1)=YO(3)                                    BLOQ 520
      M(5,K+2)=YO(4)                                    BLOQ 530
2     M(6,K+1)=VB                                       BLOQ 540
      M(1,K+2)=B                                        BLOQ 550
      M(2,K+2)=A                                        BLOQ 560
      M(3,K+2)=B                                        BLOQ 570
      RETURN                                            BLOQ 580
      END                                               BLOQ 590
```

```
      SUBROUTINE LINE(PATH,M,I,J)
C     USAGE:
C
C         CALL LINE(PATH,M,I,J)
C
C     PARAMETERS:
C
C        PATH   -   PATH MATRIX
C         M      -   PREREQUISITE DIAGRAM MATRIX
C         I      -   ROW OF PATH MATRIX
C         J      -   COLUMN OF PATH MATRIX
C
C     PURPOSE:
C
C         CLASSIFY AND TRANSFORM AN ELEMENT OF PATH MATRIX
C     REPRESENTING A SEGMENT OF LINE.
C
C     SUBROUTINES USED:
C
C         NONE
C
      IMPLICIT INTEGER*4 (A-Z)
      DIMENSION PATH(100,50),M(6,76)
      DATA HH,HB,VB,H / '--','- ','| ','-' /
      K = 4*(J-1) + 1
      M(4,K) = HH
      IF(PATH(J,I+1).EQ.0) GO TO 20
      IF(PATH(J+1,I).EQ.-99) GO TO 10
      M(4,K+1)=HB
      M(5,K+1)=VB
      M(6,K+1)=VB
      GO TO 30
10    M(5,K+1)=VB
      M(6,K+1)=VB
20    M(4,K+1)=HH
```

```
36:          M(4,K+2)=H                                  LINE 360
37:          M(4,K+3)=HH                                 LINE 370
38:   30     RETURN                                      LINE 380
39:          END                                         LINE 390
```

```
      SUBROUTINE LIN1(PATH,M,I,J)                                       LIN1  10
C     USAGE:                                                            LIN1  20
C                                                                       LIN1  30
C         CALL LIN1(PATH,M,I,J)                                         LIN1  40
C                                                                       LIN1  50
C     PARAMETERS:                                                       LIN1  60
C                                                                       LIN1  70
C        PATH   -   PATH MATRIX                                         LIN1  80
C         M      -   PREREQUISITE DIAGRAM MATRIX                       LIN1  90
C         I      -   ROW OF PATH MATRIX                                LIN1 100
C         J      -   COLUMN OF PATH MATRIX                             LIN1 110
C                                                                       LIN1 120
C     PURPOSE:                                                          LIN1 130
C                                                                       LIN1 140
C         CLASSIFY AND TRANSFORM AN ELEMENT OF PATH MATRIX              LIN1 150
C     REPRESENTING A SEGMENT OF INTERMEDIATE LINE IN THE                LIN1 160
C     PREREQUISITE DIAGRAM                                              LIN1 170
C                                                                       LIN1 180
C     SUBROUTINES USED:                                                 LIN1 190
C                                                                       LIN1 200
C         NONE                                                          LIN1 210
C                                                                       LIN1 220
      IMPLICIT INTEGER*4 (A-Z)                                          LIN1 230
      DIMENSION PATH(100,50),M(6,76)                                    LIN1 240
      DATA HH,HB,VB,H / '--','- ','| ','-' /                             LIN1 250
      K = 4*(J-1) + 1                                                   LIN1 260
      M(1,K+1)=VB                                                       LIN1 270
      M(2,K+1)=VB                                                       LIN1 280
      M(3,K+1)=VB                                                       LIN1 290
      M(4,K+1)=VB                                                       LIN1 300
      IF(PATH(J,I+1).EQ.0) GO TO 20                                     LIN1 310
      IF(PATH(J+1,I).EQ.-99) GO TO 10                                   LIN1 320
      M(5,K+1)=VB                                                       LIN1 330
      M(6,K+1)=VB                                                       LIN1 340
      GO TO 30                                                          LIN1 350
```

```
36:  10     M(4,K+1)=HH                    LIN1 360
37:         M(5,K+1)=VB                    LIN1 370
38:         M(6,K+1)=VB                    LIN1 380
39:  20     M(4,K+2)=H                     LIN1 390
40:         M(4,K+3)=HH                    LIN1 400
41:  30     RETURN                         LIN1 410
42:         END                            LIN1 420
```