



Fault-diagnosis by inserting the minimum number of test points in system graphs
by Mahesh Patel

A thesis submitted in partial fulfillment of the requirements for the degree of DOCTOR OF
PHILOSOPHY in Electrical Engineering
Montana State University
© Copyright by Mahesh Patel (1974)

Abstract:

Classical methods of fault-diagnosis in combinational and sequential logic are effective for small circuits, but are too exhaustive and do not provide any systematic way for analyzing entire computer systems. Large scale integration and increased complexity of present-day computer systems have created a need for new approaches for understanding, analyzing, and visualizing malfunctions in these Systems. Graph theory, which is extensively used in other fields, sheds a new light on the above problem.

As fault diagnosis deals both with structural and behavioral characteristics of systems, we can study them separately and superimpose behavioral characteristic on structural characteristic to obtain diagnostic tests. The structural characteristics can be studied using graph theory. Ramamoorthy and Mayeda have developed structural models of computer systems and/or logic circuits and have shown techniques to find structural resolution. For the improvement of given system structures, they have suggested insertion of what are called test points and/or blocking gates in the system. The object of this thesis has been the development of an algorithm for inserting test points in system structure to obtain maximum improvement. The algorithm is initiated using only basic information (edge connections) and generates the locations of test points. Test points obtained are the minimum for a given system graph. The algorithm is computationally efficient and applicable for the analysis of large system graphs. The thesis also includes theorems on lower bounds on the number of test points required for a system graph.

FAULT-DIAGNOSIS BY INSERTING THE
MINIMUM NUMBER OF TEST POINTS IN
SYSTEM GRAPHS

by

MAHESH PATEL

A thesis submitted in partial fulfillment of the
requirements for the degree

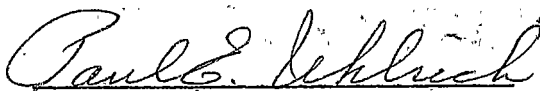
of

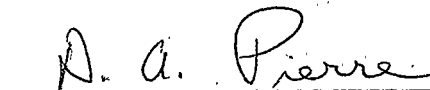
DOCTOR OF PHILOSOPHY

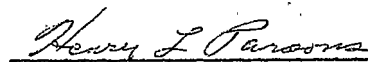
in

Electrical Engineering

Approved:


Head, Major Department


Chairman, Examining Committee


Graduate Dean

MONTANA STATE UNIVERSITY
Bozeman, Montana

August, 1974

ACKNOWLEDGMENTS

I am indebted to Dr. Donald A. Pierre who served as my advisor, for his active participation and valuable guidance during the course of this research.

I give special thanks to Dr. Craig K. Rushforth, my former advisor, for his valuable insights during the initial development of this work. I would also like to thank Dr. Donald A. Rudberg and Dr. Roy M. Johnson for their comments and helpful suggestions. I acknowledge Dr. Paul E. Uhlrich and the Department of Electrical Engineering for providing the financial support and the use of the computer facilities at Montana State University. I express my appreciation to Mrs. Jean Julian for her excellent technical typing.

Finally, I would like to thank my parents, whose guidance made this thesis possible, and my wife, Varsha, for diligently drawing figures and for her encouragement throughout the development of the thesis. To my mother, Vimala, I dedicate this thesis.

August, 1974

Mahesh Patel

TABLE OF CONTENTS

	<u>Page</u>
VITA	ii
ACKNOWLEDGMENTS	iii
TABLE OF CONTENTS	iv
LIST OF FIGURES	vii
ABSTRACT	xi
Chapter	
I. INTRODUCTION	1
1.1 Historical Review of Fault-Tolerant Computing	1
1.1.1 Redundancy techniques	2
1.1.2 Fault-diagnosis techniques	3
1.2 New Graph-Theory Approach to Diagnostics	7
1.2.1 Necessity for a new approach	7
1.2.2 Graph representation of systems	12
1.2.3 Reduction of complex system graphs	17
1.2.4 SEC graphs	24
1.3 Statement of the Problem	28

	<u>Page</u>
II. PRELIMINARY IDEAS IN SYSTEM DIAGNOSIS USING GRAPH THEORY	33
2.1 Assumptions and Definitions	33
2.2 Distinguishability	37
2.3 Test Points	42
2.3.1 Introduction	42
2.3.2 Theorems on necessary and/or sufficient conditions	46
2.3.3 Ordering a SEC graph	59
2.4 Blocking Gates	63
III. LOWER BOUND ON TEST POINTS FOR A SEC GRAPH	74
3.1 Introduction and Preliminary Theorems	74
3.2 Collections of Generators	80
3.3 D-partition and 1-distinguishability	83
3.4 Lower Bound on Test Points for 1-distinguishability of a SEC Graph	86
IV. AN ALGORITHM FOR THE MINIMUM NUMBER OF TEST POINTS	101
4.1 Introduction	101
4.2 Necessity for an Algorithm	101
4.3 Matrix Representation of a SEC Graph	106

	<u>Page</u>
4.4 Development of an Algorithm	110
4.4.1 Directed cut-sets covered by an edge	110
4.4.2 A set V of non-separating pairs of vertices	117
4.4.3 Final covering matrix Z	124
4.5 The Algorithm for Inserting the Minimum Number of Test Points on a SEC Graph for 1-distinguishability	130
V. SUMMARY AND SUGGESTED FUTURE RESEARCH . . .	147
5.1 Introduction	147
5.2 Summary and Conclusion	147
5.3 Suggestions for Future Research	150
 APPENDICES	
A - Summary of Notation for Program Listing	152
B - Program Listings for Inserting the Minimum Number of Test Points	160
C - List of Definitions	188
REFERENCES	191

LIST OF FIGURES

<u>Figure</u>		<u>Page</u>
1.1(a)	The Königsberg bridges	9
1.1(b)	A graph representation of the Königsberg bridges in Figure 1.1(a)	9
1.2(a)	A serial arithmetic unit of the STRETCH computer	14
1.2(b)	A structural representation of the serial arithmetic unit of the STRETCH computer in Figure 1.2(a)	15
1.3	A structural representation of the flow diagram 'ASCEND' in Figure 4.10 . .	16
1.4(a)	Logic circuit	18
1.4(b)	A structural representation of the logic circuit in Figure 1.4(a)	19
1.5	Sequence of operations in design processing	20
1.6	A separable graph	22
1.7(a)	A system graph with M.S.C. and link subgraphs	25
1.7(b)	A SEC graph of the system graph in Figure 1.7(a)	26
2.1	Directed graph for D-partition	39
2.2	Directed graph not 1-distinguishable by the test point method	43
2.3(a)	A SEC graph for Example 2.2	49

<u>Figure</u>		<u>Page</u>
2.3(b)	Subgraphs G_1 and G_2 of the SEC graph in Figure 2.3(a)	49
2.3(c)	Subgraphs G'_1 , G'_2 , G'_3 of the SEC graph in Figure 2.3(a)	49
2.4	A SEC graph and directed cut-sets covered by $E = \{a, d, e\}$	53
2.5	A SEC graph which is not 1-distinguishable by an edge-set $E = \{a, b\}$ (directed cut-sets covered by E contain all edges of the SEC graph)	56
2.6	A SEC graph and six linearly independent directed cut-sets	62
2.7(a)	A logic circuit and design of blocking gates on edges e_4 and e_5	65
2.7(b)	A structural representation of the logic circuit in Figure 2.7(a)	65
2.8	A SEC graph with blocking gates on edges e_1 , e_2 , e_3 and e_4	68
2.9	A SEC graph with blocking gates on edges e_1 , e_2 and e_3	68
2.10	A 1-distinguishable SEC graph under an edge set $E = \{e_1, e_2, e_3\}$	71
2.11	A 1-(1,7)-distinguishable SEC graph under an edge set $E = \{e_1, e_2, e_3\}$	71
3.1	A SEC graph with two test points on edges e_ℓ and e_m	77
3.2	A SEC graph for Example 3.1	79
3.3	A SEC graph with directed cut-sets covered by $E = \{e_1, e_2, e_3\}$	82

<u>Figure</u>		<u>Page</u>
3.4	A SEC graph with directed cut-sets covered by $E = \{e_1, e_2, e_3\}$	87
3.5	A 1-distinguishable SEC graph of 7 vertices with 2 test points	98
3.6	A 1-distinguishable SEC graph of 15 vertices with 3 test points	100
4.1	A SEC graph with directed cut-sets covered by an edge e	114
4.2	A directed path from v_p to v_q with directed cut-set $\mathcal{G}(\Omega_k^p \times \Omega_k^q)$	119
4.3	A SEC graph with no directed paths from 2 to 4 (or 4 to 2), 2 to 7 and 3 to 7	121
4.4(a)	A SEC graph with the minimum number of directed cut-sets that separate every pair of vertices	126
4.4(b)	A SEC graph with the minimum number of edges $E = \{e_1, e_6\}$; directed cut-sets covered by E separate every pair of vertices	126
4.5(a)	Flow diagram of 'MAIN PROGRAM'	138
4.5(b)	Flow diagram of 'MAIN PROGRAM'	139
4.5(c)	Flow diagram of 'MAIN PROGRAM'	140
4.6	Flow diagram of subroutine 'REACHMAT'	141
4.7(a)	Flow diagram of subroutine 'CUTSET'	142
4.7(b)	Flow diagram of subroutine 'CUTSET'	143
4.8	Flow diagram of subroutine 'ENTZMAT'	144

<u>Figure</u>		<u>Page</u>
4.9	Flow diagram of subroutine 'PARVERTX' . .	145
4.10	Flow diagram of subroutine 'ASCEND' . . .	145
4.11	Flow diagram of subroutine 'ASCENDPAR'	146
B.1	A SEC graph used in the sample program . .	181

ABSTRACT

Classical methods of fault-diagnosis in combinational and sequential logic are effective for small circuits, but are too exhaustive and do not provide any systematic way for analyzing entire computer systems. Large scale integration and increased complexity of present-day computer systems have created a need for new approaches for understanding, analyzing, and visualizing malfunctions in these systems. Graph theory, which is extensively used in other fields, sheds a new light on the above problem. As fault diagnosis deals both with structural and behavioral characteristics of systems, we can study them separately and superimpose behavioral characteristic on structural characteristic to obtain diagnostic tests. The structural characteristics can be studied using graph theory. Ramamoorthy and Mayeda have developed structural models of computer systems and/or logic circuits and have shown techniques to find structural resolution. For the improvement of given system structures, they have suggested insertion of what are called test points and/or blocking gates in the system. The object of this thesis has been the development of an algorithm for inserting test points in system structure to obtain maximum improvement. The algorithm is initiated using only basic information (edge connections) and generates the locations of test points. Test points obtained are the minimum for a given system graph. The algorithm is computationally efficient and applicable for the analysis of large system graphs. The thesis also includes theorems on lower bounds on the number of test points required for a system graph.

Chapter I

INTRODUCTION

1.1 Historical Review of Fault-Tolerant Computing

Fault-tolerant computing has emerged as an important discipline in the area of computer architecture [39]. It has also become an important design parameter of fourth generation computing systems. We define fault-tolerant computing as the ability to sustain essential operation even under multiple hardware failures and software errors. Clearly it is a challenge to achieve even partially the goal implied by the definition. Basically, the technology of fault-tolerant computing encompasses theory and techniques of fault and error detection and correction, modeling, analysis, synthesis, and the architecture of fault-tolerant systems and their evaluation. The importance of this subject area can be viewed in a different way. In the past, computers were used largely in an off-line, batch-processing mode; the consequences of undetected hardware malfunctions were relatively minor. We now see increasing use of computers in on-line, real-time applications such as the control of chemical process units, nuclear reactors, and military command and control

equipment. Incorrect computer operation in any of these applications can be potentially disastrous. At the same time, the use of multiprogramming, multiprocessing, parallel processing, microprogramming and the emergence of LSI (Large Scale Integrated circuits) have added a great deal of complexity in computer systems and have made it more and more difficult to ensure correct machine operation.

There are two fundamentally different ways of coping with faults.

1.1.1 Redundancy techniques [34,57] Here the effects of the fault can be masked because a fault-corrective property is inherent. These techniques may be subdivided into two classes:

(A) Fault-masking techniques: In this, effects of any fault are masked by additional circuits. These additional circuits are part of the system, and no switching is involved. Thus, error correction is instantaneous. N-modular redundancy voting schemes [58] are the best examples of this type. These schemes are used for applications such as spacecraft guidance, where repair is not

possible, but the system is required to operate with high reliability for a specified time period.

(B) Self-checking systems: A self-checking system should be able to detect faults within itself, identify faulty subsystems, and switch them out of the system. This should be done in a manner so that the system can continue to operate with the remaining units but possibly with reduced capabilities. Real-time systems, such as the electronic switching systems used for telephone switching, belong to this category, where interruptions in the operation are intolerable but repair is possible. With the exception of TARP (Testing and Repair Processor). The STAR (Self-Testing and Repair) Computer [2] was designed using this idea. This is an experimental computer which was designed and constructed primarily for research and evaluation of self-repair techniques. Its performance is suitable for guidance and control of unmanned interplanetary spacecraft.

1.1.2 Fault-diagnosis techniques [12,16] The process of applying tests and determining whether the computer is fault free or not is generally known as fault detection. Fault location (isolation) is the process of identifying the failures within the smallest possible set of

components. Diagnosis is the composite of both the above processes. Fault diagnosis implies no fault corrections as there are in redundancy techniques.

In one sense, fault masking and fault diagnosis are incompatible, since the effect of a masked fault can not be observed or diagnosed--at least until a further fault develops in the masking mechanism. The subject area of this work is the diagnosis problem.

We shall briefly summarize current efforts in the area of fault diagnosis. They may be classified broadly in the following way:

(A) The classical methods of diagnosis: The component elements are classified into combinational and sequential entities. Methods of test generation for combinational circuits were first started by Eldred [14] and Armstrong [1]. Their work was based on the 'sensitized paths' method. The next significant step in this area was taken by Roth [48]. He developed the D-algorithm which was based on the calculus of D-cubes. It was applicable only for single failures and small-size non-redundant networks. Another method used was that of 'Boolean difference' [50].

In elements with memory, the derivation of fault-detection and diagnosis tests is greatly complicated by the fact that the signals may be neither observable nor controllable under fault conditions. Two different approaches for sequential circuits are considered. In the first, which is called the 'circuit testing approach' [16,35], the circuit realization and the class of faults that can occur are assumed to be known. In the second approach, the 'machine identification approach' [22], the flow table of the machine to be tested is given, but circuit realization is unknown.

A great deal of research is currently being done on classical methods of fault diagnosis. The problem is well understood for non-redundant, combinational logic circuits having single, permanent type faults. For sequential logic, the main problem that is left is that of determining optimum lengths of test sequences, keeping in mind simplicity of test derivation.

No matter what method we choose and what type of circuits—combinational or sequential—we select, the number of tests required make it prohibitive to analyze or diagnose a very large digital circuit or computer system as a

whole. So it must be understood that all the works on classical methods are limited to a small part of present-day digital systems.

(B) The black-box approach: This deals with input-output relationships of the system and develops efficient tests to verify a set of functional specifications. It ignores the machine structure so it is not useful for fault-location.

(C) Design simulation: The design is simulated under known component failure modes, and the test results (test dictionary) are catalogued for future reference. The sequential analyzer [12,51,52,53] was developed by the late Professor S. Seshu. It is a set of computer programs capable of simultaneously simulating a normal digital circuit and the effect of several different faults on it and also deriving fault-detection or diagnostic test sets for the circuit. It is applicable to asynchronous sequential circuits represented by the Huffman model [23] and is based on the assumption that it is possible to reset the feedback lines to a known initial state, even under a failure condition.

(D) Self-diagnosible computer [15,28,36]: In this method, computer organization is set up in such a way that

fault-diagnosis and maintenance become easier. One such approach [36] decomposes the system into n units where a unit is a well-definable portion of the system which cannot be further decomposed for the purpose of diagnosis. By means of a given arrangement of testing links (connection assignments), each unit of the system tests a subset of units. A self-diagnosable computer of Forbes et al. [15] was based on somewhat the same principle with the following design objectives:

1. Maximum capability for self-diagnosis
2. Minimum mean-time to diagnosis
3. Minimum additional hardware to make it self-diagnosable.
4. Minimum hard-core (that portion of logic which must function correctly).

The main drawback in this method of diagnosis is that it imposes design constraints that could impede superior performance of the machine.

1.2 New Graph Theory Approach to Diagnostics

1.2.1 Necessity for a new approach So far we have discussed current trends in fault-tolerant computing, involving both diagnosis and correction. In a diagnostic

sense, it becomes clear at this stage that there is not a single method which provides simple but effective means of understanding, visualizing, and analyzing the problem. On the other hand, the need for such a method continually increases, as we add more and more components and link them with already complex machine structures. A unified approach based on graph theory seems to provide new insight into the problem since it is applicable at various levels of detail. The reason is simple. It is very easy to present system structure by points called 'vertices' and by line segments called 'edges' which connect these vertices so that the vertices and edges indicate components and relationships between these components. Such a diagram is a linear graph. A linear graph often is known by other names depending on the kind of physical system we are dealing with; it may be called a network, a net, a graph, a diagram, a structure and so on.

The first paper on graphs was written by the famous Swiss mathematician Leonard Euler (1707-1783). He solved the 'Königsberg Bridges' problem by introducing basic concepts of linear graphs. The problem is that of crossing seven bridges on two islands as shown in Figure 1.1(a)

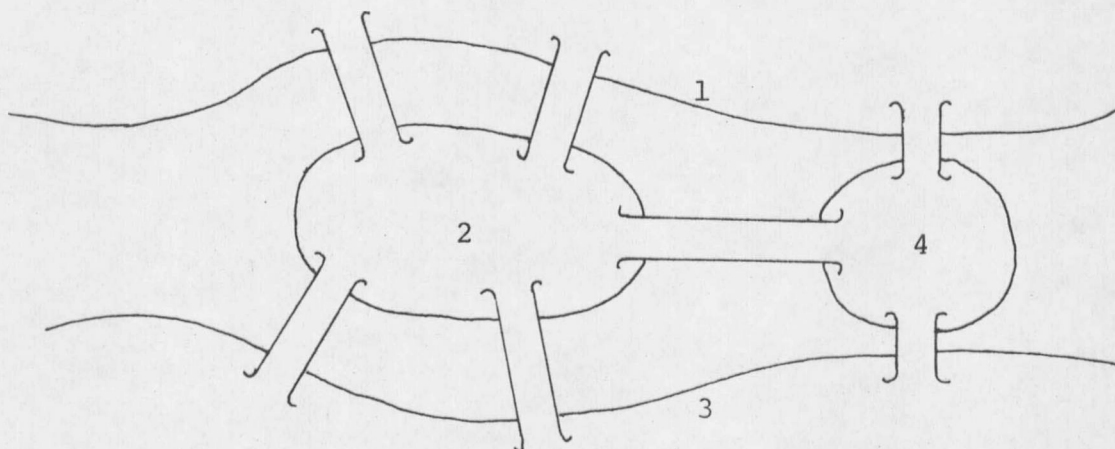


Figure 1.1(a) The Königsberg bridges.

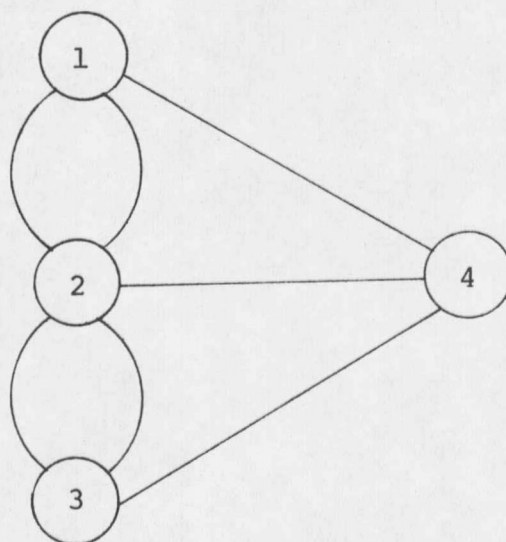


Figure 1.1(b) A graph representation of the Königsberg bridges in Figure 1.1(a).

without retracing. He prepared a graph model as shown in Figure 1.1(b) and proved that the problem has no solution. In 1847, Kirchhoff employed linear graph theory for the analysis of electrical networks to find driving point and transfer admittances. Cayley [11], with pioneering work in chemistry, was the first to count the number of isomers by enumerating the trees of a graph. The field of graph theory today encompasses a very wide field of applications; ranging from physical science [19] and engineering [54] to psychology, social science [20] and economics [3]. In social science, it is used to study organizational structure, family relationships of groups of people, and genetic problems [29]. In operations research it is used in PERT network analysis in linear, dynamic and integer programming, in traffic flow studies and in the well-known traveling salesman problem [29]. In electrical engineering, it is used to study switching theory, signal flow graphs, matrix inversion, topological analysis of passive and active electrical networks, and maximum flow in communication networks by weighted linear graphs [13,30,54]. In logic design, graph theory is used for partitioning, routing, and wire layout on integrated circuit and printed

circuit boards [7,25]. In computer science, it is used for program segmentation, detection of programming errors, critical path scheduling, and information retrieval. Tree structures are used to represent prefix, postfix, and infix formulas [6].

Looking back to our problem in system diagnosis, the introduction of graph theory can give very valuable insight. It can simplify diagnostic problems that seem complicated because of an inherent structural complexity of a system. Graph theory may augment the ability of the maintenance engineer to cope with unforeseen and unexpected problems. The advantages of using graph models for diagnostic analysis are many [41]. The principal ones are as follows:

1. Graph representation provides a uniform way of testing through a hierarchy of levels (a detailed explanation is given later on).
2. It helps to segment a large system into smaller subsystems for purposes of reducing testing time [42].
3. It provides a list of desirable sites for locating test points [30,31,32].
4. Such system graphs can be represented by connectivity matrices whose elements take binary values and are easily analyzed on a computer.

1.2.2 Graph representation of systems We look at a system from two distinct levels; the structural level and the behavioral level. For example, a sequential machine can be specified purely from the behavioral aspects like input-output relationships. Once the machine is designed, its structure (interconnection between components and flow of information) comes into being. The composite machine can then be looked upon as the superposition of behavioral characteristics of the components on its structural form. This type of separation provides simplicity and at the same time sheds a new light on the problems of diagnosis and maintenance.

Our first aim is the structural representation of any complex system. The following points must be noted while selecting proper structure:

1. It must mask out those details not pertinent to the problem at hand.
2. It must be simple enough to provide insight wherever and whenever needed.
3. It must be uniform. That means the same type of representation must be applicable to sets and subsets of the system.

It should be understood that the structural representation and the associated theories that we are going to

develop are applicable to pipeline flow, computer programming, control systems, computer systems as well as small logic circuits, or, in general, to any system whose structural representation is possible. But in this work, we are mainly concentrating on computer systems.

In Figure 1.2(a), the serial arithmetic unit of the STRETCH computer is shown [5,8]. The corresponding structural representation is in Figure 1.2(b). Similarly, Figure 1.3 is a structural representation of 'ASCEND' (For a flow diagram of 'ASCEND', see Figure 4.10).

The structure of the system is represented by interconnections of its components. The components can be discrete logical elements, like various types of gates and flip-flops, or functional units like counters, adders, etc. In most practical cases, components may be the smallest replacable modules like integrated circuit boards or flat-packs. These elements are represented by vertices of a graph. So, in general, a vertex may represent a combinational or a sequential logic unit, and, in some cases, a functioning unit of a computer or the entire computer if we consider the case of computer networks [5,46]. A directed edge in a graph represents a unidirectional

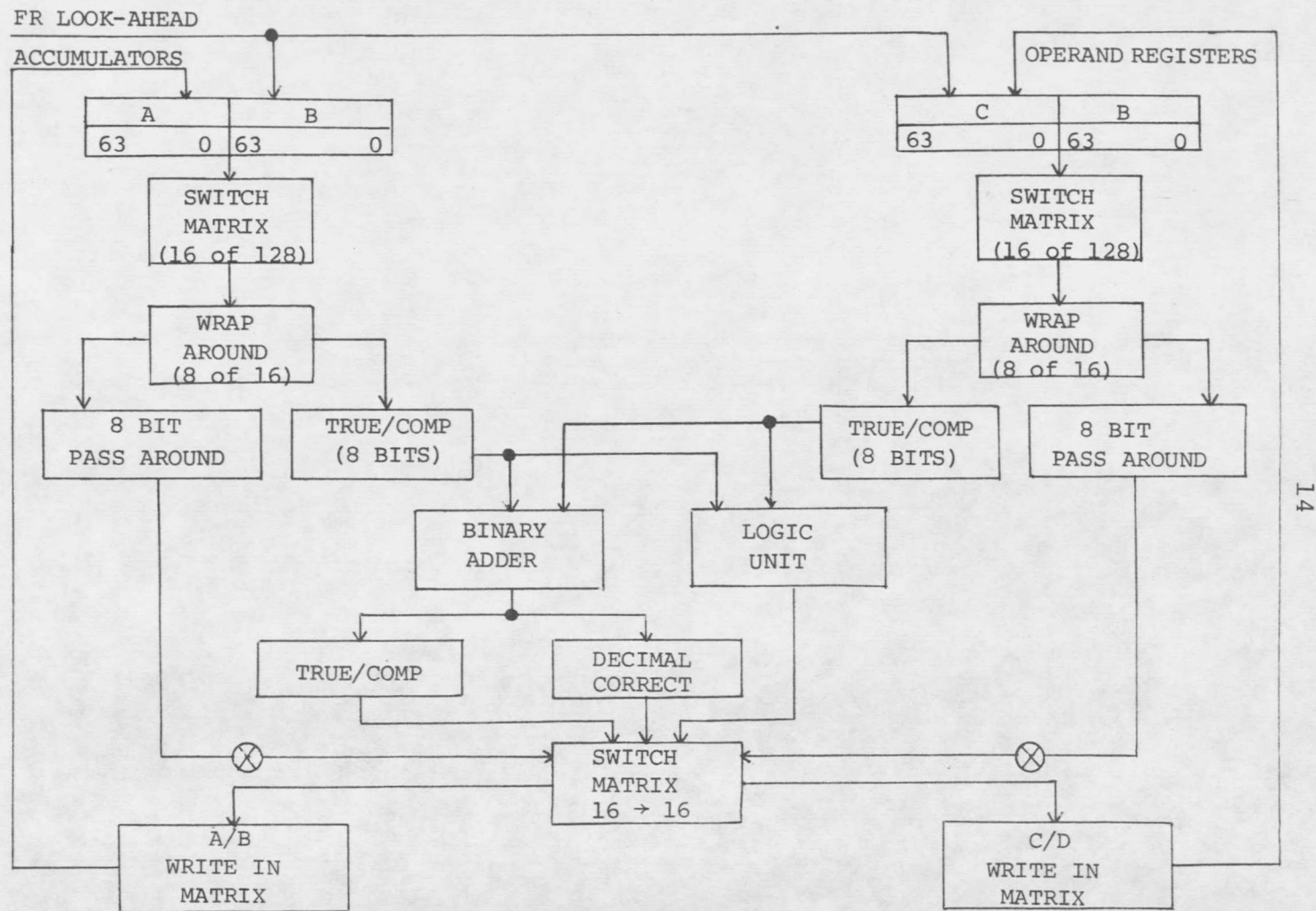
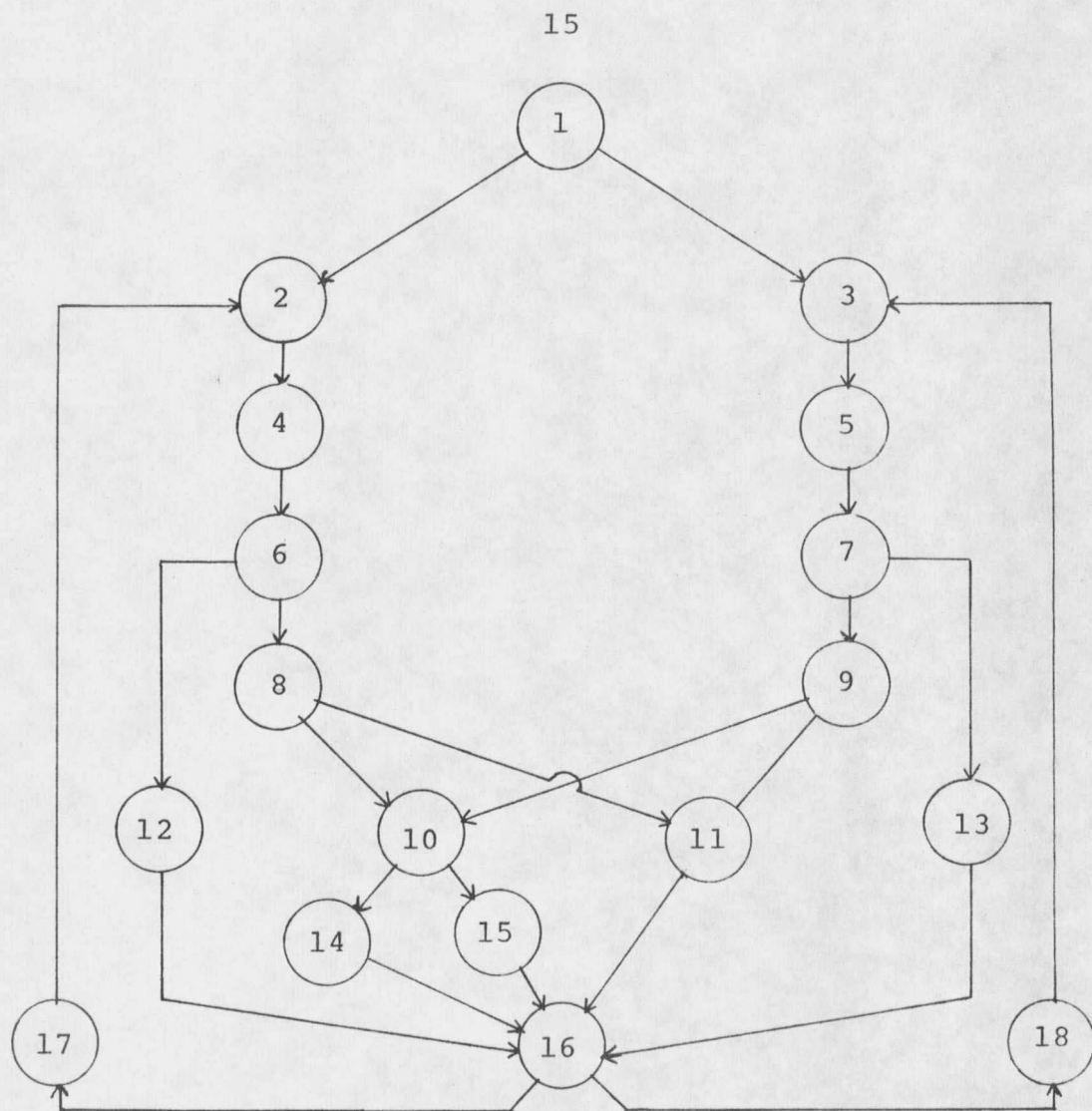


Figure 1.2(a) A serial arithmetic unit of the STRETCH computer.



- | | | | |
|-----|---------------------------|-------|-----------------------|
| 1 | Dummy vertex | 11 | Logic unit |
| 2 | Accumulators | 12,13 | 8 bits pass around |
| 3 | Operand registers | 14 | True/comp |
| 4,5 | Switch matrix (16 of 128) | 15 | Decimal correct |
| 6,7 | Wrap around (8 of 16) | 16 | Switch matrix 16 → 16 |
| 8,9 | True/comp (8 bits) | 17 | A/B write in matrix |
| 10 | Binary adder | 18 | C/D write in matrix |

Figure 1.2(b) A structural representation of the serial arithmetic unit of the STRETCH computer in Figure 1.2(a).

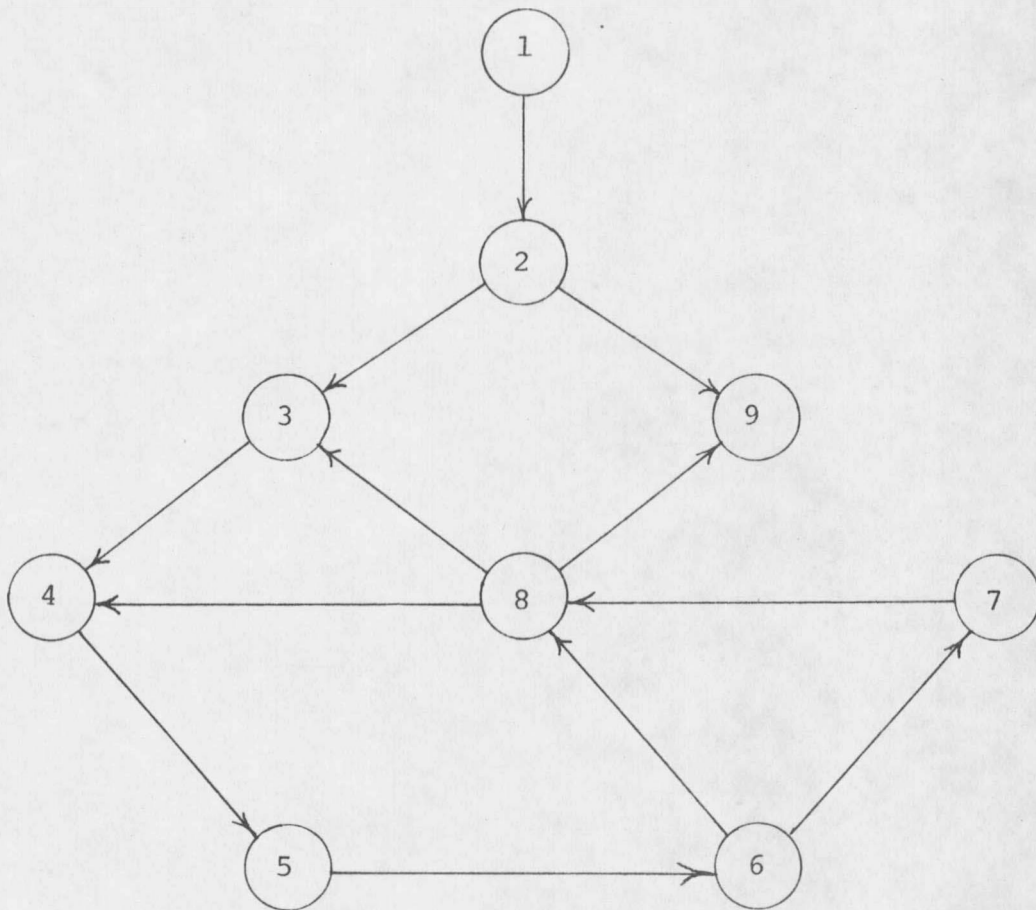


Figure 1.3 A structural representation of the flow diagram 'ASCEND' in Figure 4.10.

line of signal propagation. All graphs in the remainder of this work are directed. In Figure 1.4(a), a logic circuit is shown, and the corresponding graph with vertices and edges is shown in Figure 1.4(b). It should be noted that vertex 2 and vertex 3 contain sequential circuits.

A proposed sequence of operations [40] in design processing is shown in Figure 1.5. The method outlined in Figure 1.5 can be used to derive diagnostic procedures and also helps in component assignment in modules and sub-modules.

1.2.3 Reduction of complex system graphs If only primary (externally controllable) inputs and observable (externally controllable) outputs are used for diagnosis, the total number of test sequences will be very large. The size of the test dictionary will increase with the size of the structure and, in particular, with the number of feedback paths and memory elements. Thus, the breaking up of a large system into small subsystems is very important to improve resolution as well as to reduce the average time for test. Segmenting or partitioning is also important from another viewpoint. Usually for deriving fault detection and location procedures, we assume a single fault,

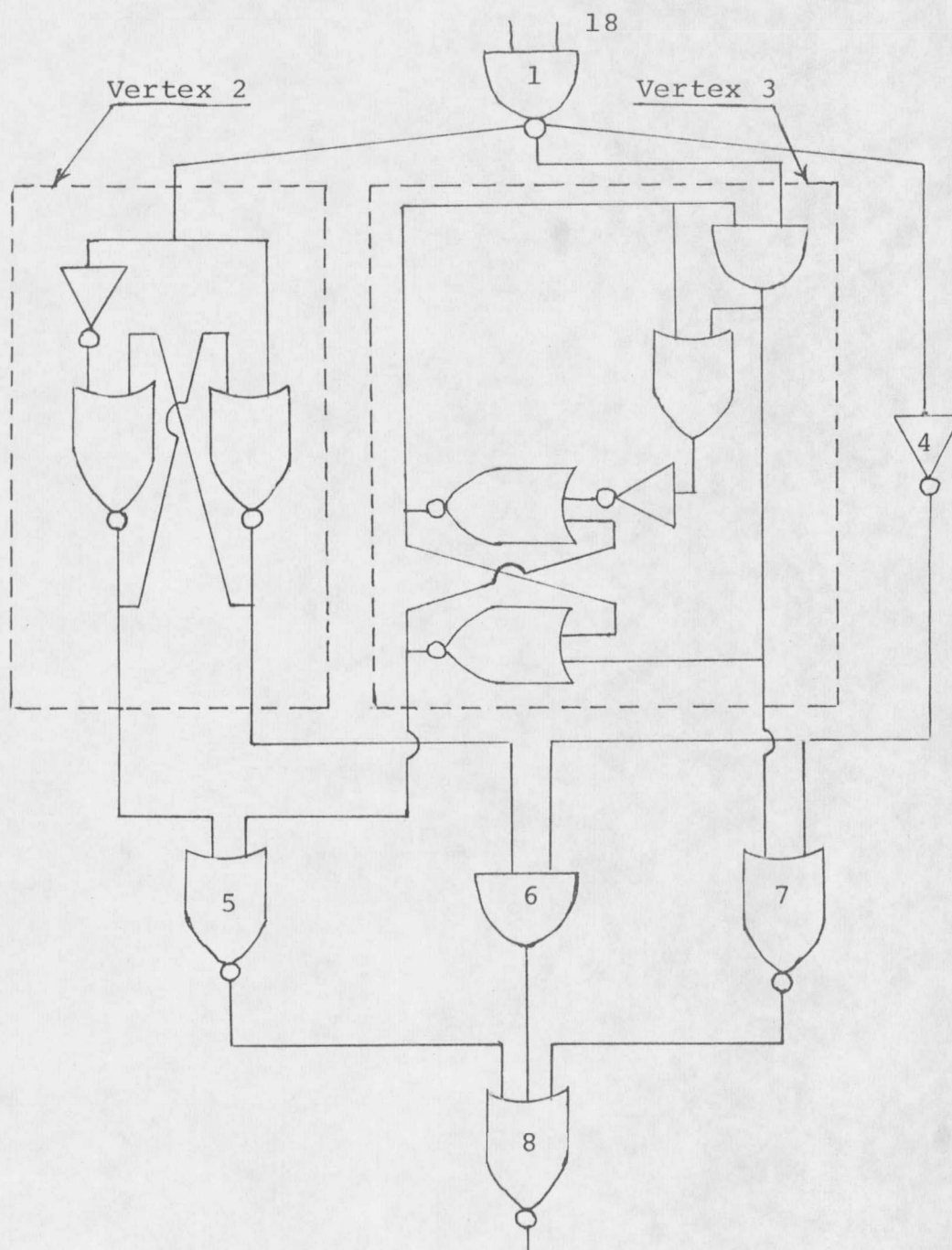


Figure 1.4(a) Logic circuit.

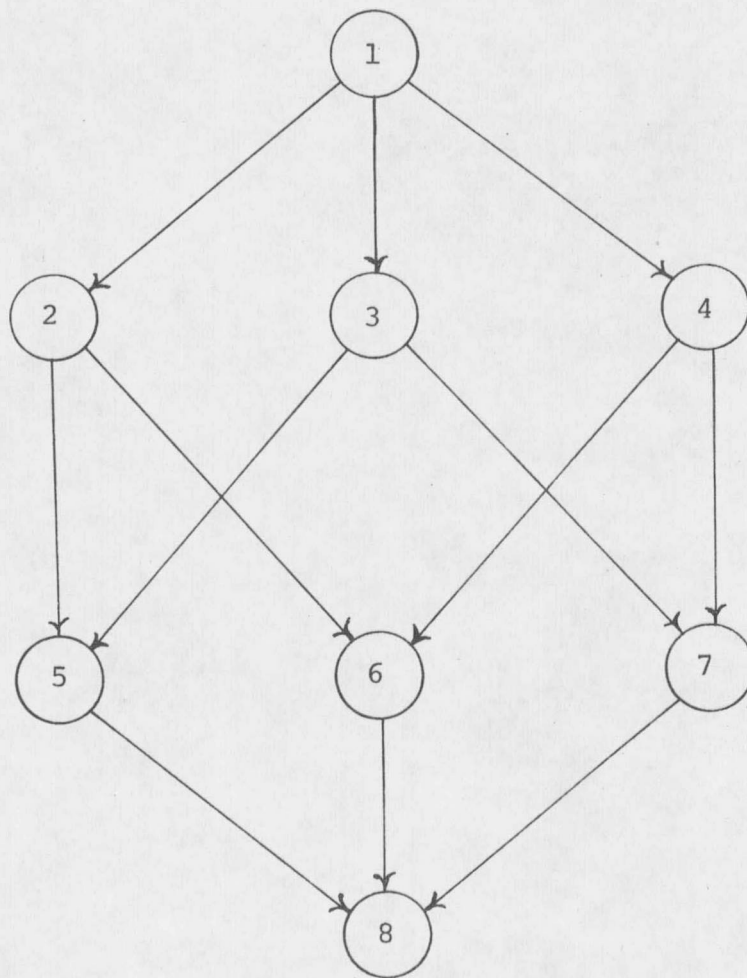


Figure 1.4(b) A structural representation of the logic circuit of Figure 1.4(a).

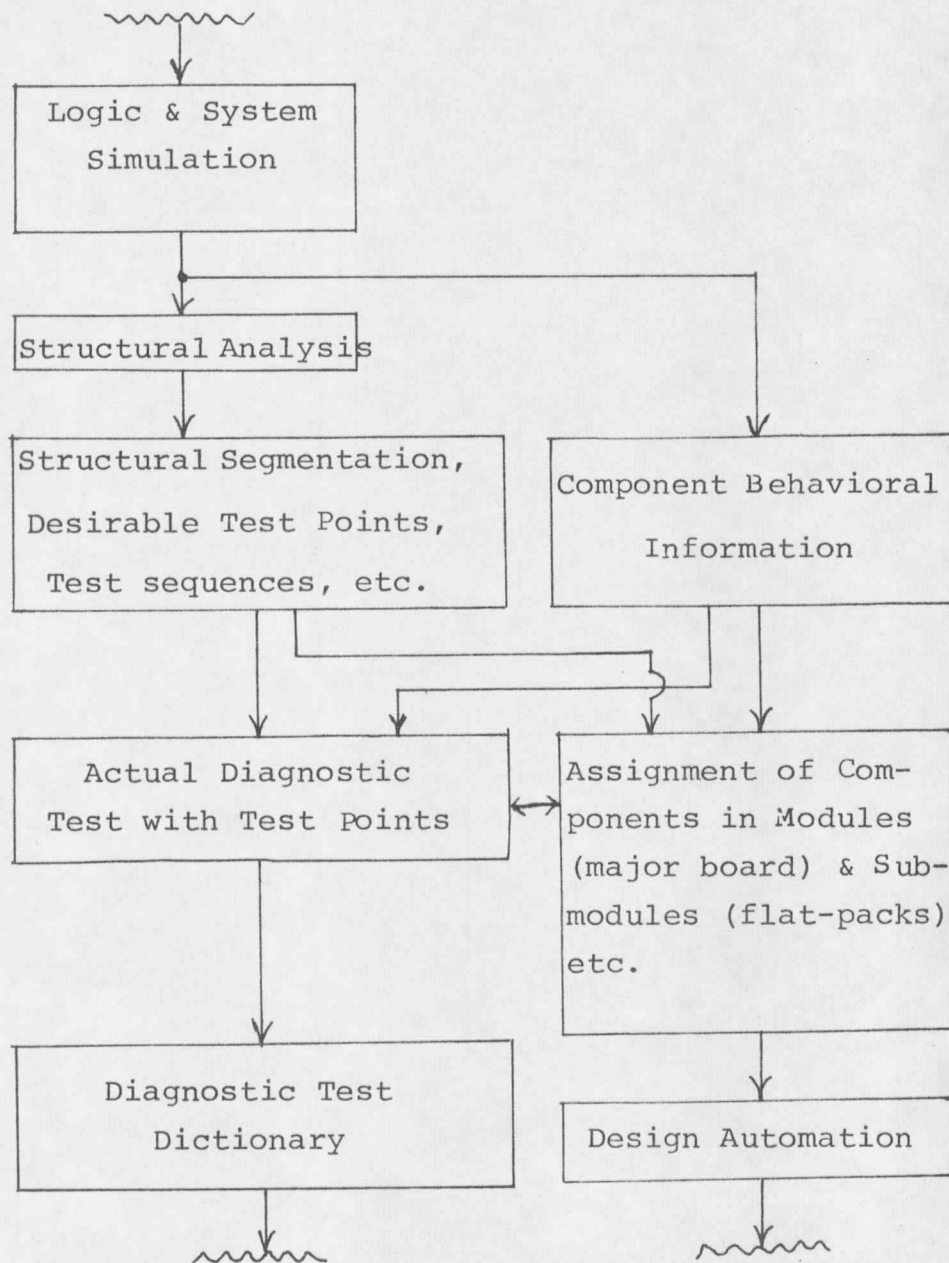


Figure 1.5 Sequence of operations in design processing.

which may not be true. Multiple faults may occur. Segmenting, at least during the diagnostic phase, tends to divide the system into mutually non-interacting subsystems and helps to diminish the effects of multiple faults. A third advantage of segmentation is that it helps in the component assignment problem in the design automation process.

Before discussing the method of structural segmentation, we define the phrases, separable graph and maximal strongly connected graph.

Definition 1.1 Essential vertex and separable graph:

Consider a graph G whose entry vertex is i and exit vertex is j . If there exists a vertex p such that every directed path from i to j passes through p , p is called an essential vertex or cut vertex; such a linear graph is called separable graph. The graph in Figure 1.6 is a separable graph.

Definition 1.2(a) Connected graph: A graph is said to be connected if there exists no partitioning of the vertices into two or more subsets such that there exists no edge (or edges) between the subsets.

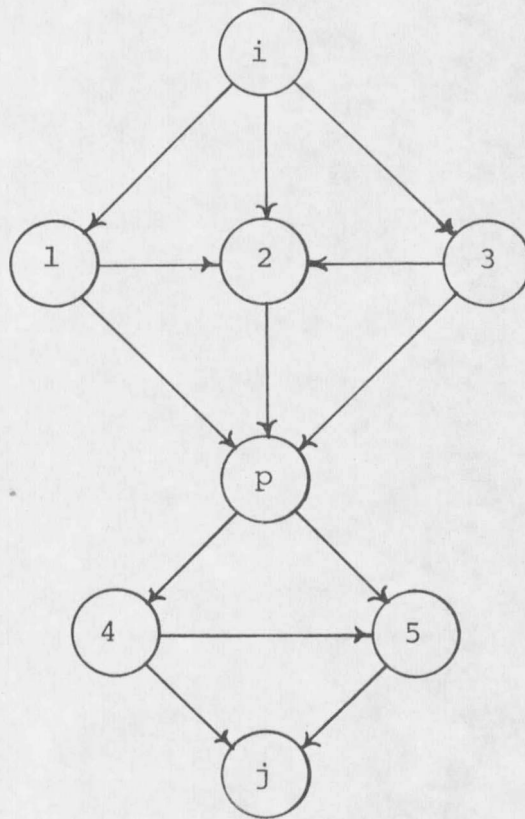


Figure 1.6 A separable graph.

Definition 1.2(b) Strongly connected graph: A directed graph is said to be strongly connected if, for every pair of distinct vertices i and j , there exists a directed path from i to j as well as from j to i .

Definition 1.3 Subgraph of a graph: A subgraph of a given graph is defined as consisting of a subset of vertices with all edges between these vertices retained.

Definition 1.4 Maximal strongly connected (M.S.C.) subgraph: An M.S.C. subgraph is a strongly connected subgraph that includes all possible vertices which are strongly connected with each other. All M.S.C. subgraphs are mutually disjoint.

Definition 1.5 Link subgraph: A link subgraph is one that contains no strongly connected subgraphs or unconnected subgraphs in it.

Algorithms [43] are available to find all M.S.C. subgraphs and link subgraphs of a graph using the connectivity matrix (Definition 4.1). M.S.C. or link subgraphs are usually replaced by single vertices, possibly with multiple inputs or outputs. The system graph after such

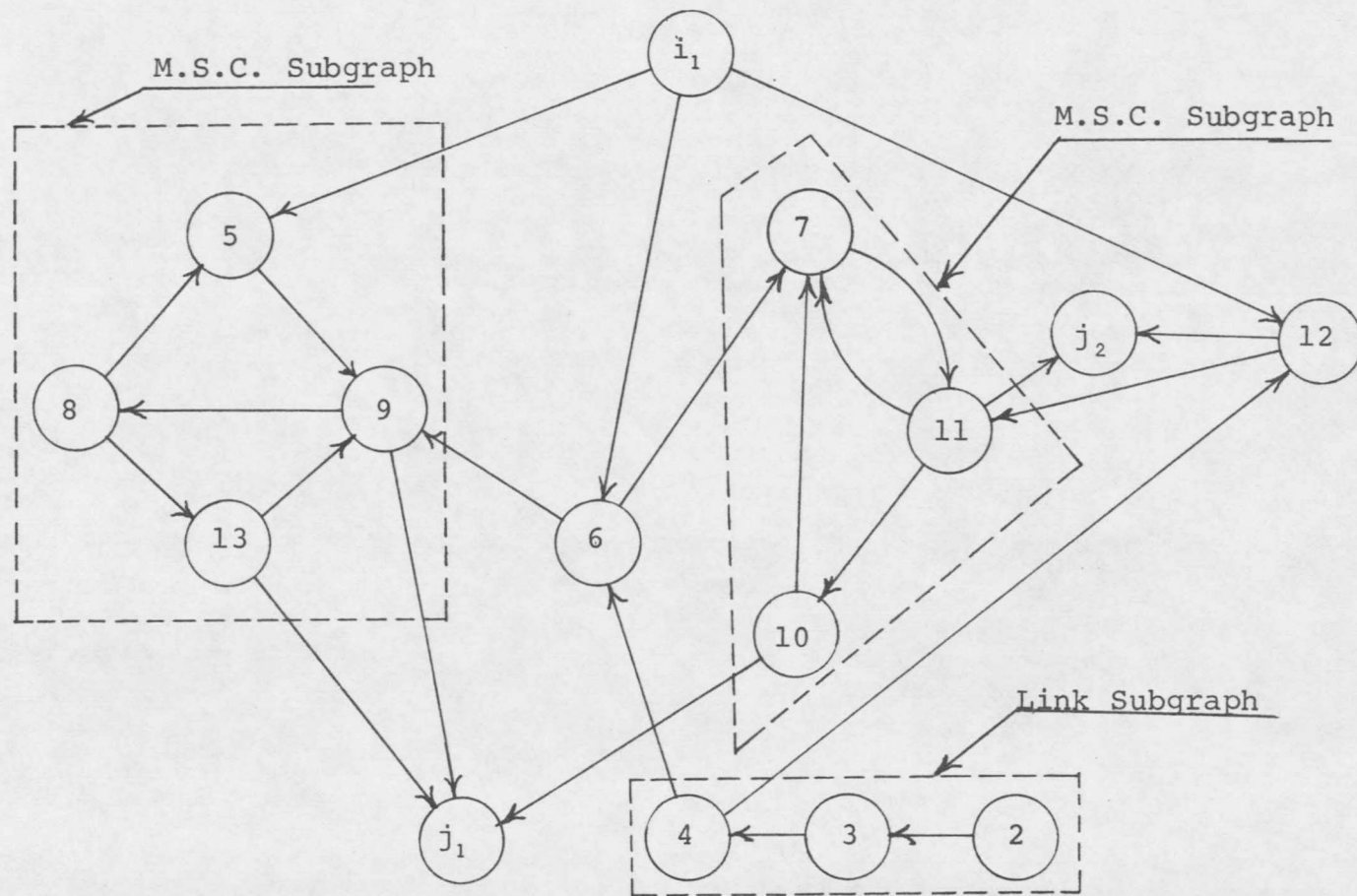
substitutions generally contains fewer vertices and feedback paths.

Definition 1.6 Entry vertex: A vertex is called an entry vertex if there is an external signal entering the system at that vertex.

Definition 1.7 Exit vertex: A vertex is called an exit vertex if an output signal leaves the system at that vertex.

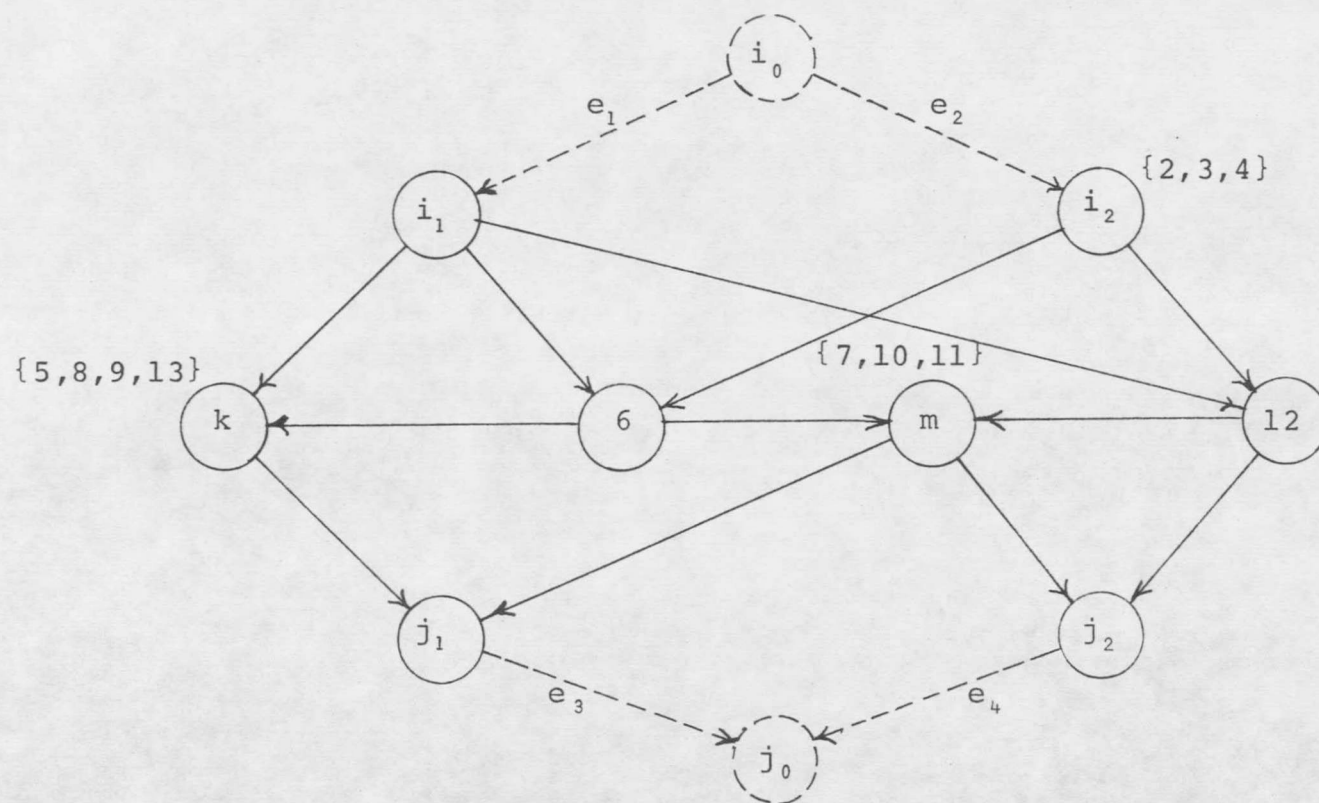
Example 1.1 In Figure 1.7(a) subgraphs consisting of vertices $\{5,8,9,13\}$ and $\{7,10,11\}$ are M.S.C. subgraphs. They are replaced by vertices k and m , respectively. The subgraph consisting of vertices $\{2,3,4\}$ is a link subgraph. It is replaced by vertex i_2 . A system graph obtained after such a reduction is called a basis graph, shown in Figure 1.7(b) without adding vertices i_0 and j_0 and edges $\{e_1, e_2, e_3, e_4\}$.

1.2.4 SEC graphs In this work we are studying properties of a special type of basis graph, one called a single-entry, single exit type of connected directed graph (a SEC graph). Research work in this area of fault diagnosis is still in



25

Figure 1.7(a) A system graph with M.S.C. and link subgraphs.



Reduced graph – without vertices i_0 and j_0 , and edges e_1, e_2, e_3, e_4

Figure 1.7(b) A SEC graph of the system graph in Figure 1.7(a).

a preliminary stage. The choice of this type of graph provides much insight into the problem, and later on helps in generalizing results to apply to the most general system graph. The definition of a SEC graph is given below:

Definition 1.8. SEC graph: A SEC graph is a connected directed graph which has a single entry (i) and a single exit (j) and which satisfies: 1) every vertex, except i and j have both incoming edges and outgoing edges; and 2) vertex i has no incoming edges, and vertex j has no outgoing edges.

In a SEC graph, a signal which is injected at vertex i may be detected only at vertex j . If any directed graph G contains n vertices, with k entry vertices $\{i_1, i_2, \dots, i_k\}$ and p exit vertices $\{j_1, j_2, \dots, j_p\}$, it can be transformed to a SEC graph as follows. Add new vertices i_0 and j_0 to G and add a directed edge from i_0 to every vertex in $\{i_1, i_2, \dots, i_k\}$ and a directed edge from every vertex in $\{j_1, j_2, \dots, j_p\}$ to j_0 . The graph G' so obtained will contain $(n+2)$ vertices and is a SEC graph from i_0 to j_0 .

In Figure 1.7(b), the addition of vertices i_0 and j_0 and edges e_1 , e_2 , e_3 and e_4 transforms a basis graph into a SEC graph.

It should be noted that if a SEC graph from i to j is separable with p as a cut-vertex, it is always possible to divide it into two subgraphs G_1 and G_2 by considering i as an entry vertex and j as an exit vertex G_1 and p as an entry vertex and j as an exit vertex G_2 as shown in Figure 1.6. Subgraphs G_1 and G_2 are SEC graphs. For this reason, we will study only nonseparable SEC graphs.

1.3 Statement of the Problem

Increased complexity of present-day computer systems and large-scale integration of microelectronic circuits have placed a heavy burden on testing and reliability. Classical methods of system diagnosis do not help much in understanding, visualizing, and analyzing such systems for somewhat obvious reasons. In classical methods, we are carrying a maze of behavioral and structural details at the same time while deriving diagnostic tests. Separation of behavioral and structural information usually simplifies the problem.

From a diagnostic point of view, the structural details can be studied using graph theory. The superposition of behavioral and structural details finally yields the required tests. Structural models for irredundant combinational networks were prepared by Russell and Kime [49]. They studied their models with regard to fault diagnosis and applied what they called the π -condensation method to structural models. This method demonstrates in a rigorous fashion the role of fan-out vertices and path reconvergence in the diagnosis of networks. A more systematic approach is cited in "A Structural Theory of Machine Diagnosis" [40] by Ramamoorthy. He suggested structural models that apply to large computer systems as well as to small combinational logic circuits.

A study of the structural models prepared by Russell and Kime [49] or Ramamoorthy [40] immediately reveals that no matter what method we use, due to the limited number of entry and exit vertices and also due to the complexities of the interconnection pattern of edges and vertices, it is impossible to identify for certain graphs which specific vertex or, equivalently, which functioning unit in the system is faulty. To simplify the problem, Ramamoorthy

and Mayeda [31,32,41] suggested the insertion of test points and blocking gates. A test point is placed to insert information (a test pattern) or to observe a result at a particular point in the system. A blocking gate blocks the flow of information through a path. Finally, diagnostic tests are performed on the system by controlling certain blocking gates, and injecting a required test pattern into a specific test point and observing test results at other specified test points. A test dictionary includes: 1) the test sequence; 2) the test pattern required to be injected, where it must be injected, and where it must be observed; and 3) which blocking gates must be controlled at that time.

As mentioned before, work in this area is in the preliminary stages, so attempts have been made to solve some of the basic problems. The value of the approach rests on the new insight it seems to provide to design and diagnostic engineers.

Chapter II is mainly devoted to the statement of assumptions, notation, and definitions which are frequently used in the remainder of this work. Theorems developed by Ramamoorthy and Mayeda [30,31,32,41] are

discussed. These theorems show how good the present system structure is from a diagnostic point of view and also show how much improvement can be obtained by the insertion of test points or blocking gates. Also necessary and sufficient conditions to get maximum distinguishability (the best improvement) are considered.

It is obvious that, by placing test points on every edge of a graph, we can find all faulty vertices by inserting signals on incoming edges and taking out signals at outgoing edges of vertices. But with such an approach, the size of the test dictionary would be unnecessarily large, with the time required for testing also excessive. If this method is applied to L.S.I., it adds to the existing pin-limitation problem. Also, the more test points, the more additional logic is required which increases the cost. The above statements show the need for realizing a minimum number of test points. Ramamoorthy and Mayeda [31] gave lower bounds on the number of test points required, but their work is not complete. Chapter III discusses the matter in detail and shows lower bounds on the number of test points required to obtain maximum distinguishability.

Chapter IV is devoted to developing an efficient algorithm to find the minimum number of test points and their locations to obtain maximum distinguishability in SEC graphs. Primary information required for the analysis is taken from the connectivity matrix of a graph (Definition 4.1). Here an attempt is made to connect the 'selection-of-minimum-test-points problem' to the well-known 'covering problem' [27,56]. Graph theory is used to reduce calculations considerably so that the algorithm can handle large system graphs efficiently.

Chapter V states conclusions from the work and outlines areas for further investigation. The appendices include computer program listings and a list of definitions used in this work.

Chapter II

PRELIMINARY IDEAS IN SYSTEM DIAGNOSIS USING GRAPH THEORY

2.1 Assumptions and Definitions

As we discussed in the last chapter, any discrete systems, Figures 1.2(a), 1.4(a) and 4.10 for example, can be shown to be isomorphic to a directed graph, Figures 1.2(b), 1.4(b) and 1.3, respectively. Vertices of the graph may be sequential or combinational elements, and directed edges represent lines of signal propagation and information flow. The theorems given in this chapter are well documented in the literature and are not proved herein. References to proofs are given, and numerous examples are presented to illustrate the concepts involved.

Definition 2.1 Directed (linear) graph: Let E and Ω be sets. If every $e \in E$ corresponds to exactly one pair $(v, v') \in \Omega$, then every member in E is an edge, every member in Ω is a vertex, and $\Omega \cup E$ is a 'linear graph'. If all edges in a linear graph are directed, then the linear graph is said to be a directed (or oriented) linear graph. 'Directed edge' is defined below.

Definition 2.2 Let e be an edge corresponding to a pair (v, v') of vertices. Then the two vertices v and v' are called the endpoints of edge e . v is the starting vertex of edge e , and v' is the final vertex of edge e . If the pair (v, v') is ordered, then e is said to be directed or is called a directed edge.

In this work we consider linear graphs only, so instead of saying 'directed linear graph', we say 'directed graph'.

Definition 2.3 A 'fault' is said to have occurred in a system if the input-output relationship for the system differs from that during its correct (fault-free) operation.

Definition 2.4 A 'fault vertex' is a vertex corresponding to a defective block, and a 'faultless vertex' is a vertex corresponding to a normally functioning block.

Assumption (1): A fault is of a permanent type so that it is sustained throughout the subsequent operation of the system until some corrective measures have been taken.

A fault is generally due to a physical malfunction in the system, and its result is an error in computation.

Assumption (2): When either an undistorted or a distorted signal is passed through a fault vertex, the signal will be distorted. On the other hand, when an undistorted (distorted) signal is passed through a faultless vertex, the signal will be undistorted (distorted).

It should be noted that the first assumption is the most common assumption made in studies of diagnosis. Even a transient fault can be modeled as a permanent fault, if the test procedure is very brief. A limited amount of study has been made of random faults [59]. In practice, the second assumption does not hold for all cases. On the other hand, by this assumption many theoretical results on diagnosis can be obtained easily. The results can usually be modified to cover particular cases where the assumption does not hold. We can see from the second assumption that once a signal is distorted, the signal can not be undistorted. Hence, by injecting a test pattern from the input vertex and monitoring an output signal, we can detect if the signal passed a fault vertex.

Definition 2.5 Directed path and directed circuit: A directed path from vertex v_1 to vertex v_k is a concatenation of direct edges leading from v_1 to v_k of the form

$(v_1, v_2), (v_2, v_3), (v_3, v_4), \dots, (v_{k-1}, v_k)$ in which all edges and vertices are distinct. If a directed path starts and terminates at the same vertex, that is $v_1 = v_k$, it is called a directed circuit.

Consider the graph in Figure 1.3; there is a directed path from 1 to 9 of the form $(1,2)(2,3)(3,4)(4,5)(5,6)(6,8)(8,9)$ and a directed circuit of the form $(4,5)(5,6)(6,7)(7,8)(8,4)$.

Definition 2.6 Measurable set $\Omega(i \times j)$: Let i and j be vertices in a directed graph G . A measurable set $\Omega(i \times j)$ is defined as a set of vertices such that vertex p is an element of $\Omega(i \times j)$ if and only if there is at least one directed path from vertex i to vertex p and at least one directed path from vertex p to vertex j . The set $\Omega(i \times j)$ is empty if there is no directed path from i to j .

Consider again the graph in Figure 1.7(b):

$\Omega(i_1 \times m) = (i_1, 6, 12, m)$ but $\Omega(k \times m)$ is empty. In the case of a SEC graph from i to j (Definition 1.8), $\Omega(i \times j)$ contains all vertices of the graph.

Definition 2.7 Complement of $\Omega(i \times j)$: Let Ω be a set of all vertices in a directed graph. Then

$$\overline{\Omega(i \times j)} = \Omega - \Omega(i \times j)$$

2.2 Distinguishability

Now let us consider the case in which a testing signal is injected from vertex i and an output signal is monitored at vertex j . If a fault is detected, the fault vertices must be in $\Omega(i \times j)$. On the other hand, if the signal monitored at j indicates the absence of a fault, then fault vertices can only be in $\overline{\Omega(i \times j)}$. Suppose monitoring at j indicates the absence of a fault, but monitoring at k indicates the presence of a fault. Then the fault vertices must be in $\overline{\Omega(i \times j)} \cap \Omega(i \times k)$. In short, by injecting signals at vertex i and monitoring signals at vertices j and k , we can find out which one of the following sets contains a fault vertex, if a fault vertex exists.

$$\Omega(i \times j) \cap \Omega(i \times k)$$

$$\Omega(i \times j) \cap \overline{\Omega(i \times k)}$$

$$\overline{\Omega(i \times j)} \cap \Omega(i \times k)$$

$$\overline{\Omega(i \times j)} \cap \overline{\Omega(i \times k)}$$

These sets form a collection called a D-partition of the measurable sets $\Omega(i \times j)$ and $\Omega(i \times k)$.

Definition 2.8 D-partition: Let n measurable sets

$\Omega(i_1 \times j_1), \Omega(i_2 \times j_2), \dots, \Omega(i_k \times j_k), \dots, \Omega(i_n \times j_n)$ be given.

A collection of sets $\bigcap_{k=1}^n \widetilde{\Omega(i_k \times j_k)}$ is called a D-partition of these measurable sets. Here $\bigcap_{k=1}^n \widetilde{\Omega(i_k \times j_k)}$ is all possible combinations of $\Omega(i_k \times j_k), k = 1, 2, \dots, n$ where $\widetilde{\Omega(i_k \times j_k)}$ is either $\Omega(i_k \times j_k)$ or $\overline{\Omega(i_k \times j_k)}$.

Example 2.1 Consider the directed graph in Figure 2.1.

For the measurable sets

$$\Omega(1 \times 4) = \{1, 3, 4, 5\}$$

$$\Omega(5 \times 6) = \{4, 5, 6\}$$

$$\begin{aligned} \text{D-partition} &= \{\Omega(1 \times 4) \cap \Omega(5 \times 6), \overline{\Omega(1 \times 4)} \cap \Omega(5 \times 6), \Omega(1 \times 4) \cap \\ &\quad \overline{\Omega(5 \times 6)}, \overline{\Omega(1 \times 4)} \cap \overline{\Omega(5 \times 6)}\} \\ &= \{(4, 5), (6), (1, 3), (2)\} \end{aligned}$$

Now, if any set in the D-partition $\{D\}$ consists of only one vertex and if that vertex is faulty, we can find out by performing the measurements corresponding to a given collection of measurable sets by which $\{D\}$ is obtained. In the above example, suppose a fault is known to

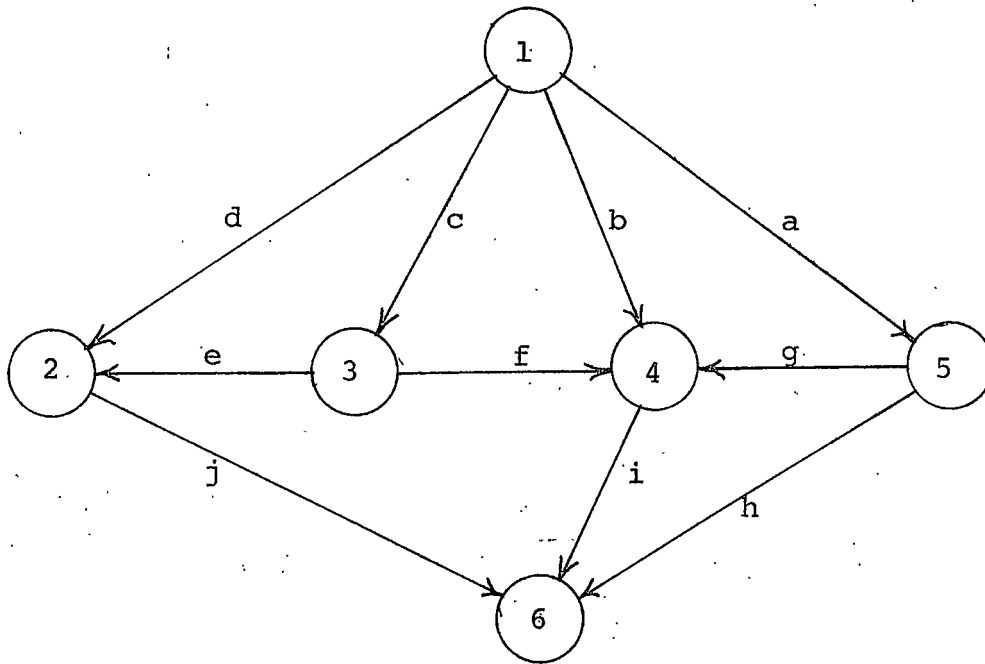


Figure 2.1 Directed graph for D-partition.

exist, but injecting signals at vertex 1 and monitoring at vertex 4 indicates the absence of a fault, and also injecting signals to vertex 5 and monitoring at vertex 6 indicates the absence of a fault. We can only conclude that the faulty vertex must be 2. On the other hand, if two vertices v_1 and v_2 are together in a set in a D-partition under a collection M of measurable sets and if one of these two vertices is faulty, then it is impossible to determine which of these vertices is faulty just by measurements corresponding to the measurable sets in M . In the above example, the collection of measurable sets $\Omega(1 \times 4)$ and $\Omega(5 \times 6)$ are not sufficient to distinguish whether vertex 4 or vertex 5 is faulty. Similarly, vertex 1 and vertex 3 are not fault-distinguishable.

Suppose all nonempty sets in a D-partition consist only of one vertex each. Then we can find exactly which vertex is faulty by measurements corresponding to a given collection of measurable sets M by which $\{D\}$ is obtained. Here it should be noted that any two nonempty sets in a D-partition of a collection of measurable sets have no vertices in common (for the proof, refer to [30]).

A D-partition provides a structural resolution of the system graph. It shows how well-conditioned the system

structure is for diagnostic purposes. Given numbers of entry and exit vertices, we find all possible measurable sets (M) and obtain a D-partition. The greater the number of vertices in any set in the D-partition, the poorer the system is from a diagnostic point of view; in other words, our ability to distinguish a fault vertex from normal vertices is more difficult or sometimes impossible. Let us first define k-distinguishability.

Definition 2.9 k-distinguishability: A directed graph is k-distinguishable under a collection M of measurable sets if

1) There exists a set R in a D-partition of M such that R contains k vertices, and 2) there are no sets in the same D-partition of M which contain more than k vertices.

For example, the directed graph of Figure 2.1 is 2-distinguishable under a collection of measurable sets $\Omega(1 \times 4)$ and $\Omega(5 \times 6)$ because there are sets (4,5) and (1,3) that contain two vertices, and there is no other set which has more than two vertices.

Our desire is to obtain 1-distinguishability for any graph, so that exactly which vertex is faulty can be

determined. However, because of a limited number of entry and exit vertices and because of the inherent structural pattern of interconnections between vertices, it is not generally possible to obtain 1-distinguishability.

2.3 Test Points

2.3.1 Introduction One possible way [31] to obtain 1-distinguishability is to create more entry and exit terminals that result in more measurable sets and in that way improve the structural resolution. These extra entry and exit terminals (apart from regular entry and exit terminals) are called test points or inner terminals. A test point is considered as an additional vertex which is placed on an edge so that we can inject as well as monitor a signal for diagnosis. However, when the distinguishability of vertices of G is considered, we do not wish to consider an inserted vertex as a member of the set of vertices to be tested.

It must be remembered that certain graphs can not be made 1-distinguishable by using any number of test points. The graph in Figure 2.2, for example, is not 1-distinguishable even though we place test points on all edges of the graph because any measurable set obtained by

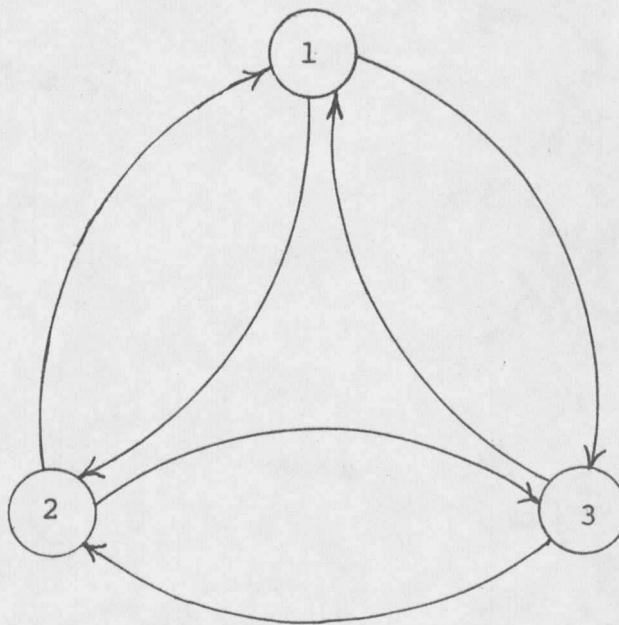


Figure 2.2 Directed graph not 1-distinguishable by the test point method.

all possible combinations of regular and extra terminals has all vertices of the graph, and hence, there exists a set in the D-partition that contains more than one vertex. In order to remove such difficulties, the approach of blocking gates [41] is used by which we can delete edges or block the flow of information whenever necessary. Properties of blocking gates are described in Section 2.4 of this chapter.

Definition 2.10 Test point: Let edge e be connected from vertex v_1 to v_2 . A test point is a point which, when assigned to e , allows a signal to be injected through e into v_2 , blocking the signal from v_1 , or enables a signal which passes through e from v_1 to v_2 to be monitored.

Definition 2.11 The symbols $v(e^+)$ and $v(e^-)$ indicate the vertices such that edge e is connected from vertex $v(e^+)$ to $v(e^-)$.

For the graph in Figure 2.1,

$$v(d^+) = 1, \quad v(d^-) = 2.$$

Definition 2.12 The symbol $\Omega(e^\pm \times j)$ is shorthand notation for $\Omega(v(e^\pm) \times j)$; similarly, $\Omega(i \times e^\pm)$ is shorthand notation

for $\Omega(i \times v(e^\pm))$, and $\Omega(e^\pm, d^\pm)$ is shorthand notation for $\Omega(v(e^\pm), v(d^\pm))$.

For the graph in Figure 2.1,

$$\Omega(a^+ \times 2) = \Omega(v(a^+) \times 2) = \Omega(1 \times 2) = \{1, 3, 2\}.$$

$$\Omega(3 \times h^-) = \Omega(3 \times v(h^-)) = \Omega(3 \times 6) = \{3, 2, 4, 6\}.$$

$$\Omega(g^+ \times j^-) = \Omega(v(g^+) \times v(j^-)) = \Omega(5 \times 6) = \{5, 4, 6\}.$$

Definition 2.13 The symbol $\Omega(G_i)$ indicates the set of all vertices in the graph G_i .

Definition 2.14 The symbol $\mathcal{E}(\Omega_1 \times \bar{\Omega}_1)$ indicates a set of all edges, each of which is directed from a vertex in Ω_1 to a vertex in $\bar{\Omega}_1$ (where $\bar{\Omega}_1 = \Omega - \Omega_1$, and Ω is the set of all vertices in the graph).

Definition 2.15 Cut-set and directed cut-set of a directed graph: Cut-set $S = \mathcal{E}(\Omega_1 \times \bar{\Omega}_1) \cup \mathcal{E}(\bar{\Omega}_1 \times \Omega_1)$ is a set of all edges connected between a set Ω_1 and its complement $\bar{\Omega}_1$. When one (but not both) of $\mathcal{E}(\Omega_1 \times \bar{\Omega}_1)$ and $\mathcal{E}(\bar{\Omega}_1 \times \Omega_1)$ is an empty set, then S is called a directed cut-set (where $\bar{\Omega}_1 = \Omega - \Omega_1$, and Ω is the set of all vertices in the graph).

Definition 2.16 Maximal connected subgraphs of a linear graph G are a set of connected subgraphs $g_1, g_2, g_3, \dots, g_j$ such that every edge and every vertex in G is in exactly one of these subgraphs.

An isolated vertex is a connected subgraph by definition.

2.3.2 Theorems on necessary and/or sufficient conditions

Definition 2.17 A directed cut-set $S = \mathcal{E}(\Omega_1 \times \bar{\Omega}_1) (\mathcal{E}(\bar{\Omega}_1 \times \Omega_1))$ is said to separate the vertices i and j , if $i \in \Omega_1$ ($i \in \bar{\Omega}_1$) and $j \in \bar{\Omega}_1$ ($j \in \Omega_1$).

Theorem 2.1 [30] Let G be a SEC graph from i to j . Suppose $S(e)$ is a directed cut-set which separates vertices i and j and contains edge e . Further suppose that there is no other directed cut-set separating i and j that contains edge e . Let G_1 and G_2 be the maximal connected subgraphs obtained from G by deleting all edges in $S(e)$. Then $\Omega(i \times e^+) = \Omega(G_1)$ and $\Omega(e^- \times j) = \Omega(G_2)$.

When there is more than one directed cut-set containing edge e and separating i and j , another theorem applies.

Theorem 2.2 [30] Let G be a SEC graph from i to j . Suppose k directed cut-sets $S_1(e), S_2(e), \dots, S_k(e)$ are those which contain edge e among all directed cut-sets separating i and j . Then $\Omega(i \times e^+) = \Omega(G'_1)$ and $\Omega(e^- \times j) = \Omega(G'_2)$ where G'_1 and G'_2 are two maximal connected subgraphs obtained from G by deleting all edges in $S_1(e), S_2(e), \dots, S_k(e)$.

In the proof of Theorem 2.2, two directed cut-sets $S_2(e)$ and $S_k(e)$ are used so that the deletion of all edges in these two cut-sets produces G'_1, G'_2 , and G'_3 (G'_3 is obtained by deleting all vertices in G'_1 and G'_2 and all edges connected to these vertices). We say that two directed cut-sets $S_1(e)$ and $S_k(e)$ are 'covered' by edge e or that edge e covers $S_1(e)$ and $S_k(e)$. Instead of two, if there exists only one directed cut-set $S(e)$ as in Theorem 2.1, we then say that edge e covers the directed cut-set $S(e)$ only.

Definition 2.18 Edge e is said to cover a directed cut-set S , if S is the only directed cut-set which separates i and j and contains edge e . Edge e covers two directed cut-sets $S_1 = \mathcal{E}(\Omega_1 \times \bar{\Omega}_1)$ and $S_2 = \mathcal{E}(\Omega_2 \times \bar{\Omega}_2)$ if and only if

1. S_1 and S_2 separate i and j ,
2. S_1 and S_2 contain edge e , and
3. Any other cut-set $S' = \mathcal{C}(\Omega_3 \times \bar{\Omega}_3)$ separating i and j and containing edge e satisfies the condition

$$(\Omega_1 \cup \Omega_2) \supset \Omega_3 \supset (\Omega_1 \cap \Omega_2)$$

Example 2.2 Consider a SEC graph from 1 to 6, as shown in Figure 2.3(a). Directed cut-sets that separate 1 and 6 and contain edge e are

$$S_1(e) = \mathcal{C}[\Omega_1 \times \bar{\Omega}_1] = \mathcal{C}[(1,2) \times (\bar{1},2)] = \{b, c, e\}$$

$$S_2(e) = \mathcal{C}[\Omega_2 \times \bar{\Omega}_2] = \mathcal{C}[(1,2,3) \times (\bar{1},2,3)] = \{b, d, e, f\}$$

$$S_3(e) = \mathcal{C}[\Omega_3 \times \bar{\Omega}_3] = \mathcal{C}[(1,2,3,4) \times (\bar{1},2,3,4)] = \{e, f, g\}$$

$$S_4(e) = \mathcal{C}[\Omega_4 \times \bar{\Omega}_4] = \mathcal{C}[(1,2,3,4,5) \times (\bar{1},2,3,4,5)] = \{e, h\}$$

$$(\Omega_1 \cup \Omega_4) \supset \Omega_k \supset (\Omega_1 \cap \Omega_4)$$

$$\text{for } \Omega_k = \Omega_2 \text{ or } \Omega_3$$

Thus, edge e covers two directed cut-sets $S_1(e)$ and $S_4(e)$. The deletion of all edges in these two cut-sets produces three subgraphs G'_1 , G'_2 , and G'_3 , as shown in Figure 2.3(c).

$$S(a) = \mathcal{C}[\Omega_5 \times \bar{\Omega}_5] = \mathcal{C}[(1) \times (\bar{1})] = \{a, b\}$$

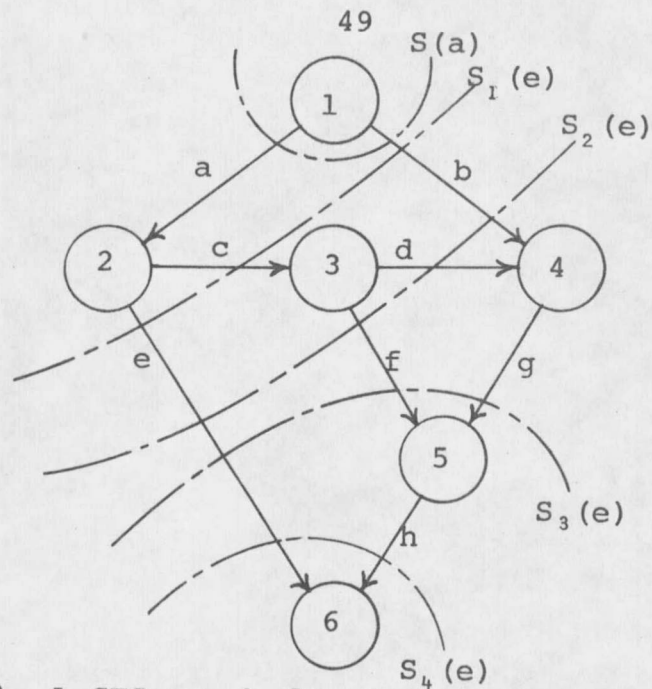


Figure 2.3(a) A SEC graph for Example 2.2.

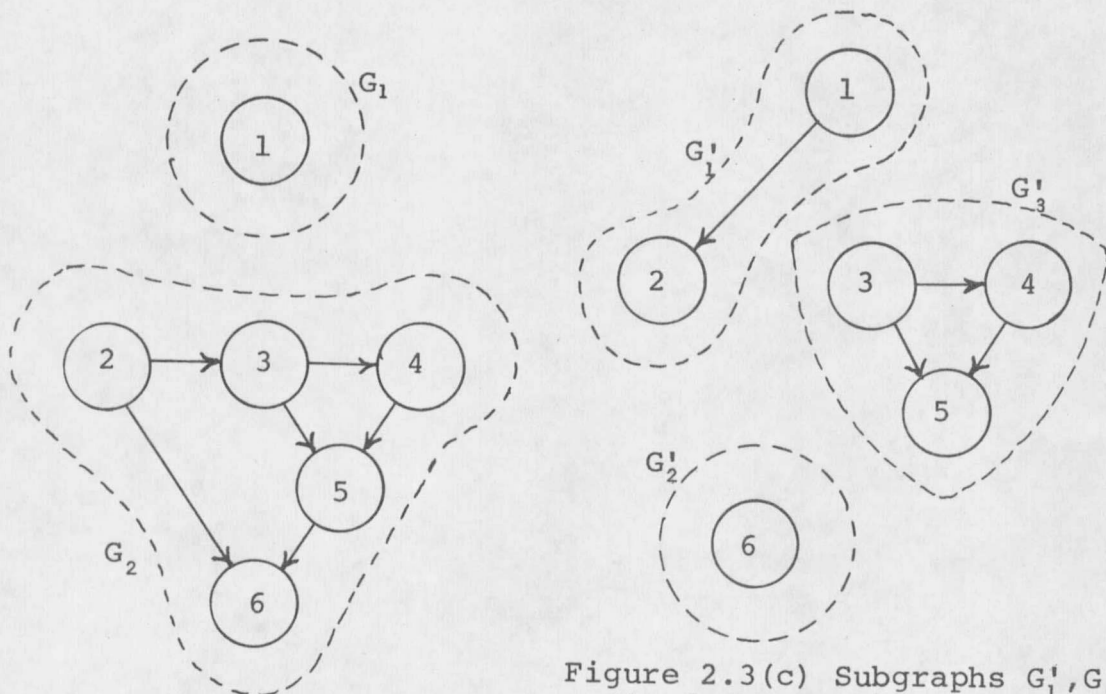


Figure 2.3(b) Subgraphs G_1 & G_2 of the SEC graph in Fig. 2.3(a). Figure 2.3(c) Subgraphs G'_1, G'_2, G'_3 of the SEC graph in Fig. 2.3(a).

This is the only directed cut-set which separates 1 and 6 and contains edge 'a'. So, edge 'a' covers $S(a)$.

Two maximal connected subgraphs G_1 and G_2 obtained by deleting all edges of $S(a)$ are shown in Figure 2.3(b).

The converses of Theorems 2.1 and 2.2 are also true.

Theorem 2.3 Let G be a SEC graph from i to j . Let $\Omega(i \times e^+)$ and $\Omega(e^- \times j)$ be measurable sets. If $\Omega(i \times e^+) \cap \Omega(e^- \times j) = \phi$, then there exist two directed cut-sets $S(e)$ and $S'(e)$ containing edge e such that $\Omega(G_e) = \Omega(i \times e^+)$ and $\overline{\Omega(G'_e)} = \Omega(e^- \times j)$, where

$$S(e) = \mathcal{G}[\Omega(G_e) \times \overline{\Omega(G'_e)}]$$

and

$$S'(e) = \mathcal{G}[\Omega(G'_e) \times \overline{\Omega(G_e)}]$$

Theorem 2.4 Let G be a SEC graph from i to j . Suppose measurable sets $\Omega(i \times e^+)$ and $\Omega(e^- \times j)$ have the property that

$$\Omega(i \times e^+) \cap \Omega(e^- \times j) \neq \phi$$

Then there exists no directed cut-set separating i and j which contains edge e .

Proofs of the above two theorems are given in [30].

Definition 2.19 For a set E of edges, the phase 'measurable sets under E ' means the total of all possible measurable sets which result when we assign test points to all edges in E .

Definition 2.20 A SEC graph from i to j being 1-distinguishable under set E of edges means that when test points are assigned to every edge in set E , measurable sets under E will be 1-distinguishable.

For example, consider the SEC graph in Figure 2.3(a). Suppose the test points are assigned on edges c and g . Then $E = \{c, g\}$, and measurable sets under E are

$$\Omega(1 \times c^+) = \{1, 2\}$$

$$\Omega(1 \times g^+) = \{1, 2, 3, 4\}$$

$$\Omega(1 \times 6) = \{1, 2, 3, 4, 5, 6\}$$

$$\Omega(c^- \times g^+) = \{3, 4\}$$

$$\Omega(c^- \times 6) = \{3, 4, 5, 6\}$$

$$\Omega(g^- \times c^+) = \phi$$

$$\Omega(g^- \times 6) = \{5, 6\}$$

Theorem 2.5 Necessary condition for 1-distinguishability:

Let G be a SEC graph from i to j . Also, let E be a set of edges and $\{S\}$ be a collection of all directed cut-sets covered by edges in E . Then if there exists an edge in G which is not in any directed cut-set in $\{S\}$, G is not 1-distinguishable under E .

Example 2.3 Consider the SEC graph in Figure 2.4. Let us take $E = \{a, d, e\}$. Then the collection of directed cut-sets $\{S\}$ covered by the edge-set E is shown in Figure 2.4 and consists of $\{S(a), S(d), S_1(e), S_4(e)\}$, where $S(a) = \{a, b\}$, $S_1(e) = \{b, c, e\}$, $S_4(d) = \{e, h\}$, and $S(d) = \{b, d, f, e\}$.

Since the edge g is not in any of these directed cut-sets, graph G is not 1-distinguishable under $E = \{a, d, e\}$. We can check the result by the D-partition approach. All possible nonempty measurable sets are

$$\Omega(1 \times 6) = \{1, 2, 3, 4, 5, 6\}$$

$$\Omega(1 \times a^+) = \{1\}$$

$$\Omega(1 \times d^+) = \{1, 2, 3\}$$

$$\Omega(1 \times e^+) = \{1, 2\}$$

$$\Omega(a^- \times 6) = \{2, 3, 4, 5, 6\}$$

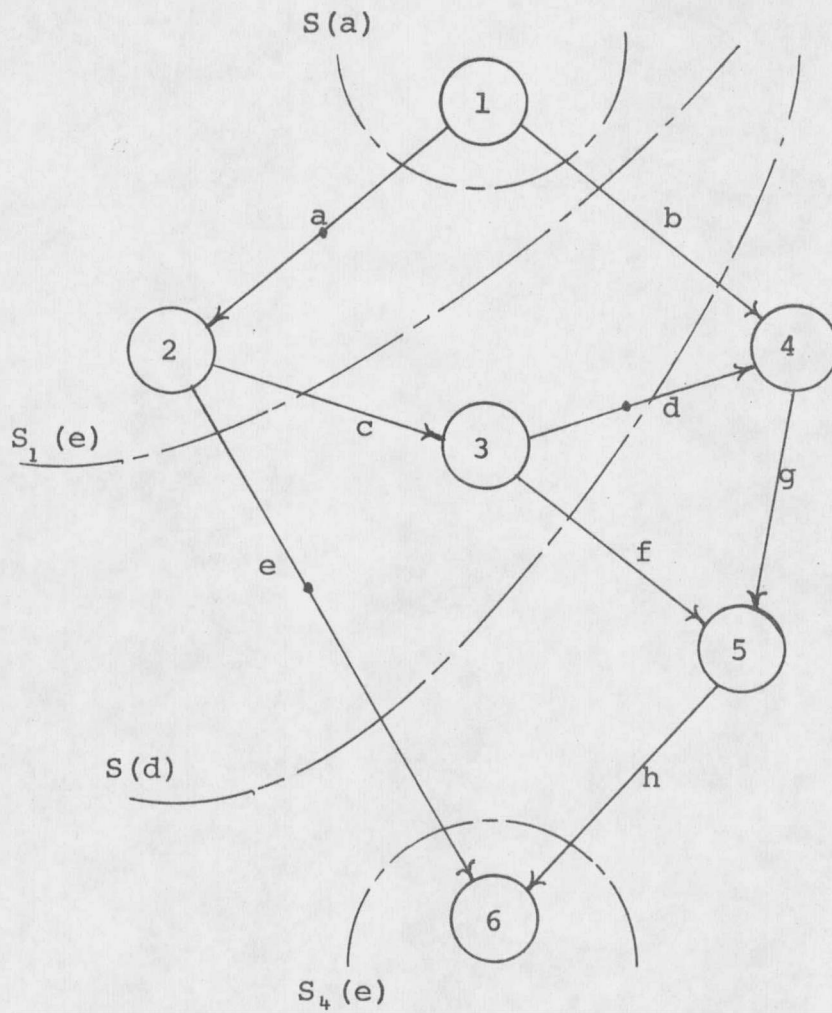


Figure 2.4 A SEC graph and directed cut-sets covered by $E = \{a, d, e\}$.

$$\Omega(a^- \times d^+) = \{2, 3\}$$

$$\Omega(a^- \times e^+) = \{2\}$$

$$\Omega(d^- \times 6) = \{4, 5, 6\}$$

and

$$\Omega(e^- \times 6) = \{6\}$$

$$\begin{aligned} &\text{In the D-partition, we have a set } \{\Omega(1 \times 6) \cap \overline{\Omega(1 \times a^+)} \cap \\ &\overline{\Omega(1 \times d^+)} \cap \overline{\Omega(1 \times e^+)} \cap \Omega(a^- \times 6) \cap \overline{\Omega(a^- \times d^+)} \cap \overline{\Omega(a^- \times e^+)} \cap \Omega(d^- \times 6) \cap \\ &\overline{\Omega(e^- \times 6)}\} \end{aligned}$$

$$\begin{aligned} &= \{(1, 2, 3, 4, 5, 6) \cap (2, 3, 4, 5, 6) \cap (4, 5, 6) \cap (3, 4, 5, 6) \cap \\ &\quad (2, 3, 4, 5, 6) \cap (1, 4, 5, 6) \cap (1, 3, 4, 5, 6) \cap (4, 5, 6) \cap \\ &\quad (1, 2, 3, 4, 5)\} \\ &= (4, 5) \end{aligned}$$

There is a set in the D-partition which contains more than one vertex. So, G is not 1-distinguishable.

Theorem 2.5 provides a necessary condition for 1-distinguishability. It is not a sufficient condition. Suppose there is a set E of edges and let $\{S\}$ be a collection of all directed cut-sets covered by edges in E , and

suppose that every edge of graph G is in a cut-set in $\{S\}$. Is graph G 1-distinguishable under E ? The answer is NO, in general.

Example 2.4 Consider the SEC graph of Figure 2.5. Let $E = \{a, b\}$. A collection of directed-cuts, $\{S\} = \{S_1(a), S_2(a), S_1(b), S_2(b)\}$, is covered by $E = \{a, b\}$. All the edges of the graph are in directed cut-sets of $\{S\}$, but still graph G is not 1-distinguishable. Checking the result by a D-partition, vertices 4 and 5 are together in a set in the D-partition under the measurable sets under E . Similarly, so also are vertices (4,6) and (5,6).

The following two theorems of Ramamoorthy and Mayeda [30,31] give a necessary and sufficient conditions for 1-distinguishability for a SEC graph.

Let us re-examine Example 2.4. Pairs of vertices, except (4,5), (4,6), and (5,6), are separated by the directed cut-sets of $\{S\}$. As the above pairs of vertices are not separated by any of the directed cut-sets in $\{S\}$, we do not have 1-distinguishability. However, if all possible pairs of vertices of graph G are separated by the direct-cuts of $\{S\}$, we may establish a necessary and sufficient condition.

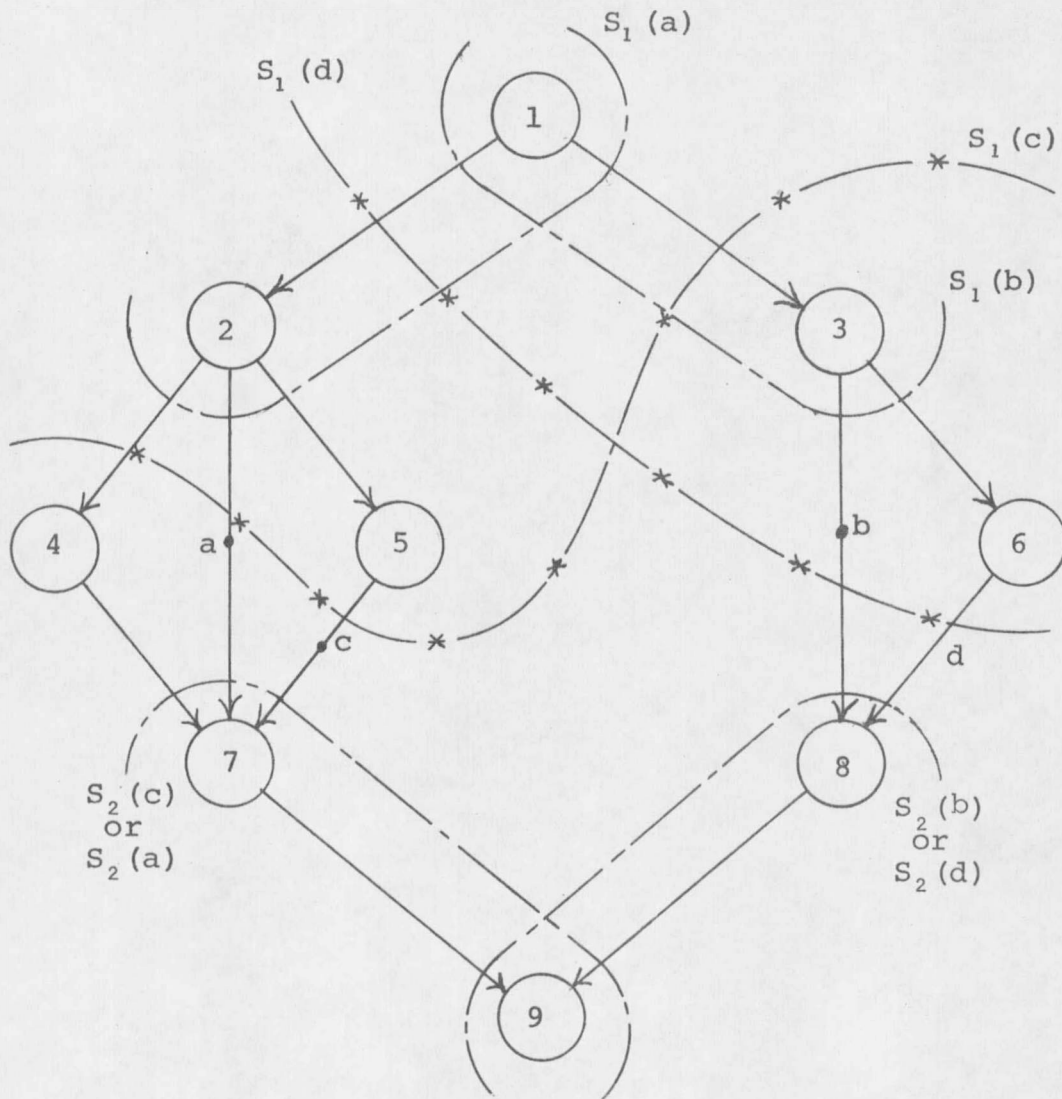


Figure 2.5 A SEC graph which is not 1-distinguishable by an edge-set $E = \{a, b\}$ (The directed cut-sets covered by E contain all edges of the SEC graph).

Theorem 2.6 Let G be a SEC graph from i to j and E be a set of edges. Let $\{\Omega\}$ be a collection of all measurable sets produced by assigning test points to all edges in E . Then if and only if for every pair of vertices v_r and v_s in G there exists a measurable set $\Omega' \in \{\Omega\}$ such that Ω' contains either v_r or v_s but not both, G is 1-distinguishable under E .

Theorem 2.7 Let G be a SEC graph from i to j and E be a set of edges. Let $\{S\}$ be a collection of all possible directed cut-sets separating i and j and covered by edges in E . Then G is 1-distinguishable under E if and only if for any pair of vertices v_r and v_s there exists a directed cut-set in $\{S\}$ which separates v_r and v_s .

In Example 2.4, nonempty measurable sets under $E = \{a, b\}$ are

$$\Omega(1 \times 9) = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

$$\Omega(1 \times a^+) = \{1, 2\}$$

$$\Omega(1 \times b^+) = \{1, 3\}$$

$$\Omega(a^- \times 9) = \{7, 9\}$$

and

$$\Omega(b^- \times 9) = \{8, 9\}.$$

There exists a pair of vertices $(4,5)$ such that both of them are in $\Omega(1 \times 9)$, and they are not in $\Omega(1 \times a^+)$, $\Omega(1 \times b^+)$, $\Omega(a^- \times 9)$ and $\Omega(b^- \times 9)$. There is, therefore, no measurable set $\Omega' \in \{\Omega\}$ such that Ω' contains either 4 or 5 but not both. By Theorem 2.6, graph G is not 1-distinguishable under E . Now the collection $\{S\}$ contains directed cut-sets

$$S_1(a) = \mathcal{G}[(1,2) \times (3,4,5,6,7,8,9)]$$

$$S_2(a) = \mathcal{G}[(1,2,3,4,5,6,8) \times (7,9)]$$

$$S_1(b) = \mathcal{G}[(1,3) \times (2,4,5,6,7,8,9)]$$

and

$$S_2(b) = \mathcal{G}[(1,2,3,4,5,6,7) \times (8,9)].$$

None of these separates the pairs of vertices $(4,5)$, $(5,6)$, and $(4,6)$. By Theorem 2.7, therefore, G is not 1-distinguishable. It can be seen, however, that the addition of one more edge, c , in $E = \{a,b\}$ adds one more directed cut-set $S_1(c)$ in $\{S\}$, where $\{S\}$ is now covered by $E = \{a,b,c\}$ and $S_1(c) = \mathcal{G}[(1,2,5) \times (3,4,6,7,8,9)]$. But still none of the directed cut-sets separates the pair of vertices $(4,6)$. Thus, by Theorem 2.7, G is not 1-distinguishable.

Finally, the addition of edge d in $E = \{a, b, c\}$ makes graph G 1-distinguishable because the new added directed cut-set $S_1(d)$ in $\{S\}$ separates $(4, 6)$.

$$S_1(d) = \mathcal{G}[(1, 3, 6) \times (2, 4, 5, 7, 8, 9)].$$

Proofs of the above theorems are given in [30].

2.3.3 Ordering a SEC graph It should be noted that it is an exhaustive task to find directed cut-sets covered by a given set of edges E , and it is even more laborious to check whether $\{S\}$ separates each pair of vertices of graph G , because the number of pairs of vertices increases very rapidly with increases in the number of vertices of the graph. Simplification may be obtained for special types of SEC graphs by the study of ordering of vertices.

Definition 2.21 Two vertices v_r and v_s are ordered as $v_r > v_s$ if there exists no directed paths from v_s to v_r .

It can be proved that it is always possible to order vertices of a SEC graph G from i to j , provided the graph contains no directed circuits.

Definition 2.22 Ring sum ($\oplus$) of two sets S_1 and S_2 : The ring sum of S_1 and S_2 is denoted by

$$S_1 \oplus S_2 \triangleq (S_1 \cup S_2) - (S_1 \cap S_2).$$

Definition 2.23 Let $S_1, S_2, S_3, \dots, S_m$ be sets. Then $S_1, S_2, S_3, \dots, S_m$ are said to be 'linearly independent sets' if

$$S_1 \oplus S_2 \oplus S_3 \oplus \dots \oplus S_m \neq \phi$$

where ϕ is a null set.

Example 2.5 $S_1 = \{1,3,6\}$, $S_2 = \{1,2,4\}$, $S_3 = \{2,3,4,6\}$, and $S_4 = \{3,4,5,7\}$ are given sets.

$S_1 \oplus S_2 \oplus S_3 = \phi$. So, sets S_1, S_2 , and S_3 are not linearly independent. However, $S_1 \oplus S_2 \oplus S_4 = \{2,5,6,7\} \neq \phi$, so, sets S_1, S_2 , and S_4 are linearly independent.

Theorem 2.8 Necessary and sufficient condition for 1-distinguishability [30] Let G be a SEC graph from i to j which contains no directed circuits. Suppose the ordering of vertices is unique. Then G is 1-distinguishable under a set E of edges if and only if edges in E cover all linearly independent directed cut-sets which separate i and j .

Example 2.6 Consider the SEC graph of Figure 2.6. The vertices are ordered as follows:

$$1 > 4 > 3 > 2 > 6 > 5 > 7$$

This ordering is unique. $S_1, S_2, \dots, S_6$ are $(n-1)$ linearly independent directed cut-sets which separate 1 and 7. This graph is 1-distinguishable under edge-set $E = \{a, b, c, d\}$, edges of which cover all linearly independent directed cut-sets $S_1, S_2, \dots, S_6$.

The above condition results in very simple calculations, but it applies to a very small class of graphs. From the above theorem, if the condition of uniqueness of ordering of vertices is removed, we have only a sufficient condition.

Theorem 2.9 Sufficient condition for 1-distinguishability

[30]: Let G be a SEC graph from i to j consisting of n vertices. If $(n-1)$ linearly independent directed cut-sets separating i and j are covered by edges in set E , then G is 1-distinguishable under set E .

When a set E of edges is a directed cut-set, the sufficient condition for 1-distinguishability becomes easier.

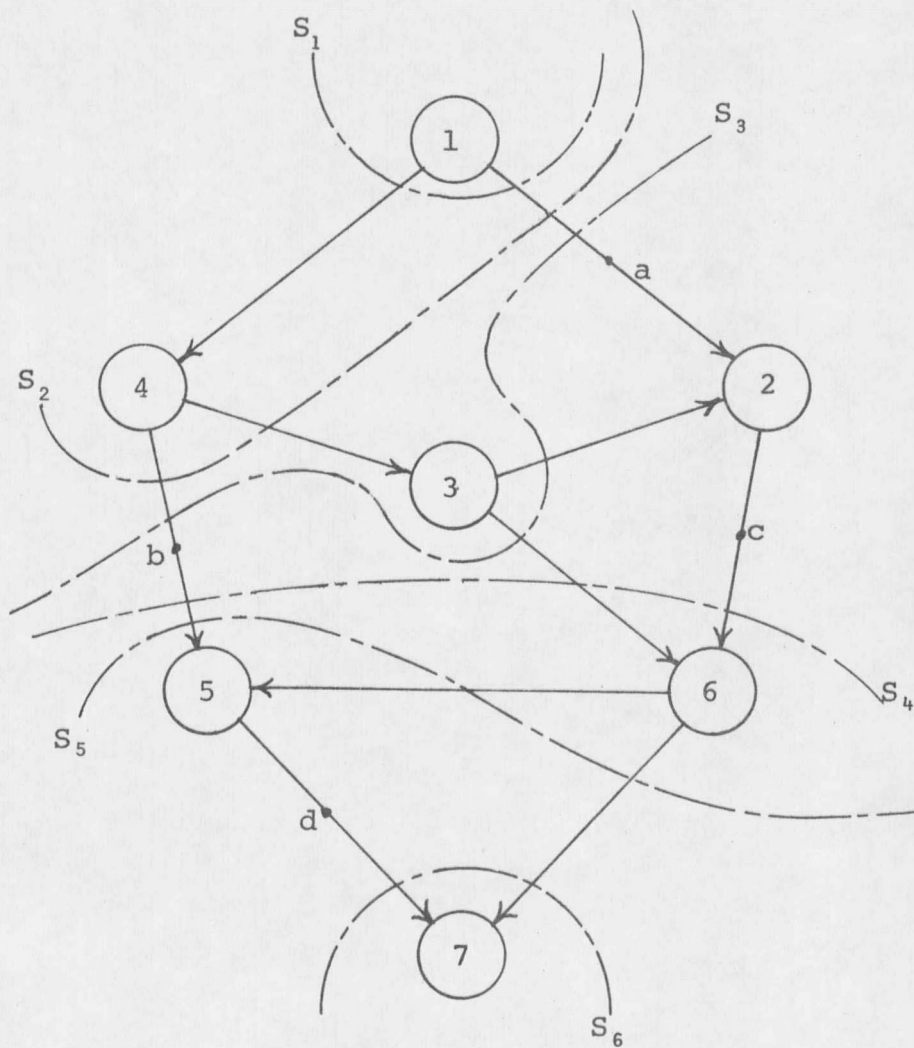


Figure 2.6 A SEC graph and six linearly independent directed cut-sets.

Theorem 2.10 Sufficient condition for 1-distinguishability: Let G be a SEC graph from i to j . Suppose a set E of edges is a directed cut-set separating i and j in G . Then G is 1-distinguishable under E if and only if every edge in G is in a cut-set in $\{S\}$ where $\{S\}$ is a collection of all directed cut-sets separating i and j covered by edges in E .

2.4 Blocking Gates

So far we have discussed the test point approach, but we can easily verify that whenever a directed circuit is in a SEC graph, 1-distinguishability becomes impossible. To overcome such difficulties, blocking gates are used instead of test points [30,41].

Definition 2.24 A blocking gate being assigned on an edge means that information flow on that edge can be blocked or unblocked. In contrast to a test point, a blocking gate can not be used to inject or monitor a signal. It merely blocks information. Thus, test signals will be inserted at the primary (specified) input points, and results will be monitored at the primary output points.

Implementation of blocking gates varies according to practical situations. In the simplest case, the

blocking of an edge e could be equivalent to imposing a signal of logical 0 or 1 levels; in more complex cases, a blocking gate can consist of several logic gates.

Example 2.7 Consider the circuit diagram shown in Figure 2.7(a). Its equivalent structural diagram is shown in Figure 2.7(b). Suppose we have to block information flow through edges e_4 and e_5 ; it can be done by introducing OR and AND logic gates and imposing signal 1 and 0 on input X and Y on edges e_4 and e_5 , respectively. Unblocking of edges e_4 and e_5 is achieved by changing the signals on X and Y from 1 to 0 and 0 to 1, respectively. Under blocked condition, the signal on edge e_2 passes through vertex 3 to e_6 regardless of the signal on edge e_4 . A similar condition prevails for the other blocked edge.

Definition 2.25 The symbol $\Omega(i \times j)_E$ is a measurable set of a subgraph obtained by deleting all edges in set E.

Definition 2.26 The measurable sets by gate-set E means all possible measurable sets which are produced by assigning blocking gates to all edges in set E.

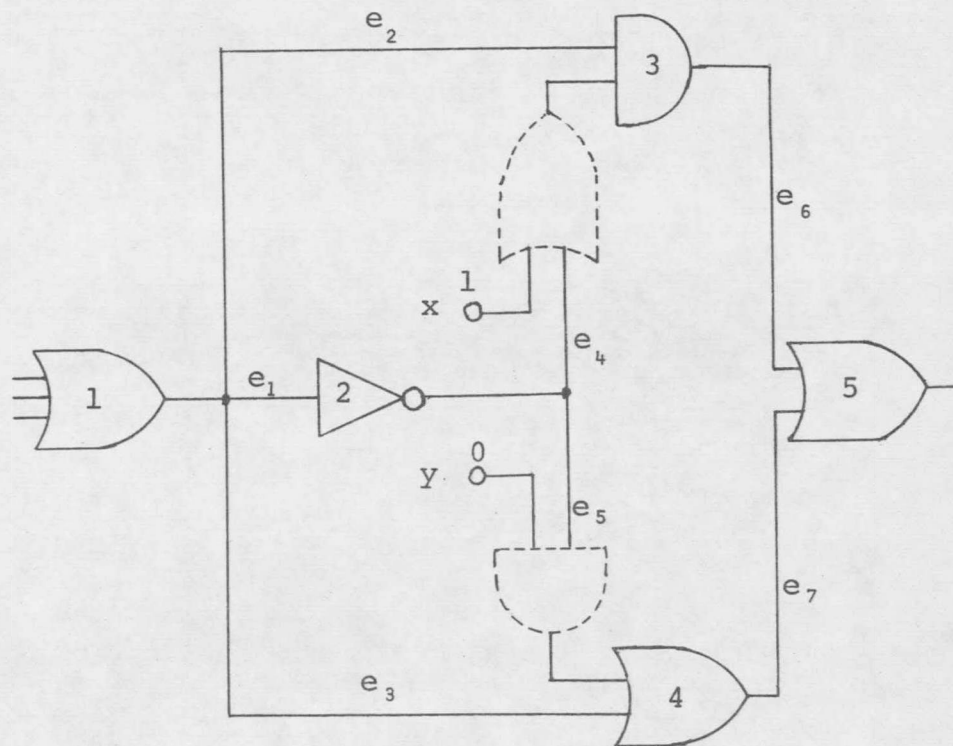


Figure 2.7(a) A logic circuit and design of blocking gates on edges e_4 and e_5 .

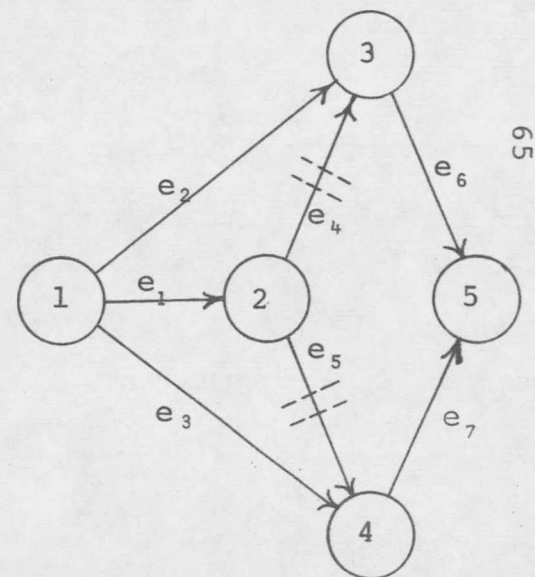


Figure 2.7(b) A structural representation of the logic circuit in Figure 2.7(a).

Example 2.8 Consider a SEC graph as shown in Figure 2.8.

Suppose $E = \{e_1, e_2, e_3\}$

$$\Omega(1 \times 6)_{e_1} = \Omega(1 \times 6)_{e_1 e_3} = \{1, 2, 3, 5, 6\}$$

$$\Omega(1 \times 6)_{e_2 e_3} = \Omega(1 \times 6)_{e_1 e_2 e_3} = \{1, 5, 6\}$$

$$\Omega(1 \times 6)_{e_2} = \{1, 2, 3, 4, 5, 6\}$$

$$\Omega(1 \times 6)_{e_1 e_2} = \{1, 3, 5, 6\}$$

Measurable sets $\Omega(1 \times 6)_{e_1}$, $\Omega(1 \times 6)_{e_2 e_3}$, $\Omega(1 \times 6)_{e_2}$, and $\Omega(1 \times 6)_{e_1 e_2}$ are the measurable sets by gate set $\{e_1, e_2, e_3\}$.

Definition 2.27 Suppose $\Omega(i \times j)_E$ is not the same as $\Omega(i \times j)_E^*$ for every proper subset E^* of E . Then E is called the minimum set for $\Omega(1 \times j)_E$.

In the above example, $E = \{e_1, e_2, e_3\}$ is not the minimum set for $\Omega(1 \times 6)_E$, as there exists proper subset $E^* = \{e_2, e_3\}$ such that $\Omega(1 \times 6)_E = \Omega(1 \times 6)_{E^*}$.

The minimum set for a measurable set may not be unique. For example, (e_1, e_2) and (e_1, e_4) are minimum sets for measurable sets $\{1, 3, 5, 6\}$.

Let E be the minimum set for $\Omega(k \times j)_E \neq \emptyset$ of a SEC graph from i to j . Then by deleting all edges in E from G , the resultant graph is connected.

We apply the same method that we used in test points to obtain a D-partition in the case of blocking gates. In the above example, we use measurable sets $\Omega(1 \times 6)_{e_1}$, $\Omega(1 \times 6)_{e_2 e_3}$, $\Omega(1 \times 6)_{e_2}$, and $\Omega(1 \times 6)_{e_1 e_2}$ for gate sets $E = \{e_1, e_2, e_3\}$. In all of the above measurable sets, entry vertex 1 and exit vertex 6 appear--a property which is true for any SEC graph and for any gate-set selection, as we are always injecting and monitoring signals at primary entry and exit vertices, respectively. In a D-partition, therefore, wherever an entry vertex appears, an exit vertex appears in the same set. By Definition 2.9, any SEC graph from i to j is not 1-distinguishable under any gate-set E . Obviously, there are SEC graphs that are 1-distinguishable under gate-sets if we neglect one of the vertices i or j . Hence, we need the following definition:

Definition 2.28 A SEC graph G from i to j is 1-(i,j)-distinguishable under a gate-set E if G is 1-distinguishable under set E when vertex j is disregarded.

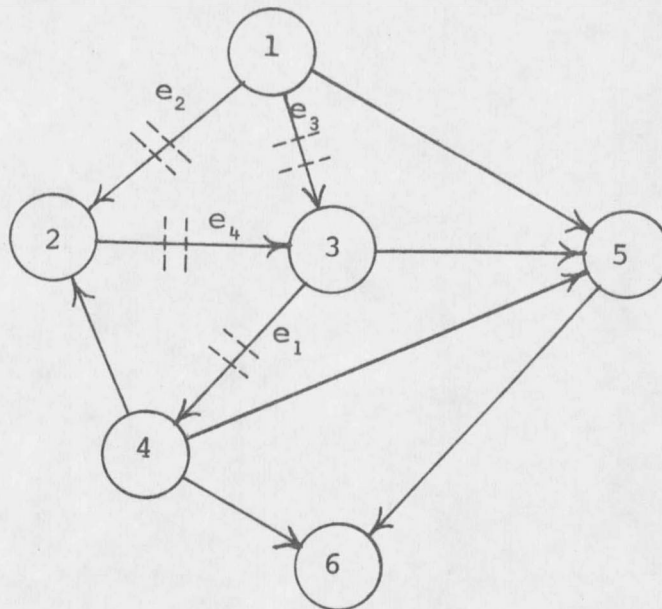


Figure 2.8 A SEC graph with blocking gates on edges e_1 , e_2 , e_3 , and e_4 .

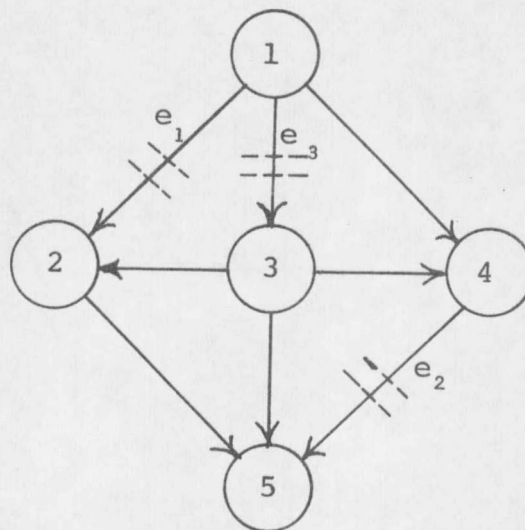


Figure 2.9 A SEC graph with blocking gates on edges e_1 , e_2 and e_3 .

Example 2.9 Consider the SEC graph of Figure 2.9. Let gate-set $E = \{e_1, e_2, e_3\}$

$$\Omega(1 \times 5)_{e_1} = \{1, 2, 3, 4, 5\}$$

$$\Omega(1 \times 5)_{e_2} = \Omega(1 \times 5)_{e_1 e_2} = \{1, 2, 3, 5\}$$

$$\Omega(1 \times 5)_{e_3} = \{1, 2, 4, 5\}$$

$$\Omega(1 \times 5)_{e_2 e_3} = \{1, 2, 5\}$$

$$\Omega(1 \times 5)_{e_1 e_3} = \{1, 4, 5\}$$

$$\Omega(1 \times 5)_{e_1 e_2 e_3} = \phi$$

The D-partition of these measurable sets consists of $\{(1, 5), (2), (3), (4)\}$. If we disregard vertex 5, each set in the D-partition contains at most one vertex. Hence, G is 1-(1,5)-distinguishable under $E = \{e_1, e_2, e_3\}$.

Using Definition 2.28, we can modify Theorem 2.6 as follows so that it will apply to blocking gates.

Theorem 2.11 Let G be a SEC graph from i to j and E be a set of edges. Also let $\{\Omega\}$ be a collection of all measurable sets produced by assigning blocking gates to all edges in E . Then, if and only if for every pair of vertices

v_r and v_s in G without considering vertex j (i.e., $v_r \neq j$ and $v_s \neq j$) there exists a measurable set such that Ω contains either v_r or v_s but not both, G is $1-(i,j)$ -distinguishable under E .

This theorem provides a necessary and sufficient condition for $1-(i,j)$ -distinguishability in the case of blocking gates.

Consider again Example 2.8. For the pair of vertices $(1,5)$, there is no measurable set $\Omega' \in \{\Omega\}$ such that Ω' contains either 1 or 5 but not both, so G is not $1-(1,6)$ -distinguishable.

In the case of test points, we find by Theorem 2.10 that a SEC graph G from i to j is 1 -distinguishable under a set E of edges if E is a directed cut-set separating i and j and if every edge of the graph is at least in one directed cut-set of $\{S\}$ covered by E . But Theorem 2.10 is not true, in general, for the blocking gate approach. A SEC graph in Figure 2.10 is 1 -distinguishable under $E = \{e_1, e_2, e_3\}$ because a collection of the directed cut-sets $\{S\}$ is covered by E and contains all edges of the graph and because E is a directed cut-set separating i and j . On the other hand, by Theorem 2.11, there exists a pair of

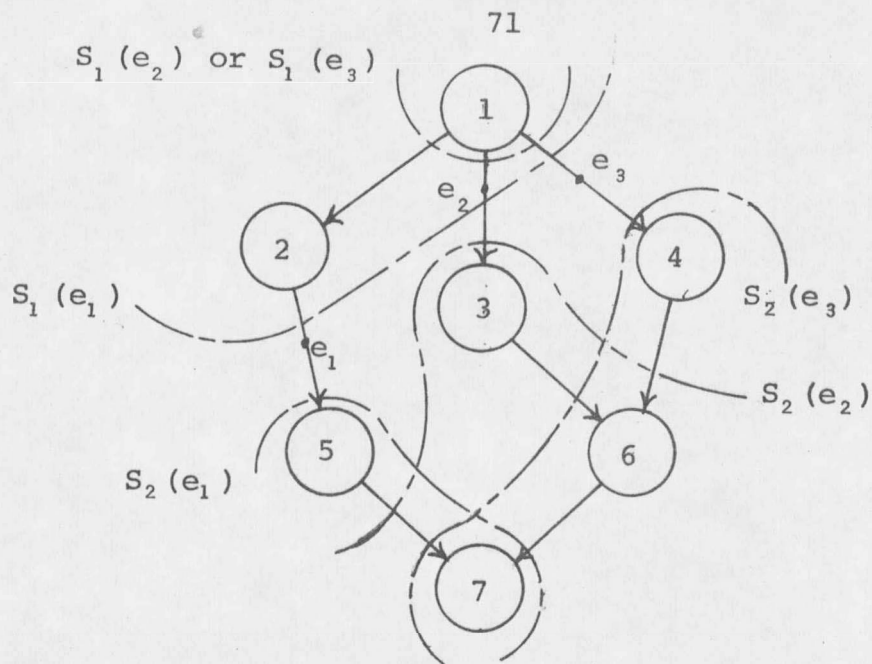


Figure 2.10 A 1-distinguishable SEC graph under an edge set $E = \{e_1, e_2, e_3\}$.

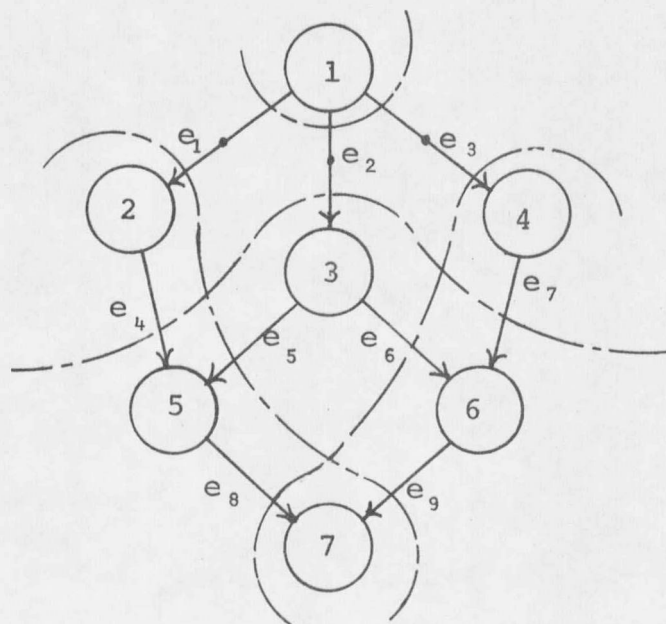


Figure 2.11 A 1-(1,7)-distinguishable SEC graph under an edge set $E = \{e_1, e_2, e_3\}$.

vertices $(2,5)$ such that either $(2,5) \subset \Omega(1 \times 7)_E^*$ or $(2,5) \subset \overline{\Omega(1 \times 7)}_E^*$ for every $E \subset E$. Hence, G is not $1-(1,7)$ -distinguishable under E . However, we can modify Theorem 2.10 for blocking gates as follows:

Theorem 2.12 Sufficient condition for $1-(i,j)$ -distinguishability: Let G be a SEC graph from i to j . Suppose a set E is a directed cut-set $\mathcal{G}(\Omega \times \overline{\Omega})$ such that either Ω consists only of vertex i or $\overline{\Omega}$ consists only of vertex j . Then G is $1-(i,j)$ -distinguishable if and only if every edge in G is in a directed cut-set in $\{S\}$ where $\{S\}$ is a collection of directed cut-sets covered by E .

Proofs of Theorems 2.11 and 2.12 are given in [30].

Example 2.10 Consider a SEC graph G from 1 to 7 as shown in Figure 2.11. Suppose $E = \{e_1, e_2, e_3\}$. Then E is a directed cut-set $\mathcal{G}(\Omega \times \overline{\Omega})$ such that Ω consists only of entry-vertex 1, and every edge in G is in a directed cut-set in $\{S\}$ where $\{S\}$ is a collection of directed cut-sets covered by E . Thus, G is $1-(1,7)$ -distinguishable.

It is not possible to obtain $1-(i,j)$ -distinguishability for all SEC graphs. For example, the SEC graph of Figure 2.5 is not $1-(i,j)$ -distinguishable,

even though we place blocking gates on all edges of the graph. A SEC graph is max-(i,j)-distinguishable if we put blocking gates on all edges. Some of these gates could be redundant. Reference [41] provides an algorithm to find a nearly minimum set of blocking gates for maximum distinguishability.

Chapter III

LOWER BOUND ON TEST POINTS FOR A SEC GRAPH.

3.1 Introduction and Preliminary Theorems

As discussed before, it is obvious that if test points are placed on every edge of a graph, without directed circuits, we can find any faulty vertex. But at the same time, this approach increases the size of the test dictionary, the number of test points to be controlled, and the cost of the system. It is useful to obtain upper and lower bounds on the number of test points necessary to make a SEC graph 1-distinguishable.

It is very easy to obtain an upper bound. Suppose a SEC graph consists of n vertices and contains no directed circuits. Then the number N of test points necessary to have 1-distinguishability need not be more than $(n-1)$. That is $N \leq (n-1)$. Similarly, for k -distinguishability, there need not be more than $(n-k)$ test points. That is $N \leq (n-k)$.

However, it is not so easy to obtain the lower bound on the number of test points necessary to make a SEC graph 1-distinguishable. Ramamoorthy and Mayeda [31] give a lower bound as follows:

Theorem 3.1 Suppose a SEC graph from i to j consists of n vertices and contains no directed circuits. Then the number N of test points necessary for 1-distinguishability can not be less than $\log_3 n$, i.e., $N \geq \log_3 n$. However, Ramamoorthy and Mayeda conjecture that there is in fact no such SEC graph that is 1-distinguishable with exactly that many test points.

This necessitates further study of graph theory. Before we derive an exact lower bound, we give other theorems that explain why the exact lower bound is not $\log_3 n$.

Definition 3.1 $G(\Omega(i \times j))$ or $G(i, j)$ is defined as the subgraph of G containing all vertices in the measurable set $\Omega(i \times j)$ and all edges connected between these vertices.

Using Definition 2.11, instead of writing $G(i, j)$, we can write $G(v(e_\ell^+), v(e_m^-))$ or $G(e_\ell^+, e_m^-)$, if $i = v(e_\ell^+)$ and $j = v(e_m^-)$.

Definition 3.2 A proper subgraph of a given graph with N_1 vertices is defined as consisting of a subset n of vertices ($n < N_1$) with all edges between those vertices retained.

Theorem 3.2 In a SEC graph G from i to j without directed circuits, consider two test points, say those on edges e_ℓ and e_m .

Suppose

$$\Omega(i \times e_\ell^+) \cap \Omega(e_m^- \times j) \neq \phi$$

Then $G(i, e_m^+)$ is a proper subgraph of $G(i, e_\ell^+)$. Similarly, if $\Omega(i \times e_m^+) \cap \Omega(e_\ell^- \times j) \neq \phi$, then $G(i, e_\ell^+)$ is a proper subgraph of $G(i, e_m^+)$.

Proof: Since $\Omega(i \times e_\ell^+) \cap \Omega(e_m^- \times j) \neq \phi$, there must be at least one vertex, say, v which is common to both the sets $\Omega(i \times e_\ell^+)$ and $\Omega(e_m^- \times j)$ as shown in Figure 3.1. Since v is in $\Omega(i \times e_\ell^+)$, by Definition 2.6 of measurable sets, there must be at least one directed path from i to v and v to $v(e_\ell^+)$. Similarly, there must be at least one directed path from $v(e_m^-)$ to v and from v to j , as v is also in $\Omega(e_m^- \times j)$. Therefore, there exists a directed path from $v(e_m^-)$ to $v(e_\ell^+)$ through vertex v .

Again using the definition of a measurable set,

$$\Omega(i \times e_\ell^+) = \Omega(i \times e_m^+) \cup v \cup v(e_m^-) \cup \Omega \quad (3.1)$$

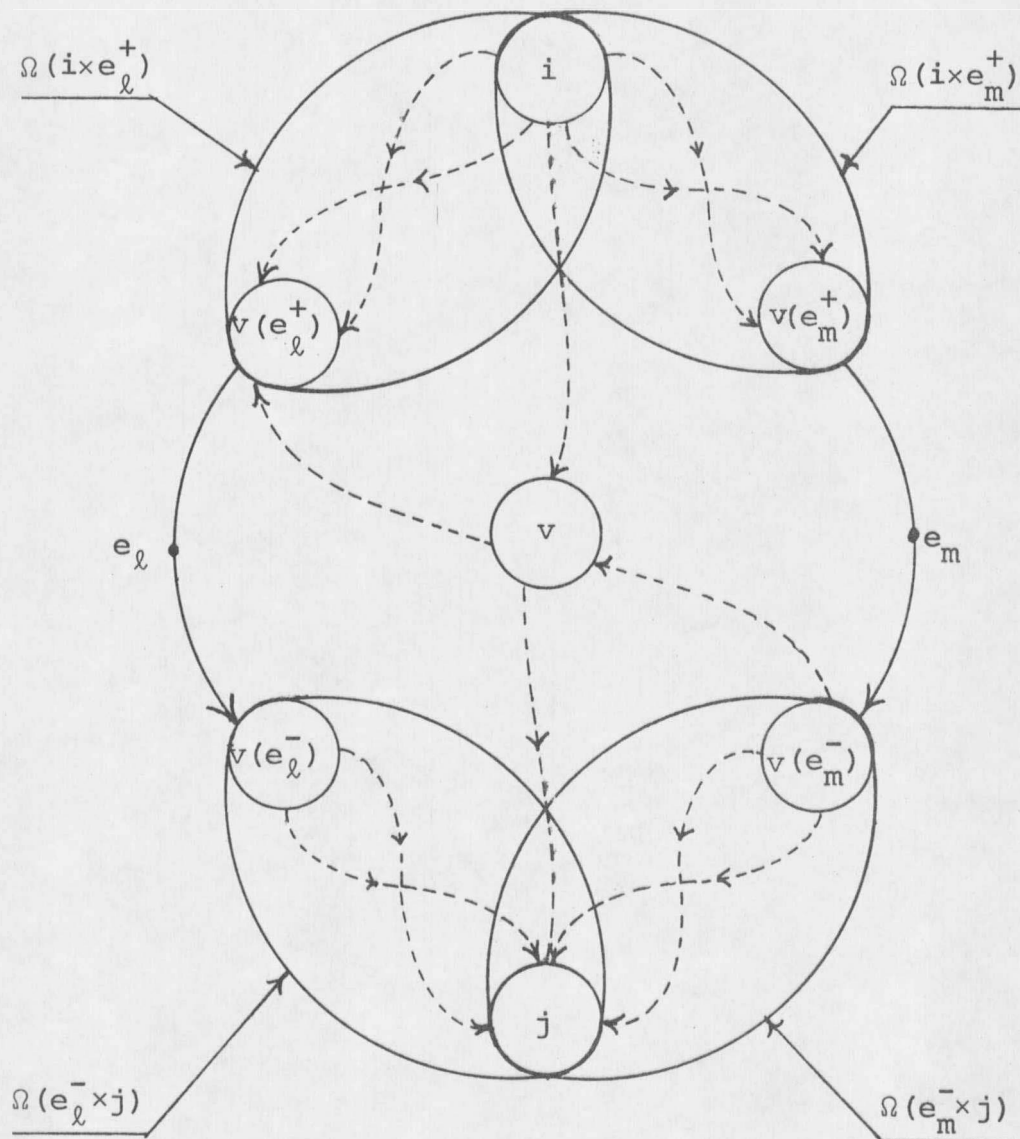


Figure 3.1 A SEC graph with two test points on edges e_l and e_m .

where Ω is a set of vertices in any other directed paths from i to $v(e_\ell^+)$, which do not pass through $v(e_m^+)$. From equation (3.1),

$$\Omega(i \times e_m^+) \subset \Omega(i \times e_\ell^+)$$

By Definition 3.2 of proper subgraph, it is obvious that $G(i, e_m^+)$ is a proper subgraph of $G(i, e_\ell^+)$.

Similarly, we can prove the second half of the theorem. It should be noted that the converse of Theorem 3.2 is not true, in general.

Example 3.1 Consider the SEC graph of Figure 3.2.

$\Omega(1 \times e_\ell^+) = \{1, 2, 3, 4\}$, $\Omega(e_m^- \times 6) = \{5, 6\}$, and therefore,

$$\Omega(1 \times e_\ell^+) \cap \Omega(e_m^- \times 6) = \phi$$

But still, $\Omega(1 \times e_m^+) \subset \Omega(1 \times e_\ell^+)$ or $G(1, e_m^+)$ is a proper subgraph of $G(1, e_\ell^+)$.

Theorem 3.3 In a SEC graph G from i to j without directed circuits, consider two test points, say those on edge e_ℓ and e_m . Suppose edge $e_m \in G(i, e_\ell^+)$, then,

$\Omega(i \times e_\ell^+) \cap \Omega(e_m^- \times j) \neq \phi$. Similarly, if edge $e_\ell \in G(i, e_m^+)$, then

$$\Omega(i \times e_m^+) \cap \Omega(e_\ell^- \times j) \neq \phi$$

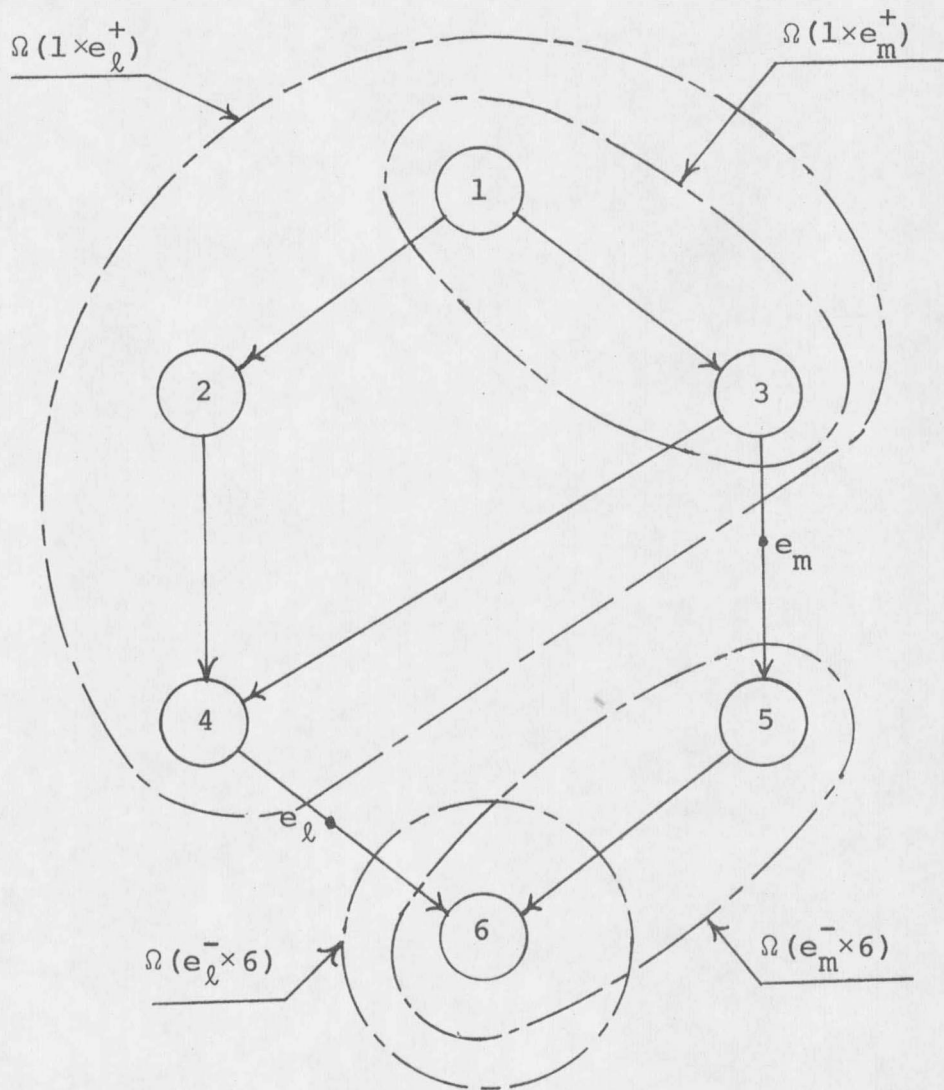


Figure 3.2 A SEC graph for Example 3.1.

Proof: $G(i, e_\ell^+)$ is a subgraph. Then by Definition 3.1, $G(i, e_\ell^+)$ contains all vertices in the measurable set $\Omega(i \times e_\ell^+)$ and all edges connected between these vertices. Since edge $e_m \in G(i, e_\ell^+)$, then vertex $v(e_m^-)$ must be in a measurable set $\Omega(i \times e_\ell^+)$. But $v(e_m^-)$ is also in the measurable set $\Omega(e_m^- \times j)$. Therefore, there exists at least one vertex $v(e_m^-)$ which is common to both sets $\Omega(i \times e_\ell^+)$ and $\Omega(e_m^- \times j)$. Hence, $\Omega(i \times e_\ell^+) \cap \Omega(e_m^- \times j) \neq \emptyset$. Similarly, we can prove the second part of the theorem.

3.2 Collections of Generators

Let us study measurable sets in detail, as they are directly related to actual measurements necessary to locate fault vertices. It is suggested in reference [30] that it is not necessary to use all possible measurable sets $\{\Omega\}$ generated by the assigned test points. Instead, a subcollection $\{\Omega'\}$ gives the same D-partition as obtained by $\{\Omega\}$.

Definition 3.3 Let $\{\Omega\}$ be a collection of all measurable sets produced by a set E . Also, let $\{\Omega'\}$ be a subcollection of $\{\Omega\}$. Then $\{\Omega'\}$ is called a collection of generators of $\{\Omega\}$ if:

1. A D-partition obtained by $\{\Omega'\}$ is the same as that formed by $\{\Omega\}$; and
2. A D-partition obtained by any proper sub-collection of $\{\Omega'\}$ is not the same as that formed by $\{\Omega\}$.

For a given $\{\Omega\}$, there may be many collections of generators. Furthermore, the number of sets in each collection of generators of $\{\Omega\}$ may be different.

Theorem 3.4 For a set $E = \{e_1, e_2, \dots, e_k\}$ of a SEC graph from i to j , a collection consisting of measurable sets $\Omega(i \times e_p^+)$ and $\Omega(e_p^- \times j)$ for $p = 1, 2, \dots, k$ contains a collection of generators of $\{\Omega\}$.

Theorem 3.5 If an edge e in a set E covers one directed cut-set, then only one of $\Omega(i \times e^+)$ and $\Omega(e^- \times j)$ can be in a collection of generators of $\{\Omega\}$.

Theorem 3.6 For a set $E = \{e_1, e_2, e_3, \dots, e_k\}$, the number of sets in a collection of generators of $\{\Omega\}$ is at most $2k$.

For the proofs of the above three theorems, refer to [30].

Example 3.2 Consider the SEC graph of Figure 3.3. Let $E = \{e_1, e_2, e_3\}$; then nonempty measurable sets in $\{\Omega\}$ are

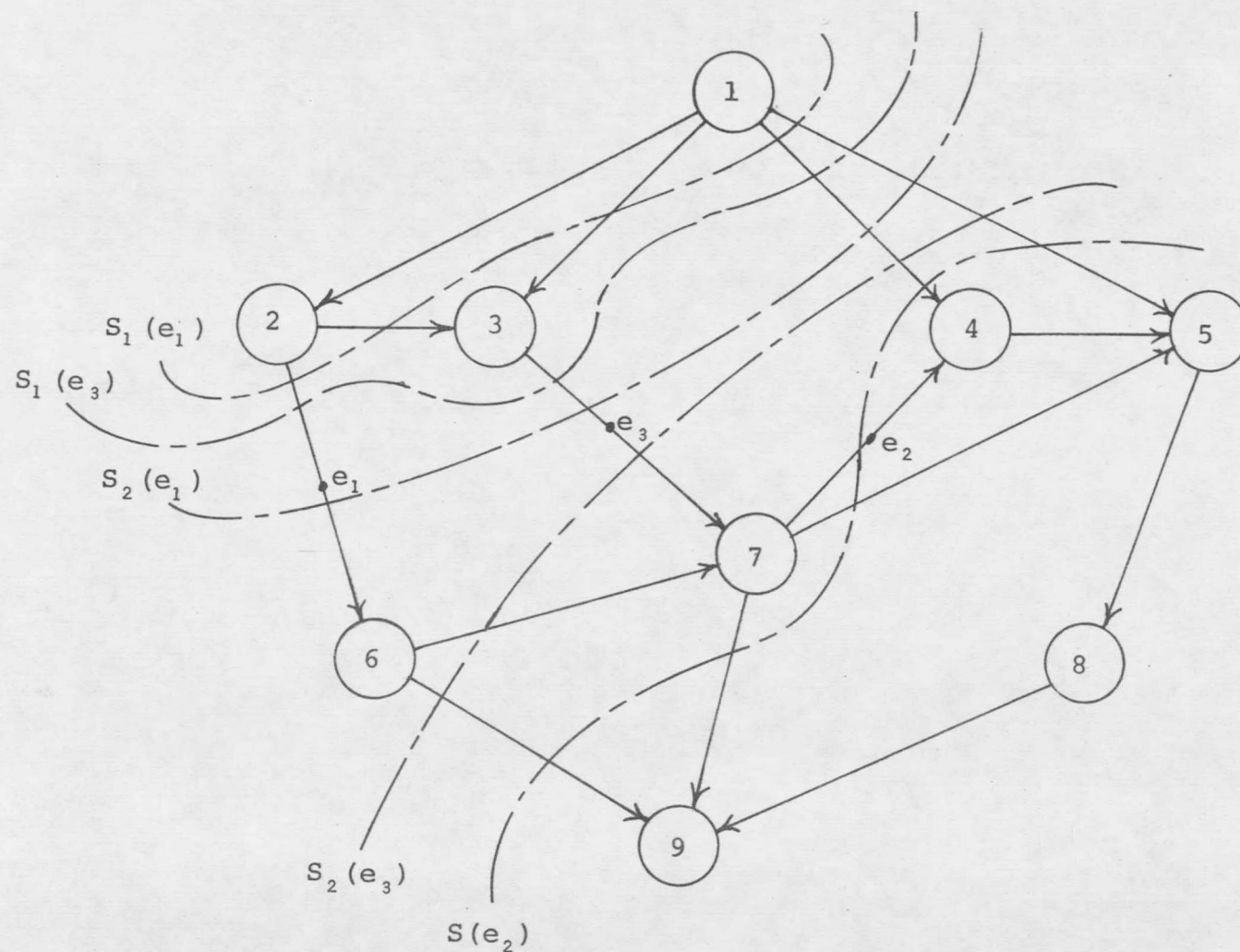


Figure 3.3 A SEC graph with directed cut-sets covered by $E = \{e_1, e_2, e_3\}$.

$\{\Omega(1 \times 9), \Omega(1 \times e_1^+), \Omega(1 \times e_2^+), \Omega(1 \times e_3^+), \Omega(e_1^- \times 9), \Omega(e_1^- \times e_2^+), \Omega(e_2^- \times 9), \Omega(e_3^- \times 9), \Omega(e_3^- \times e_2^+)\}$.

By Theorem 3.5, because edge e_3 in E covers one directed cut-set, only one of $\Omega(1 \times e_3^+)$ and $\Omega(e_3^- \times 9)$ can be in a collection of generators of $\{\Omega\}$.

A collection of generators $\{\Omega'\}$ of $\{\Omega\}$ has 5 sets.

$$\{\Omega'\} = \{(1 \times e_1^+), (e_1^- \times 9), (1 \times e_2^+), (e_2^- \times 9), (1 \times e_3^+)\}$$

3.3 D-partition and l-distinguishability

Theorem 3.7 Suppose a SEC graph from i to j without directed circuits contains N test points on the edge-set E . Then, there can be at the most 3^N nonempty sets in the D-partition of a collection of measurable sets under E .

Proof: By Definition 3.3, the D-partition obtained by a collection of generators $\{\Omega'\}$ is the same as that obtained by a collection of all possible measurable sets $\{\Omega\}$. So let us consider the D-partition obtained by $\{\Omega'\}$ instead of $\{\Omega\}$. The number of elements in $\{\Omega'\}$ is at the most $2N$ by Theorem 3.6. Suppose $E = \{e_1, e_2, \dots, e_N\}$. Then the D-partition under E can be expressed as

$$D = \bigcap_{k=1}^N \{\Omega(i \times e_k^+) \cap \Omega(e_k^- \times j)\} \quad (3.2)$$

where $\{\overline{\Omega(i \times e_k^+) \cap \Omega(e_k^- \times j)}\}$ is one of $\overline{\Omega(i \times e_k^+) \cap \Omega(e_k^- \times j)}$, $\overline{\Omega(i \times e_k^+) \cap \Omega(e_k^- \times j)}$, $\overline{\Omega(i \times e_k^+) \cap \Omega(e_k^- \times j)}$, or $\overline{\Omega(i \times e_k^+) \cap \Omega(e_k^- \times j)}$.

Let us remove empty sets from the above D-partition. There are two extreme possibilities:

Case (I) Every edge e_k of E covers only one directed cut-set. Then by Theorem 3.5, only one of $\Omega(i \times e_k^+)$ and $\Omega(e_k^- \times j)$ can be in a collection of generators

$$\text{Here,} \quad \overline{\Omega(i \times e_k^+)} = \overline{\Omega(e_k^- \times j)}$$

$$\overline{\Omega(i \times e_k^+)} = \overline{\Omega(e_k^- \times j)}$$

$$\text{Hence,} \quad \Omega(i \times e_k^+) \cap \Omega(e_k^- \times j) = \phi$$

(3.3)

$$\overline{\Omega(i \times e_k^+)} \cap \overline{\Omega(e_k^- \times j)} = \phi$$

Case (II) Every edge e_k covers two directed cut-sets (which is maximum). Here we have at most two different measurable sets $\Omega(i \times e_k^+)$ and $\Omega(e_k^- \times j)$ in $\{\Omega'\}$. Thus, by Theorem 2.4,

$$\Omega(i \times e_k^+) \cap \Omega(e_k^- \times j) = \phi \quad (3.4)$$

Comparing Cases (I) and (II), we choose Case (II) since it has the smaller number of empty sets. Removing all those empty sets, there can be at the most 3^N non-empty sets in the D-partition.

Theorem 3.8 In order to have a SEC graph of n vertices 1-distinguishable under an edge set E , at least each of $(n-1)$ vertices of the graph must be in one and only one set of the D-partitions obtained by measurable sets under E .

Proof: By Definition 2.9 on k -distinguishability, each set in a D-partition must have only one vertex for 1-distinguishability. Furthermore, any two nonempty sets in the D-partition have no vertex in common. Therefore, if any vertex is in the D-partition, it must be in one and only one set of that D-partition.

Suppose a SEC graph is 1-distinguishable with at least $(n-2)$ vertices in the D-partition, having each vertex in one and only one set. It is clear that if one of the remaining two vertices is faulty, it is impossible to distinguish which is faulty by measurements corresponding to measurable sets under E . Therefore, the graph is not

1-distinguishable. Hence, each of $(n-1)$ vertices must be in one and only one set of a D-partition to obtain 1-distinguishability.

Example 3.3 Consider the SEC graph of Figure 3.4.

Let $E = \{e_1, e_2, e_3\}$,

$$\Omega(1 \times e_1^+) = \{1, 2\}, \quad \Omega(e_1^- \times 8) = \{6, 8\}$$

$$\Omega(1 \times e_2^+) = \{1, 3\}, \quad \Omega(e_2^- \times 8) = \{6, 8\}$$

$$\Omega(1 \times e_3^+) = \{1, 4\}, \quad \Omega(e_3^- \times 8) = \{7, 8\}$$

D-partition is $\{(1), (2), (3), (4), (6), (7), (8)\}$.

Vertex 5 is not in the D-partition. The graph is 1-distinguishable under E .

3.4 Lower Bound on Test Points for 1-distinguishability of a SEC Graph

Before Theorem 3.9, two definitions are appropriate.

Definition 3.4 For every vertex v in a SEC graph, with N test points on edges $\{e_1, e_2, e_3, \dots, e_N\}$, a $2N$ -tuple $S = [s_1 s_2 s_3, \dots, s_{2N}]$ is assigned, where s_i is a binary valued variable such that

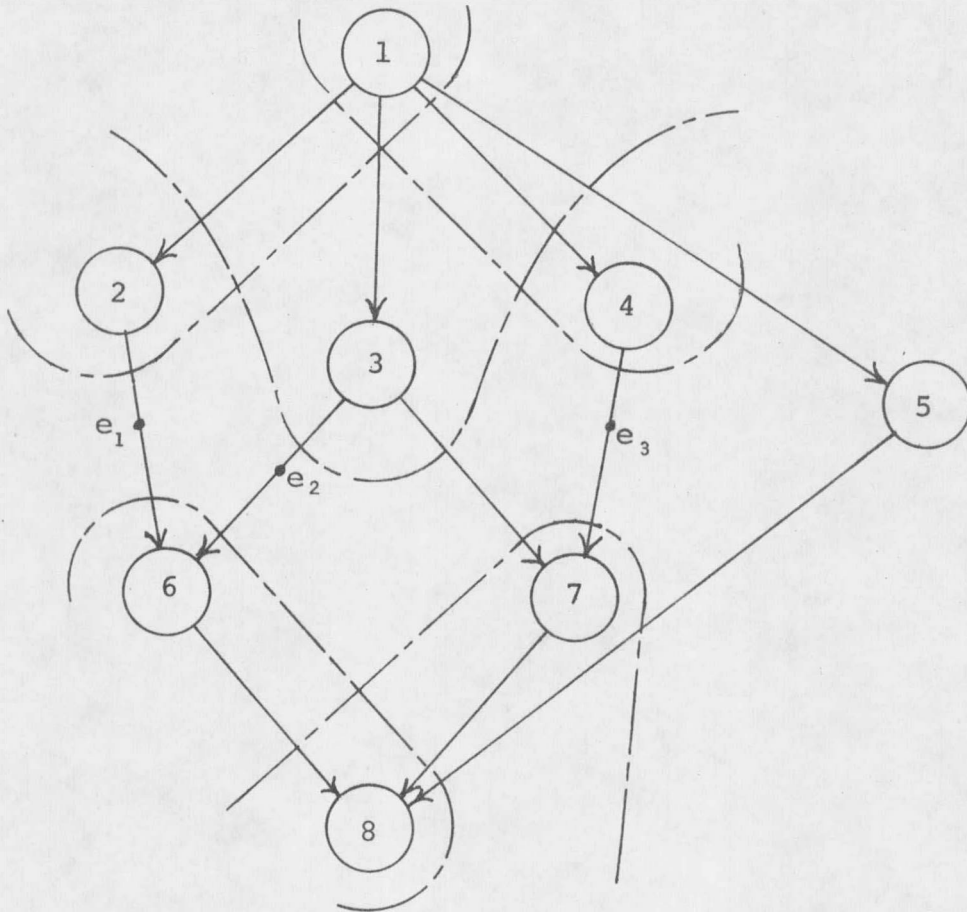


Figure 3.4 A SEC graph with directed cut-sets covered by $E = \{e_1, e_2, e_3\}$.

$$s_{2k-1} = \begin{cases} 1 & \text{if } v \in \Omega(i \times e_k^+) \\ 0 & \text{otherwise} \end{cases}$$

$$s_{2k} = \begin{cases} 1 & \text{if } v \in \Omega(e_k^- \times j) \\ 0 & \text{otherwise} \end{cases}$$

Suppose the $2N$ -tuple $S = [s_1 s_2 s_3, \dots, s_{2N}]$ is assigned to a vertex. Then s_i indicates the presence or the absence of a vertex in a particular measurable set, depending on the value assigned to s_i , which represents that measurable set. If a vertex v is not in any measurable set, the $2N$ -tuple for it is $[0000 \dots 00]$.

Consider again the SEC graph in Figure 3.4 with $E = \{e_1, e_2, e_3\}$. A $2N$ -tuple S for vertex 1 is $[101010]$. Similarly, for vertices 2, 3, 4, 5, 6, 7, and 8, we have $[100000]$, $[001000]$, $[000010]$, $[000000]$, $[010100]$, $[000001]$, and $[010101]$, respectively.

We know that if a test point is assigned on any edge e of a SEC graph from i to j ,

$$\Omega(i \times e^+) \cap \Omega(e^- \times j) = \phi$$

Hence, the $2N$ -tuple $S = [s_1 s_2 s_3 \dots s_{2N}]$ defined above must have $s_{2k-1} s_{2k} \neq 1$ for $k = 1, 2, 3, \dots, N$.

Definition 3.5* $T = \{S\}$ is defined as a set whose elements are obtained by assigning all possible values to the $2N$ -tuple $S = [s_1 s_2 s_3 \dots s_{2N}]$, where $s_i \in (0,1)$, except $s_{2k-1} \cdot s_{2k} \neq 1$ for $k = 1, 2, \dots, N$.

For example, for 2 test points, we have a 4-tuple $S = [s_1 s_2 s_3 s_4]$ and $T = \{0000, 0001, 0010, 0100, 0101, 0110, 1000, 1001, 1010\}$.

Theorem 3.9 There is no SEC graph G from i to j which is 1-distinguishable with a total number of test points $N \leq \log_3 n$, where n is a number of vertices in the graph.

Proof: Assume that such a SEC graph exists which is 1-distinguishable with a total number of vertices $n \geq 3^N$.

Case (I) $n > 3^N$: Let $n = 3^N + 1$ be the total number of vertices. By Theorem 3.8, each of $(n-1) = 3^N$ vertices must be in one and only one set of a D -partition obtained by measurable sets under an edge set E on which N test points are placed. Assume that vertex v_a is not in any set in the D -partition. There are at most 3^N nonempty sets in the D -partition by Theorem 3.7. Hence, it is possible for

* The use of S and $\{S\}$ in this chapter differs from the usage in other chapters.

each of 3^N vertices to occupy one and only one set of the D-partition.

Now, let us find corresponding measurable sets using Definition 3.4 and 3.5. It is obvious that for every vertex v , which is in the D-partition, there is exactly one $2N$ -tuple $S = [s_1 s_2 s_3 \dots s_{2N}]$. It is possible to assign a unique $2N$ -tuple to each of 3^N vertices. A collection of all these $2N$ -tuples must be the same as that of elements in a set T (Definition 3.5). But in the set T , there is a $2N$ -tuple $S_b = [000000 \dots 00]$. Let it be assigned to vertex v_b . Then, vertex v_b is not in any measurable sets under E and hence, not in the D-partition. This means that there are two vertices v_a and v_b that are not in the D-partition. Using the proof technique of Theorem 3.8, we can prove that graph G is not 1-distinguishable which contradicts the assumption.

Case (II) $n = 3^N$: Let all of the vertices of the graph be in a D-partition (this is a less rigid condition than that of Theorem 3.8) and also assume that each of them is in one and only one set of the D-partition. Similar to case (I), here also it is possible for every vertex v in the

D-partition to be assigned to exactly one $2N$ -tuple

$S = [s_1 s_2 s_3, \dots, s_{2N}]$, and finally we have a set T .

Let the $2N$ -tuples $S_\ell = [100100 \dots 00]$ and $S_m = [001000 \dots 00]$ be assigned to vertices v_ℓ and v_m , respectively. By Definition 3.4,

$$v_\ell \in \Omega(i \times e_1^+)$$

and

$$v_\ell \in \Omega(e_2^- \times j)$$

Hence,

$$\text{Also, } \Omega(i \times e_1^+) \cap \Omega(e_2^- \times j) \neq \emptyset \quad (3.5)$$

$$v_m \notin \Omega(i \times e_1^+)$$

and

$$v_m \in \Omega(i \times e_2^+)$$

Hence

$$\Omega(i \times e_2^+) \not\subset \Omega(i \times e_1^+)$$

or

$$G(i, e_2^+) \text{ is not a proper subgraph of } G(i, e_1^+). \quad (3.6)$$

The same type of results may be obtained for some other vertices too.

For the two test points on edges e_1 and e_2 , results in (3.5) and (3.6) are contradictory by Theorem 3.2, and hence contradict the assumption of Case (II).

Theorem 3.10 Suppose a SEC graph from i to j consists of n vertices and contains no directed circuits. Then the number N of test points necessary for 1-distinguishability can not be less than $\lceil \log_2(n+1) - 1 \rceil$. That is $N \geq \lceil \log_2(n+1) - 1 \rceil$.

Proof: First of all, let us examine the following two conditions:

(1) Let a test point be placed on any edge e_k of the graph. Then

$$\Omega(i \times e_k^+) \cap \Omega(e_k^- \times j) = \phi \quad (3.7)$$

(2) By Theorem 3.2, let two test points be placed on any two edges, say on edges e_ℓ and e_m of the graph and suppose

$$\Omega(i \times e_\ell^+) \cap \Omega(e_m^- \times j) \neq \phi$$

Then $G(i, e_m^+)$ is a proper subgraph of $G(i, e_\ell^+)$, or, (3.8)

$$\Omega(i \times e_m^+) \subset \Omega(i \times e_l^+).$$

Now let the $2N$ -tuple $S = [s_1 s_2 s_3 \dots s_{2N}]$ be assigned to the vertex v of the graph. According to Definition 3.4, S indicates the absence or the presence of the vertex v in all measurable sets obtained by assigning N test points in the graph. From the proof of the last Theorem 3.9, it becomes clear that we are searching for the maximum possible assignments of the above $2N$ -tuple so that corresponding measurable sets satisfy conditions (1) and (2). It is found that this can be obtained if we discard from all possible assignments of $2N$ -tuples at least those assignments which do not satisfy the following third condition:

(3) Let k and m be any even and odd ($k < 2N$, $m < 2N$) numbers, respectively. Then in each assignment of the $2N$ -tuple $S = [s_1 s_2 s_3 \dots s_{2N}]$,

$$s_k = 0 \text{ for any } s_m = 1 \quad (3.9)$$

There are at most $2(2^N - 1)$ such unique assignments of $2N$ -tuples $S = [s_1 s_2 s_3 \dots s_{2N}]$ out of 2^{2N} , which satisfy condition (3) and so in turn conditions (1) and (2) excluding assignment $[000000 \dots 00]$.

We show this by the following step-by-step argument:

1. If a test point is placed on any edge e , there are at most two measurable sets (in the collection of generators) $\Omega(i \times e^+)$ and $\Omega(e^- \times j)$.
2. Any vertex may be in the above two measurable sets or it may not be.
3. Depending on statement 2, there is a maximum of $(2^N - 1)$ possible ways of placing any vertex in the measurable sets, since the number of test points is N , excluding the case in which the vertex is not in any of the measurable sets.
4. Now for each of the above $(2^N - 1)$ possible arrangements, there are only two possible ways to place a vertex in $\Omega(i \times e^+)$ and $\Omega(e^- \times j)$, considering condition (3) of the theorem.
5. Therefore, the maximum possible number of assignments is $2(2^N - 1)$.

By assigning each of these $2(2^N - 1)$ assignments to one and only one vertex of the graph, we have a unique collection of measurable sets, and hence each and every set in a D-partition under these measurable sets contains one and only one vertex, and also the D-partition has at most $2(2^N - 1)$ nonempty sets.

Therefore, by Theorem 3.8, a graph of at most $n = 2(2^N - 1) + 1 = (2^{N+1} - 1)$ vertices is 1-distinguishable with N test points or $N \leq [\log_2(n+1) - 1]$.

Example 3.4 Let us show that there exists a SEC graph from i to j of at most 7 vertices which is 1-distinguishable with 2 test points.

Since we have two test points, say, on the edges e_ℓ and e_m , we have at most four measurable sets in a collection of generators. They are $\Omega(i \times e_\ell^+)$, $\Omega(e_\ell^- \times j)$, $\Omega(i \times e_m^+)$, and $\Omega(e_m^- \times j)$.

The problem is to place the maximum number of vertices in the above measurable sets so that these measurable sets satisfy conditions (1) and (2) of Theorem 3.10 and make the graph 1-distinguishable. This can be done very easily by assigning 0 or 1 to s_i of the $2N$ -tuple $S = [s_1 s_2 s_3 s_4]$ depending on whether a vertex is in a particular measurable set or not.

Condition (1) rules out the possibility of the following assignments to $S = [s_1 s_2 s_3 s_4]$: $\{0011, 0111, 1011, 1100, 1101, 1110, 1111\}$.

The remaining assignments are {0000, 0001, 0010, 0100, 0101, 1000, 1010, 0110, 1001}. Let us associate one and only one value to each of these assignments as follows:

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9
$\Omega(i \times e_\ell^+)$	0	0	0	0	0	1	1	0	1
$\Omega(e_\ell^- \times j)$	0	0	0	1	1	0	0	1	0
$\Omega(i \times e_m^+)$	0	0	1	0	0	0	1	1	0
$\Omega(e_m^- \times j)$	0	1	0	0	1	0	0	0	1

[Note: vertex $i = 7$ and vertex $j = 5$]

$$\text{Therefore, } \Omega(i \times e_\ell^+) = \{v_6, v_7, v_9\}$$

$$\Omega(e_\ell^- \times j) = \{v_4, v_5, v_8\}$$

$$\Omega(i \times e_m^+) = \{v_3, v_7, v_8\}$$

$$\Omega(e_m^- \times j) = \{v_2, v_5, v_9\}$$

$$\text{Now } \Omega(i \times e_\ell^+) \cap \Omega(e_m^- \times j) = v_9 \neq \phi,$$

$$\text{and } \Omega(i \times e_m^+) \not\subset \Omega(i \times e_\ell^+).$$

Hence, even though the above measurable sets satisfy condition (1), they do not satisfy condition (2) of Theorem 3.10. Therefore, there is no SEC graph of 9 vertices which is 1-distinguishable with 2 test points.

After removing assignments [0110] and [1001] (so that the remaining assignments satisfy condition (2)), we have the following collection of measurable sets:

	v_1	v_2	v_3	v_4	v_5	v_6	v_7
$\Omega(i \times e_\ell^+)$	0	0	0	0	0	1	1
$\Omega(e_\ell^- \times j)$	0	0	0	1	1	0	0
$\Omega(i \times e_m^+)$	0	0	1	0	0	0	1
$\Omega(e_m^- \times j)$	0	1	0	0	1	0	0

Therefore,

$$\Omega(i \times e_\ell^+) = \{v_6, v_7\}$$

$$\Omega(e_\ell^- \times j) = \{v_4, v_5\}$$

$$\Omega(i \times e_m^+) = \{v_3, v_7\}$$

$$\Omega(e_m^- \times j) = \{v_2, v_5\}$$

One possible SEC graph with the above measurable sets which is 1-distinguishable with 2 test points and 7 vertices is shown in Figure 3.5.

Theorem 3.10 also agrees with

$$N \geq \log_2(n+1) - 1 \Big|_{n=7} = N \geq \log_2(7+1) - 1$$

$$N \geq 2$$

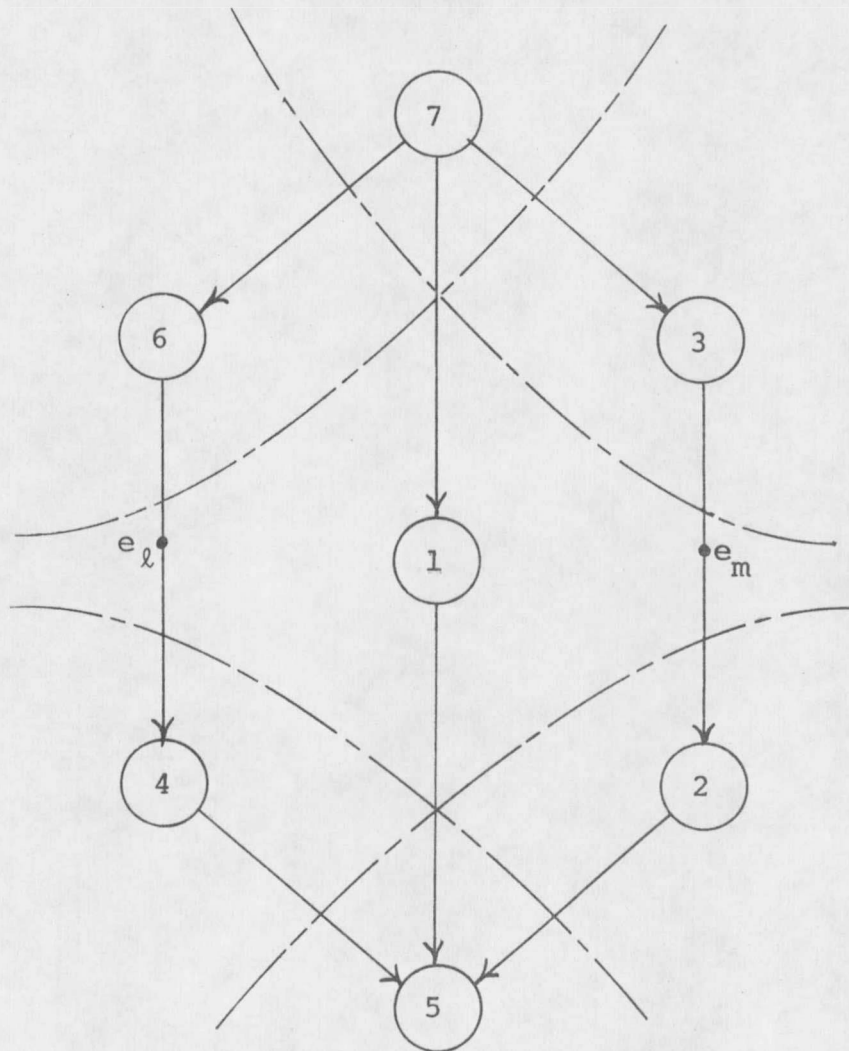


Figure 3.5 A 1-distinguishable SEC graph of 7 vertices with 2 test points.

Theorem 3.11 For a SEC graph from i to j consisting of n vertices and containing no directed circuits, the number N of test points for k -distinguishability is bounded by

$$N \geq \lceil \log_2 (n+k) - k \rceil$$

The proof of this theorem is similar to that of Theorem 3.10. It is a generalization for a lower bound on the number of test points for k -distinguishability.

It is found that there exists one and only one collection of measurable sets which makes a SEC graph 1-distinguishable with N -test points and $n = 2^{N+1} - 1$ vertices. For a 3 test point case $N = 3$, such a collection of measurable sets is

$$\Omega(1 \times e_p^+) = \{1, 2, 3, 5\}$$

$$\Omega(e_p^- \times 15) = \{9, 12, 13, 15\}$$

$$\Omega(1 \times e_q^+) = \{1, 2, 4, 6\}$$

$$\Omega(e_q^- \times 15) = \{10, 12, 14, 15\}$$

$$\Omega(1 \times e_r^+) = \{1, 3, 4, 7\}$$

$$\Omega(e_r^- \times 15) = \{11, 13, 14, 15\}$$

One possible 1-distinguishable SEC graph of 15 vertices with the above measurable sets is shown in Figure 3.6.

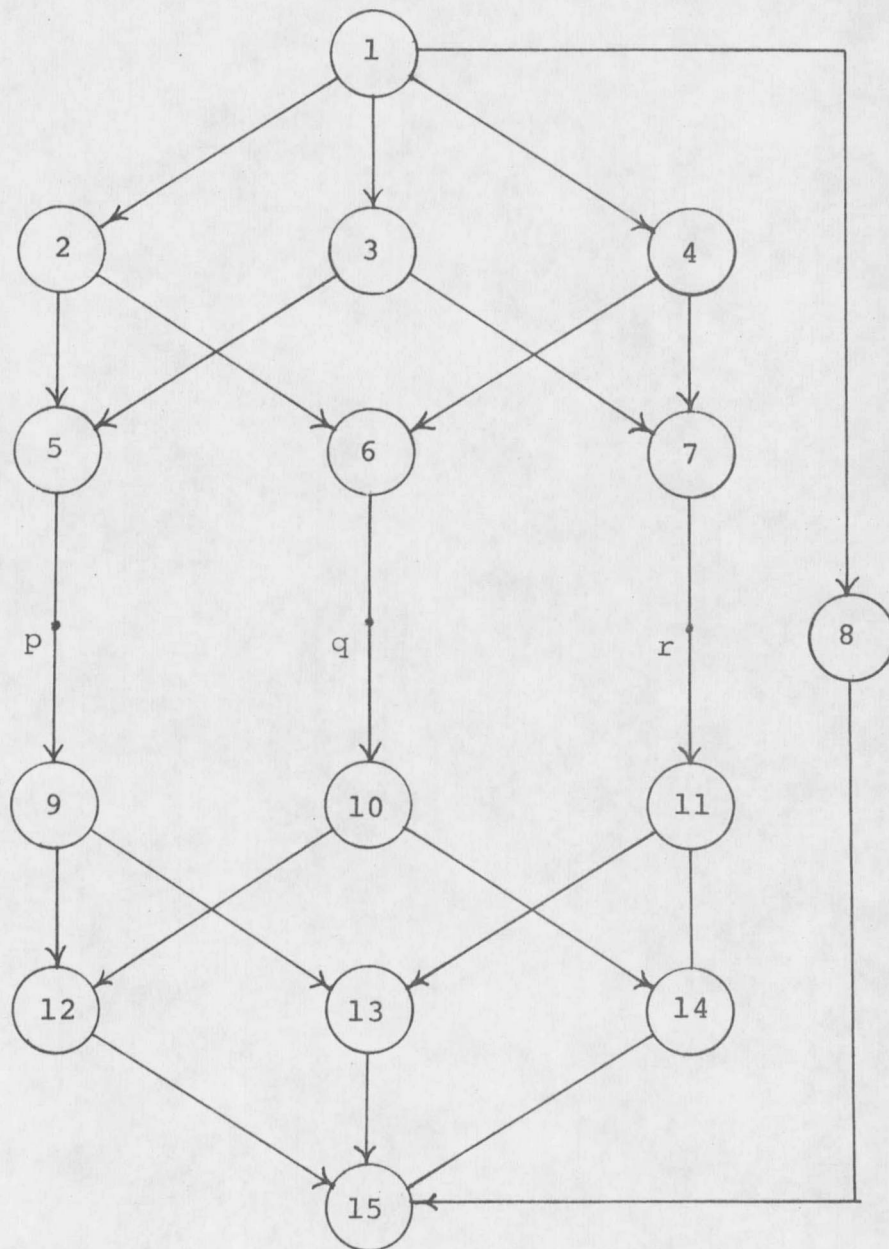


Figure 3.6 A 1-distinguishable SEC graph of 15 vertices with 3 test points.

Chapter IV

AN ALGORITHM FOR THE MINIMUM NUMBER OF TEST POINTS

4.1 Introduction

In this chapter, we develop an efficient algorithm for inserting test points in a SEC graph to obtain 1-distinguishability. Or, in other words, we improve structural resolution to the extent that we find exactly which module or submodule represented as a vertex of a graph is faulty. As discussed before, 1-distinguishability could be readily obtained by placing test points on every edge of a graph, but our aim is to do the same with the minimum number of test points.

4.2 Necessity for an Algorithm

In Chapter II, we examined six basic theorems on test points. Theorems 2.6, 2.7, and 2.8 provide necessary and sufficient conditions for 1-distinguishability, while Theorem 2.5 gives only necessary conditions and Theorems 2.9 and 2.10 give only sufficient conditions. If the system graph and the location of test points on it are given, we can directly check, using these theorems, whether or not a given graph with given test points is 1-distinguishable. But in

general, our problem is the reverse; that is, we want to determine the location of test points to make the given graph 1-distinguishable. For this purpose, the direct application of the above theorems is difficult and leads to lengthy computation.

For example, the direct application of a D-partition requires an arbitrary selection of an edge-set on which test points should be inserted and then a check to see whether the graph is 1-distinguishable under it. Hence, it is exhaustive.

Theorem 2.8 transfers the problem of finding the location of test points for 1-distinguishability to that of finding a collection of $(n-1)$ [n = number of vertices in the graph] linearly independent directed cut-sets with all edges of the graph in it and covering these directed cut-sets with a set E of edges. There is an efficient algorithm to order the vertices of a graph; once ordered it is easy to find $(n-1)$ linearly independent directed cut-sets. But unfortunately, it is applicable to a SEC graph which has a unique ordering of vertices, hence, the approach is limited in application.

Theorem 2.5 solves the same problem by searching edge-set E whose directed cut-sets cover all edges of the graph. But Theorem 2.5 is only a necessary condition.

Theorems 2.6 and 2.7 are similar. They provide necessary and sufficient conditions. In Theorem 2.6 we need an edge-set E whose collection of measurable sets $\{\Omega\}$ is such that for every pair of vertices v_r and v_s , there must be at least one Ω such that either v_r or v_s must be in it, but not both. In Theorem 2.7, directed cut-sets covered by E must separate every pair of vertices of the graph. For a large graph of, say, 100 vertices, we have to check $\binom{100}{2}$ or approximately 5000 pairs of vertices; this is too lengthy.

Theorem 2.10 relieves the problem somewhat; we search for an edge-set E which must be a directed cut-set separating an entry and an exit vertex in addition to satisfying the conditions of Theorem 2.5. Theorem 2.10, however, is only a sufficient condition. In addition to this, the number of test points may not be minimum. As a matter of fact, direct application of any of the above theorems does not show how we can find the minimum number of test points for 1-distinguishability. Hence, an algorithm should be developed with the following attributes:

1. It must achieve a reduction in computation to gain efficiency.
2. It must be general enough to apply to a large range of system graphs.
3. It must be easily implemented on a computer.

As it is easy to find directed cut-sets of a graph, a satisfactory solution may be obtained by considering Theorems 2.5 and 2.7 together. In Theorem 2.5, we have to find a collection of directed cut-sets $\{S\}_{E_1}$ covered by a minimum edge-set E_1 which contains all edges of the graph. Naturally, checking every edge instead of every pair of vertices is far less exhaustive. As this theorem provides only a necessary condition, sufficiency can be achieved by finding a pair of vertices which are not separated by $\{S\}_{E_1}$. This may necessitate more directed cut-sets $\{S\}_{E_2}$ that separate the remaining pair of vertices, and hence to cover these $\{S\}_{E_2}$ requires more edges E_2 of the graph. The number of test points may increase, but they must be added to make to graph 1-distinguishable. For certain graphs, there is a possibility of the separation of all pairs of vertices of the graph $\{S\}_{E_1}$ only. The above approach immediately raises the problem of finding the pairs of vertices which are not separated by

directed cut-sets $\{S\}_{E_1}$. This search method must be simple enough to make the algorithm efficient. As will be seen later in this chapter, the method suggested reduces the computation, compared to that required in an all-pairs-of-vertices approach, and the number of test points obtained is the minimum.

The main problems in the development of an algorithm to find the location of test points for 1-distinguishability of a SEC graph, are summarized as follows:

1. We require an efficient way to find directed cut-sets that are covered by edge e from a set of directed cut-sets $\{S(e)\}$, all of which contain edge e and separate the entry from the exit vertex.
2. We must find a collection V of pairs of vertices, whose separation is not guaranteed by a directed cut-set of $\{S\}_{E_1}$. Here $\{S\}_{E_1}$ contains all edges of the graph.
3. We require a method of searching for the edge-set E with the minimum number of edges so that directed cut-sets covered by E contain all edges of the graph and also separate every pair of vertices in a set V .

In the rest of the chapter, we will discuss step-by-step a solution of the above problems and finally give an algorithm for placing the minimum number of test points on a SEC graph for 1-distinguishability.

4.3 Matrix Representation of a SEC Graph

Generally, system graphs can be represented as matrices whose elements take binary values for ease of manipulation on a computer. Many properties of a graph can be studied by manipulations on a connectivity matrix of the graph. Hence, let us first of all define this and other matrices used in the algorithm.

Definition 4.1 Connectivity matrix $C = \{c_{ij}\}$ is an $(n \times n)$ matrix that characterizes a graph G with a vertex set $\Omega = \{1, 2, 3, \dots, n\}$ and an edge-set $E = \{e_1, e_2, e_3, \dots, e_m\}$, such that

$$c_{ij} = 1 \begin{cases} \text{If and only if there is a directed} \\ \text{edge from vertex } i \text{ to vertex } j \end{cases} \\ = 0 \quad \text{otherwise}$$

Definition 4.2 Reachability matrix $R = \{r_{ij}\}$ is an $(n \times n)$ matrix for graph G with a vertex set $\Omega = \{1, 2, 3, \dots, n\}$ and an edge-set $E = \{e_1, e_2, e_3, \dots, e_m\}$, such that

$$r_{ij} = 1 \begin{cases} \text{If and only if there is at least one} \\ \text{directed path (i.e., a concatenation} \\ \text{of directed edges) leading from } i \\ \text{to } j \end{cases} \\ = 0 \quad \text{otherwise}$$

It is possible to generate matrix R from matrix C [21,37,43,44,45]. Before we show the equations, let us consider the following matrix:

Definition 4.3 Matrix $C^k = \{c_{ij}^k\}$ is an $(n \times n)$ matrix for a graph G with a vertex-set $\Omega = \{1, 2, 3, \dots, n\}$ and an edge-set $E = \{e_1, e_2, e_3, \dots, e_m\}$, such that

$$c_{ij}^k = \begin{cases} 1 & \text{If and only if there exists at least} \\ & \text{one directed path (i.e., a concatenation} \\ & \text{of directed edges) of the length } k \text{ from} \\ & \text{ } i \text{ to } j \\ 0 & \text{otherwise} \end{cases}$$

Theorem 4.1 Matrix $C^k = C^{k-1} \cdot C = C \cdot C^{k-1} = \{c_{ij}^k\}$

where

$$c_{ij}^k = \sum_{p=1}^n c_{ip}^{k-1} c_{pj}$$

Theorem 4.2 [26,47] There exists an integer k such that for all x ($k \leq x \leq n$)

$$\sum_{i=1}^k C^i = \sum_{i=1}^x C^i \quad \text{and} \quad R = \sum_{i=1}^k C^i + I$$

(Here, I is the identity matrix, and n is the number of vertices of a graph).

In Theorems 4.1 and 4.2, the summations and multiplications are Boolean operations of OR and AND, respectively. For the proof of both theorems, refer to [21,26,47]. These theorems are useful for the generation of matrix R from matrix C.

Example 4.1 The connectivity matrix C and the reachability matrix R for the graph in Figure 2.11 are as follows:

$$C = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{matrix} & \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{matrix}$$

$$C^2 = C^1 \cdot C^1 =$$

	1	2	3	4	5	6	7
1	0	0	0	0	1	1	0
2	0	0	0	0	0	0	1
3	0	0	0	0	0	0	1
4	0	0	0	0	0	0	1
5	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0

$$C^3 = C^1 \cdot C^2 =$$

	1	2	3	4	5	6	7
1	0	0	0	0	0	0	1
2	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0

$$C^4 = C^3 \cdot C^1 = [0]$$

Hence,

$$R = \sum_{i=1}^4 C^i + I$$

	1	2	3	4	5	6	7
1	1	1	1	1	1	1	1
2	0	1	0	0	1	0	1
3	0	0	1	0	1	1	1
R = 4	0	0	0	1	0	1	1
5	0	0	0	0	1	0	1
6	0	0	0	0	0	1	1
7	0	0	0	0	0	0	1

4.4 Development of an Algorithm

4.4.1 Directed cut-sets covered by an edge According to

Definition 2.18, in order to find directed cut-sets

$S_1(e) = \mathcal{G}[\Omega_1 \times \bar{\Omega}_1]$ and $S_2(e) = \mathcal{G}[\Omega_2 \times \bar{\Omega}_2]$ covered by an edge

e , we have to first find all possible directed cut-sets

$\{S_k(e)\}$, each of which contains edge e and also separates

the entry and exit vertices, and then check one by one to

determine which one of these satisfies condition (3) of

Definition 2.18, i.e.

$$(\Omega_1 \cup \Omega_2) \supset \Omega_k \supset (\Omega_1 \cap \Omega_2)$$

where

$$S_k = \mathcal{G}[\Omega_k \times \bar{\Omega}_k]$$

Much computation is required in the above approach. The following theorem helps to find directed cut-sets covered by any edge.

Theorem 4.3 Let G be a SEC graph from i to j .

$$(1) \text{ If, } \overline{\Omega(i \times e^+)} \neq \Omega(e^- \times j)$$

$$\text{then, } S_1(e) = \xi[\Omega(i \times e^+) \times \overline{\Omega(i \times e^+)}]$$

$$\text{and, } S_2(e) = \xi[\overline{\Omega(e^- \times j)} \times \Omega(e^- \times j)]$$

are directed cut-sets covered by edge e .

$$(2) \text{ If, } \overline{\Omega(i \times e^+)} = \Omega(e^- \times j)$$

then, $S_1(e) = S_2(e) = S(e)$ is a directed cut-set covered by edge e .

Proof:

Case (I): Conditions (1) and (2) of Definition 2.18 are satisfied by $S_1(e)$ and $S_2(e)$. Hence, we have to prove only that for any other directed cut-set $S_k(e) = \xi[\Omega_k \times \overline{\Omega_k}]$, which separates i and j and contains edge e , condition (3) is satisfied, i.e.,

$$[\Omega(i \times e^+) \cup \overline{\Omega(e^- \times j)}] \supset \Omega_k \supset [\Omega(i \times e^+) \cap \overline{\Omega(e^- \times j)}] \quad (4.1)$$

Since $S_1(e)$ and $S_2(e)$ are directed sut-sets that contain edge e , by Theorem 2.3, $\Omega(i \times e^+) \cap \Omega(e^- \times j) = \phi$.

Also,

$$\overline{\Omega(e^- \times j)} = \Omega - \Omega(e^- \times j) \quad [\text{Definition 2.7}]$$

Hence,

$$\Omega(i \times e^+) \subseteq \overline{\Omega(e^- \times j)}$$

Therefore,

$$\Omega(i \times e^+) \cup \overline{\Omega(e^- \times j)} = \overline{\Omega(e^- \times j)} \quad (4.2)$$

Similarly,

$$\Omega(i \times e^+) \cap \overline{\Omega(e^- \times j)} = \Omega(i \times e^+)$$

Using results (4.2) and (4.3), condition (3) can be restated as,

$$\overline{\Omega(e^- \times j)} \supset \Omega_k \supset \Omega(i \times e^+) \quad (4.4)$$

Or, in other words, we have to prove that a set $\Omega(i \times e^+)$ is a proper subset of Ω_k and also we have to prove that Ω_k is a proper subset of $\overline{\Omega(e^- \times j)}$ for any other directed cut-set $S_k(e) = \mathcal{E}[\Omega_k \times \overline{\Omega_k}]$ which contains edge e and separates i and j .

Let us assume that there exists a directed cut-set $S_m(e) = \xi[\Omega_m \times \bar{\Omega}_m]$ which contains the edge e and separates i and j such that $\Omega_m \subset \Omega(ixe^+)$. Then there is at least one vertex v , where $v \in \Omega(ixe^+)$ and $v \notin \Omega_m$ as shown in Figure 4.1.

Since $v \in \Omega(ixe^+)$, there is a directed path from i to v and v to $v(e^+)$. Hence, there is no directed cut-set $S_m(e) = \xi[\Omega_m \times \bar{\Omega}_m]$ which separates i and j and also contains the edge e , which contradicts the assumption.

Similarly, we can prove that any Ω_k is a proper subset of set $\Omega(e^- \times j)$. This proves Case (I).

Case (II): Here, $\overline{\Omega(ixe^+)} = \Omega(e^- \times j)$

$$\begin{aligned} S_1(e) &= \xi[\Omega(ixe^+) \times \overline{\Omega(ixe^+)}] \\ &= \xi[\Omega(ixe^+) \times \Omega(e^- \times j)] \end{aligned}$$

Now,

$$\begin{aligned} \overline{\Omega(e^- \times j)} &= \Omega - \Omega(e^- \times j) \\ &= \Omega - \overline{\Omega(ixe^+)} \\ &= \Omega(ixe^+) \end{aligned}$$

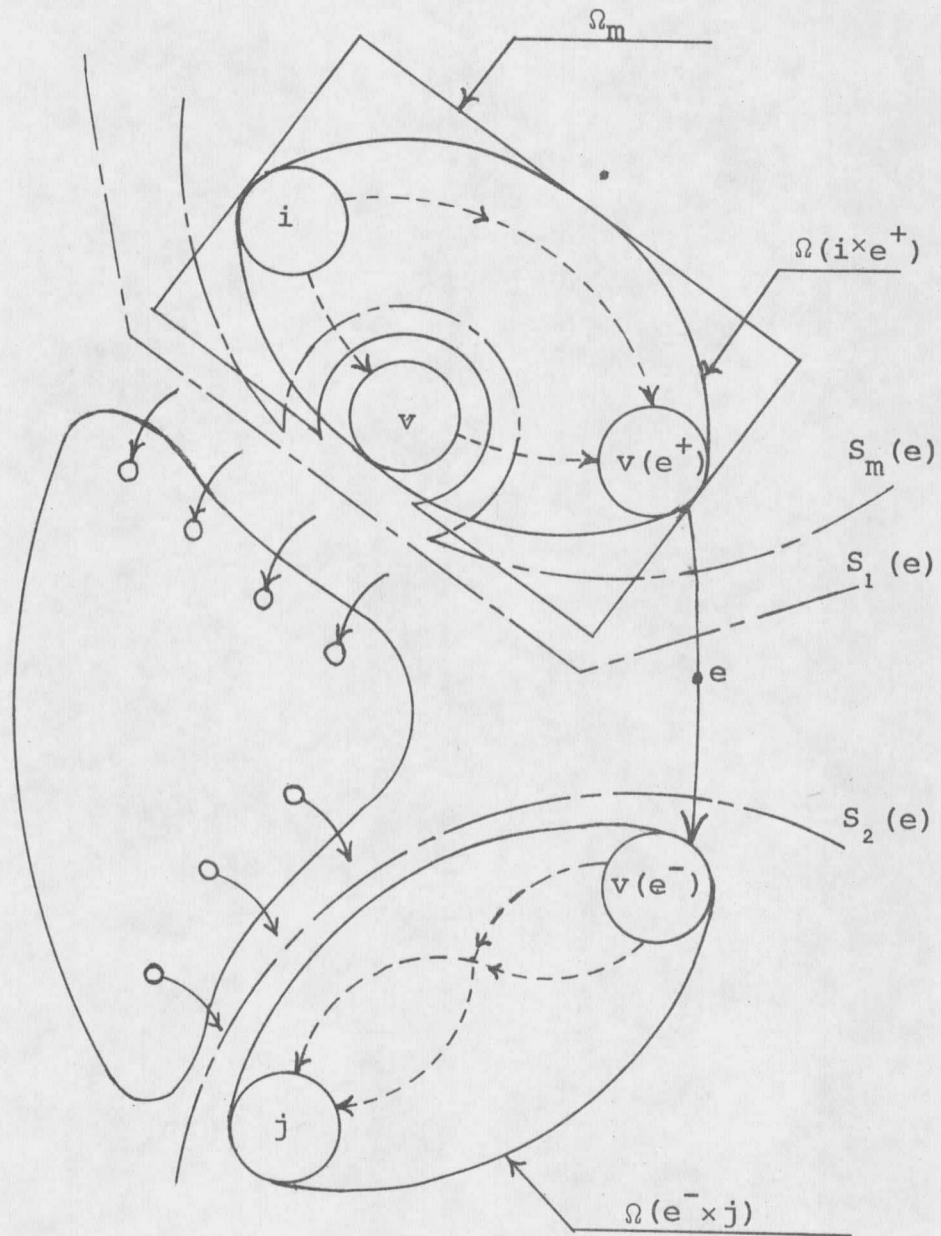


Figure 4.1 A SEC graph with directed cut-sets covered by an edge e .

$$\begin{aligned}
S_2(e) &= \xi[\Omega(e^- \times j) \times \Omega(e^- \times j)] \\
&= \xi[\Omega(i \times e^+) \times \Omega(e^- \times j)] \\
&= S_1(e) = S(e)
\end{aligned}$$

For an illustration of Theorem 4.3, see Example 2.2 of Chapter II.

It is very easy to check, using the reachability matrix R , whether there is a directed path from any vertex α to $v(e^+)$. Similarly, from $v(e^-)$ to α , the connectivity matrix C provides information on the connection of edges between various vertices and on the direction. Hence, the use of Theorem 4.3, along with matrices C and R , makes the search for directed cut-sets covered by any edge e more systematic and computationally efficient as follows:

$$\text{A vertex } \alpha \in \Omega(i \times v(e^+))$$

if and only if

$$r_{\alpha, v(e^+)} = 1$$

Similarly,

$$\alpha \in \Omega(v(e^-) \times j)$$

if and only if

$$r_{v(e^-), \alpha} = 1$$

Here,

$r_{i, \alpha} = 1$ and $r_{\alpha, j} = 1$, because the graph is a SEC graph.

Checking all vertices, we will have measurable sets $\Omega(i \times v(e^+))$ and $\Omega(v(e^-) \times j)$. Then by the use of the connectivity matrix C and Theorem 4.3, directed cut-sets $S_1(e)$ and $S_2(e)$ covered by edge e are obtained.

$$e_k \in S_1(e)$$

if and only if $c_{v(e_k^+), v(e_k^-)} \neq 0$, for $v(e_k^+) \in \Omega(i \times v(e^+))$ and $v(e_k^-) \notin \Omega(i \times v(e^+))$. Also, $e_k \in S_2(e)$ if and only if $c_{v(e_k^+), v(e_k^-)} \neq 0$, for $v(e_k^+) \notin \Omega(v(e^-) \times j)$ and $v(e_k^-) \in \Omega(v(e^-) \times j)$.

Example 4.2 Consider again the SEC graph of Figure 2.11 and matrices C and R of Example 4.1. We want to find directed cut-sets $S_1(e_3)$ and $S_2(e_3)$ covered by an edge e_3 .

1. $v(e_3^+) = 1$ and $v(e_3^-) = 4$.

2. Measurable sets:

(a) Column 1 of matrix R has a '1' entry at row 1 only. Hence, $\Omega(1 \times 1) = \{1\}$.

(b) Row 4 of matrix R has a '1' entry at columns 4, 6, and 7. Hence, $\Omega(4 \times 7) = \{4, 6, 7\}$.

3. Directed cut-sets $S_1(e_3)$ and $S_2(e_3)$:

(a) Row 1 of matrix C has '1' entries at columns 2, 3, and 4. Hence, edges e_1 , e_2 , and e_3 from vertex 1 to 2, 1 to 3, and 1 to 4, respectively, are in $S_1(e_3)$.

(b) Column 4 of matrix C has a '1' entry at row 1 only. Hence, an edge e_3 from vertex 1 to 4 is in $S_2(e_3)$.

Column 6 of matrix C has '1' entries at rows 3 and 4. But vertex $4 \in \Omega(4 \times 7)$. Discarding vertex 4, an edge e_6 from vertex 3 to 6 is in $S_2(e_3)$.

Similarly, checking '1' entries in column 7 of matrix C, $e_8 \in S_2(e_3)$.

4. $S_1(e_3) = \{e_1, e_2, e_3\}$ and $S_2(e_3) = \{e_3, e_6, e_8\}$.

4.4.2 A set V of non-separating pairs of vertices Consider first the following theorem:

Theorem 4.4 Suppose there is a directed path (i.e., concatenation of directed edges) from a vertex v_p to another vertex v_q of a SEC graph from i to j . Then this pair of vertices (v_p, v_q) is separated by at least one of a collection of directed cut-sets $\{S\}$ which contains all edges of the graph.

Proof: The proof is obvious if there is only one edge in a directed path from v_p to v_q . So let us consider the case in which there is more than one edge in a directed path $P = \{e_1, e_2, e_3, \dots, e_{k-1}, e_k, \dots, e_{\ell-1}, e_\ell, \dots, e_m, e_{m+1}\}$ from v_p to v_q as shown in Figure 4.2.

As stated in the theorem, each of the edges in path P must be in at least one of the directed cut-sets of $\{S\}$. Let there be a directed cut-set $S = \mathcal{G}[\Omega_k \times \bar{\Omega}_k]$ of $\{S\}$ which contains edge e_k . This directed cut-set will separate v_p and v_q ; or

$$(v_p, \alpha_1, \alpha_2, \dots, \alpha_{k-1}) \in \Omega_k$$

and

$$(\alpha_k, \alpha_{k+1}, \dots, \alpha_{m-1}, \alpha_m, v_q) \in \bar{\Omega}_k.$$

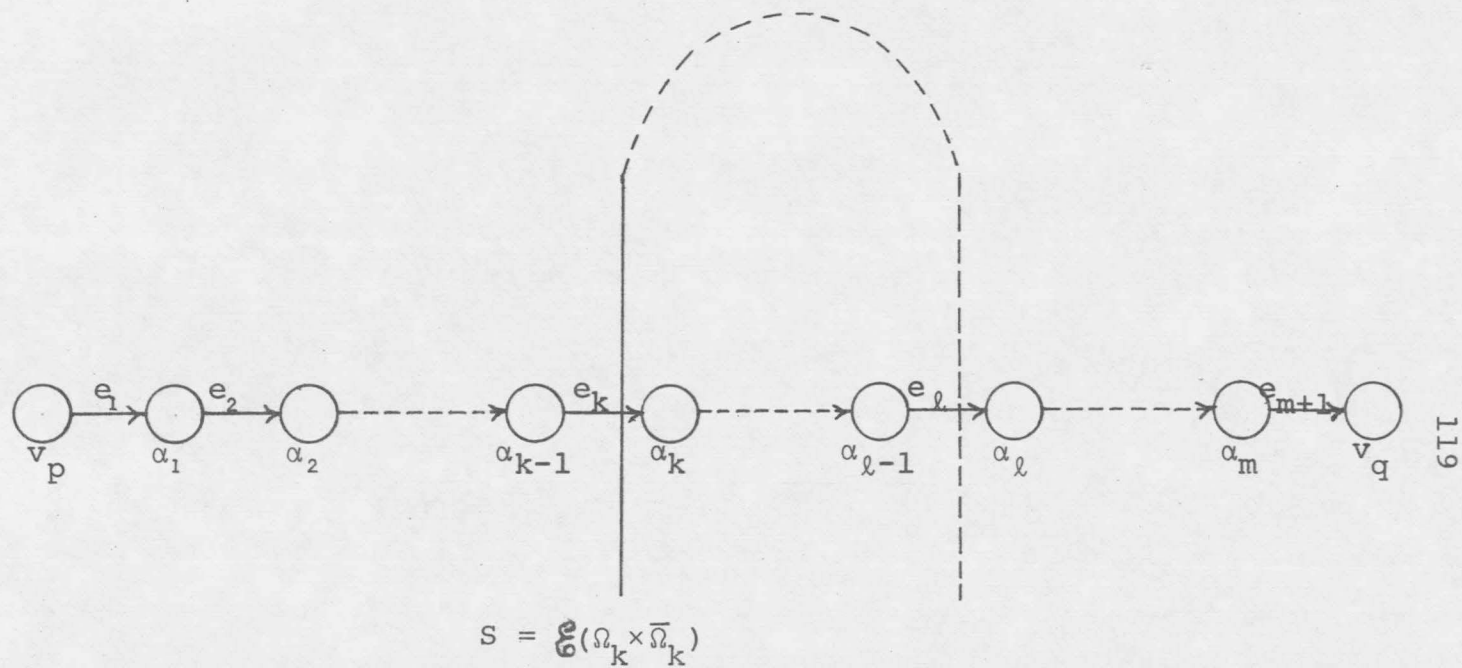


Figure 4.2 A directed path from v_p to v_q with directed cut-set $\mathcal{E}(\Omega_k \times \bar{\Omega}_k)$.

Suppose the theorem statement is not true, i.e.,

$(v_p, v_q) \in \Omega_k$ or $(v_p, v_q) \in \bar{\Omega}_k$. This is possible if and only if this directed cut-set $\{S\}$ contains at least one other edge e_ℓ of path P . But then S is no longer a directed cut-set, which contradicts the assumption.

In fact, any directed cut-set S of $\{S\}$, which contains any edge of path P , can separate the pair of vertices v_p and v_q .

It should be remembered that the converse of Theorem 4.4 is not true, in general. That is, suppose there is no directed path from a vertex v_p to another vertex v_q of a SEC graph from i to j . Even then, this pair of vertices may be separated by at least one of a collection of directed cut-sets $\{S\}$ which contains all edges of the graph.

Example 4.3 Consider the SEC graph of Figure 4.3.

$\{S\} = \{S_1, S_2, S_3, S_4, S_5, S_6\}$ is a collection of directed cut-sets which contains all edges of the graph. There is no directed path from vertex 2 to 4 (or 4 to 2). Similarly, for pairs of vertices (2,7) and (3,7) no directed path exists. Therefore, according to Theorem 4.4, the separation of all pairs of vertices except (2,4), (2,7),

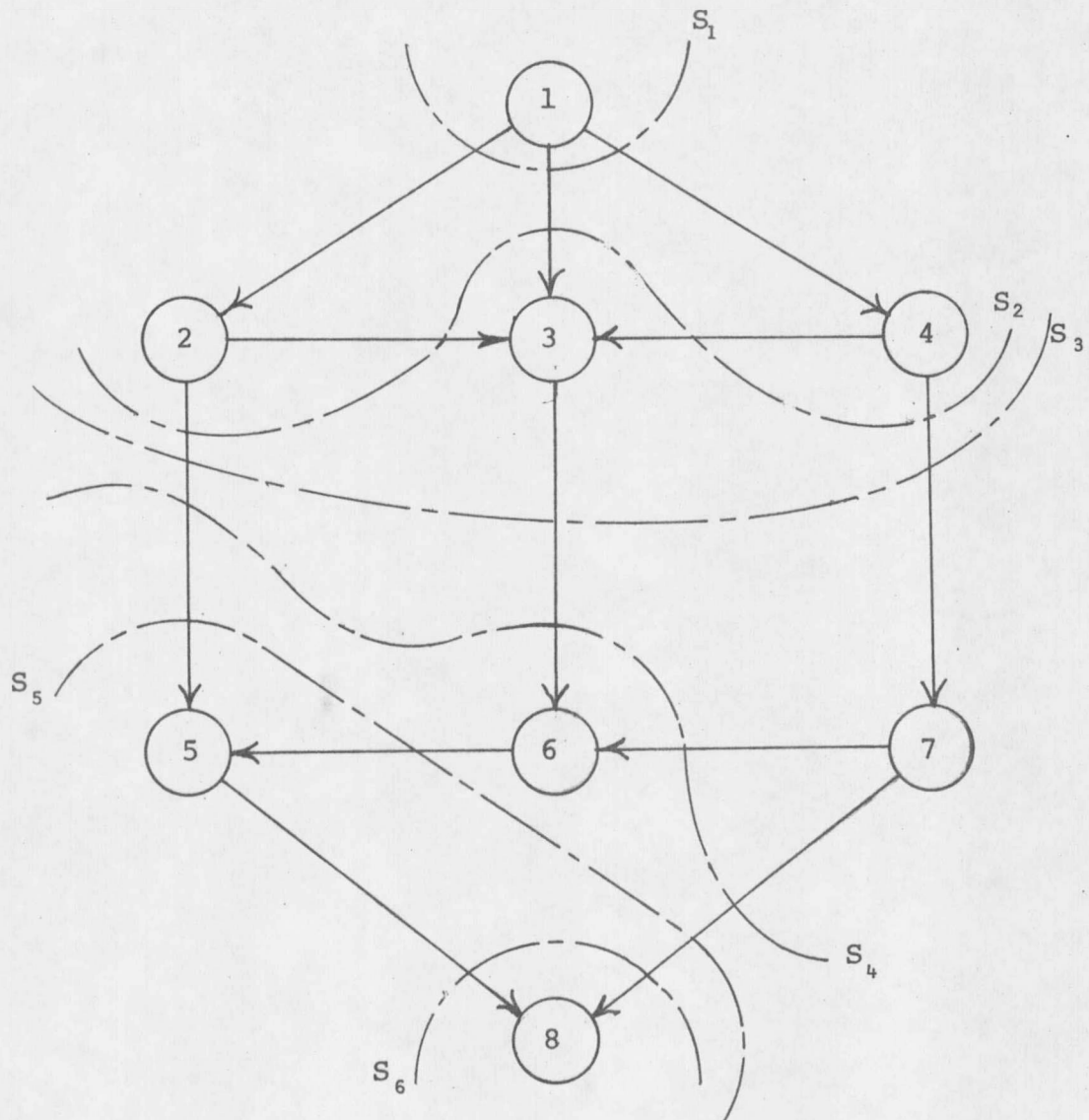


Figure 4.3 A SEC graph with no directed paths from 2 to 4 (or 4 to 2), 2 to 7 and 3 to 7.

and $(3,7)$, is guaranteed by $\{S\}$. However, the pair $(3,7)$ is separated by directed cut-set S_3 . Similarly, $(2,7)$ is separated by S_2 and S_3 because the converse of Theorem 4.4 is not true, in general.

There is no directed cut-set in $\{S\}$ which separates the pair $(2,4)$.

Definition 4.4 A set $V = \{(v_i, v_j)\}$ is a collection of all possible pairs of vertices of a SEC graph G such that there is no directed path from vertex v_i to vertex v_j or from v_j to v_i in the graph G .

In Example 4.3, $V = \{(2,4), (2,7), (3,7)\}$.

In the reachability matrix R of a graph G , suppose $r_{\alpha, \beta} = 1$. Then there is a directed path from a vertex α to a vertex β , and according to Theorem 4.4, if there is a directed path from α to β , separation of pairs of vertices is guaranteed by at least one of the directed cut-sets of $\{S\}$ which contains all edges of G . Hence, with the help of Theorem 4.4 and matrix R , a set V can be found [Definition 4.4] as follows:

$$(v_i, v_j) \in V$$

if and only if

$$r_{v_i, v_j} = 0 \quad \text{and} \quad r_{v_j, v_i} = 0 \quad (4.5)$$

The reachability matrix R of the SEC graph of Figure 4.3 is as follows:

$$R = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{matrix} & \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{matrix}$$

Checking entries of R using (4.5), we obtain

$$V = \{(2,4), (2,7), (3,7)\}$$

Definition 4.5 Total graph [9]: A directed graph is said to be total graph, if every pair of distinct vertices is joined in at least one direction by a directed path.

Corollary 4.1 Let G be a total SEC graph from i to j and E be a set of edges. Let $\{S\}$ be a collection of all possible directed cut-sets separating i and j and covered by edge E . Then G is 1-distinguishable under E , if all edges of G are in $\{S\}$.

The proof is given by using Theorems 4.4 and 2.7. For a total graph, set V is empty. It is found that the ordering of vertices [Definition 2.21] for a total graph is unique, so the proof of Corollary 4.1 can be established by using Theorem 2.8 also.

4.4.3 Final covering matrix Z Given that G is a SEC graph and that $\{S\}_{\min}$ has the minimum number (n_{d_1}) of directed cut-sets that separate every pair of vertices of G , let n_{E_1} be the minimum number of edges that cover all directed cut-sets of $\{S\}_{\min}$. Then G is 1-distinguishable under this set of edges. But, is this the minimum number of test points for this graph to be 1-distinguishable? The answer, in general, is NO!

This can be easily explained. On one extreme, we may need $n_{E_1} = n_{d_1}$ number of edges to cover all directed cut-sets of $\{S\}_{\min}$. But suppose there is a set $\{S\}$, instead of $\{S\}_{\min}$, that has more directed cut-sets

$(n_{d_2}) [n_{d_1} \leq n_{d_2} \leq 2n_{d_1}]$ than n_{d_1} ; even then on the other extreme, we may need $n_{E_2} = (n_{d_2}/2) [\leq n_{d_1}]$, provided that every two directed cut-sets are covered by one edge.

Example 4.4 Consider the SEC graph of Figure 4.4(a). The minimum number of directed cut-sets $\{S\}_{\min}$ which separate every pair of vertices of the graph is three.

$$\{S\}_{\min} = \{S_1(e_2), S_1(e_6), S_1(e_8)\}$$

as shown in Figure 4.4(a). Each of these directed cut-sets is covered by only one edge. Hence, $E_1 = \{e_2, e_6, e_8\}$.

Instead of $\{S\}_{\min}$, there are four directed cut-sets

$$\{S\} = \{S_1(e_1), S_2(e_1), S_1(e_6), S_2(e_6)\}$$

as shown in Figure 4.4(b), that separate every pair of vertices of graph G , and they are covered by only two edges $E_2 = \{e_1, e_6\}$, which is less than E_1 .

In short, we are searching for a collection of directed cut-sets $\{S\}$, instead of $\{S\}_{\min}$, that separates every pair of vertices of the graph and is covered by the minimum set of edges. This relates our problem to that of

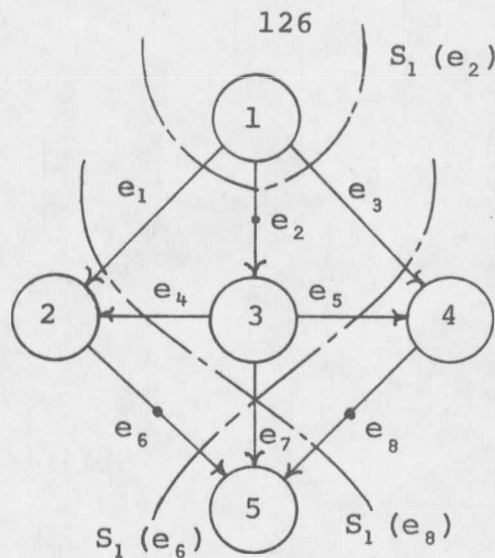


Figure 4.4(a) A SEC graph with the minimum number of directed cut-sets that separate every pair of vertices.

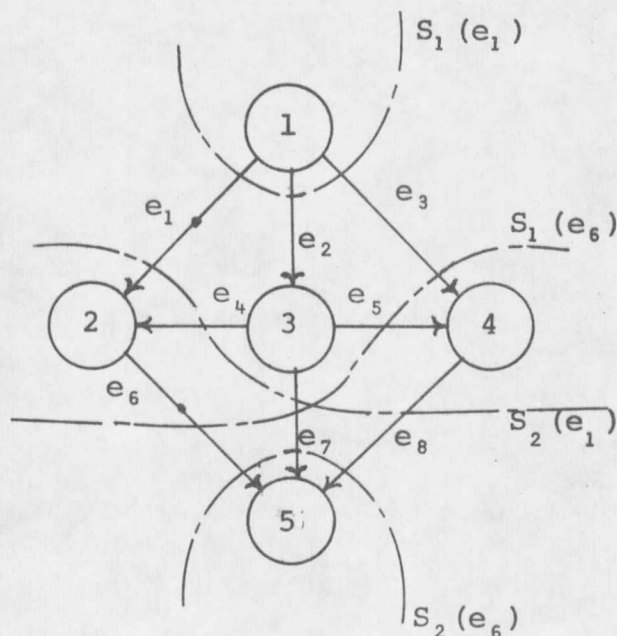


Figure 4.4(b) A SEC graph with the minimum number of edges $E = \{e_1, e_6\}$; directed cut-sets covered by E separate every pair of vertices.

a well-known covering problem. Before going into the details, let us first define matrices Q and P .

Definition 4.6 Matrix $Q = \{q_{e_i, e_j}\}$ of a SEC graph G is associated with vertex set $\Omega = \{v_1, v_2, v_3, \dots, v_n\}$ and the edge-set $E = \{e_1, e_2, e_3, \dots, e_m\}$ and is an $(m \times m)$ matrix such that

$$q_{e_i, e_j} = \begin{cases} 1 & \text{If and only if an edge } e_j \text{ is in a directed cut-set covered by an edge } e_i, \\ 0 & \text{otherwise} \end{cases}$$

Example 4.5 As shown in Section 4.4.1, using matrices C and R , entries in the matrix Q are entered. For the SEC graph of Figure 2.11,

	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8	e_9
e_1	1	1	1	0	1	0	0	0	1
e_2	1	1	1	1	0	0	1	0	0
e_3	1	1	1	0	0	1	0	1	0
e_4	0	1	1	1	1	0	0	0	1
$Q = e_5$	1	0	1	1	1	1	0	0	1
e_6	1	0	1	0	1	1	1	1	0
e_7	1	1	0	0	0	1	1	1	0
e_8	0	0	1	0	0	1	0	1	1
e_9	1	0	0	0	1	0	0	1	1

The edges corresponding to the minimum number of rows that cover all columns of Q are in an edge-set E . Directed cut-sets covered by E contain all edges of the graph. The graph may or may not be 1-distinguishable under E since only a necessary condition for 1-distinguishability [Theorem 2.5] is satisfied by this method.

Definition 4.7 Matrix $P = \{p_{e_k, (v_i, v_j)}\}$ of a SEC graph is associated with a vertex set $\Omega = \{v_1, v_2, v_3, \dots, v_n\}$, an edge-set $E = \{e_1, e_2, e_3, \dots, e_m\}$, and a set $V = \{v_i, v_j\}$ [Definition 4.4], and is an $(m \times n_v)$ matrix such that

$$= 1 \begin{cases} \text{If and only if a pair of ver-} \\ \text{tices } (v_i, v_j) \text{ of a set } V \text{ is} \\ \text{separated by a directed cut-set} \\ \text{covered by an edge } e_k. \end{cases}$$

$$p_{e_k, (v_i, v_j)} = 0 \quad \text{otherwise}$$

where n_v is a number of elements of the set V .

Using the reachability matrix R , entries in matrix Q are generated as follows:

Let (v_i, v_j) be an element of a set V . Then any one of the directed cut-sets covered by an edge e separates this pair of vertices (v_i, v_j) , if and only if

$$r_{v_i, v}(e^+) \neq r_{v_j, v}(e^+)$$

and

$$r_{v(e^-), v_i} \neq r_{v(e^-), v_j}$$

Example 4.6 For the SEC graph of Figure 2.11,

$$V = \{(2,3), (2,4), (2,6), (3,4), (4,5), (5,6)\}$$

$$P = \begin{matrix} & \begin{matrix} (2,3) & (2,4) & (2,6) & (3,4) & (4,5) & (5,6) \end{matrix} \\ \begin{matrix} e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ e_6 \\ e_7 \\ e_8 \\ e_9 \end{matrix} & \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 \end{bmatrix} \end{matrix}$$

Definition 4.8 Final covering matrix $Z = [Q:P]$ is an $(n \times (m+n_v))$ matrix of submatrices Q and P defined in Definitions 4.6 and 4.7.

As seen in this section, the covering of matrix Q satisfies only a necessary condition. But the covering of matrix Z satisfies both necessary and sufficient conditions for 1-distinguishability (Theorem 2.7), because by considering submatrix P along with submatrix Q in the final covering matrix Z , we are taking care of those pairs of vertices (contained in set V) which may or may not be separated by a minimum row-set E , which covers all columns of Q . Hence, covering all columns of Z by a minimum set of rows (an edge-set $E_{\min}$), we are separating all pairs of vertices of the graph by directed cut-sets covered by $E_{\min}$ (Theorem 2.7). Moreover, we have the minimum number of test points to obtain 1-distinguishability of a SEC graph.

4.5 An Algorithm for Inserting the Minimum Number of Test Points on a SEC Graph for 1-distinguishability

This algorithm is prepared for a SEC graph without directed circuits.

The algorithm prepares a connectivity matrix C and then generates the reachability matrix R from matrix C .

It finds all elements in the set V (Definition 4.4).

Then, assuming test points on each edge of the graph, the

algorithm obtains measurable sets $\Omega(i \times e^+)$ and $\Omega(e^- \times j)$, and obtains directed cut-sets covered by an edge of the graph. The final phase prepares a final covering matrix Z and solves it as a minimum cover problem to obtain the minimum number of edges for 1-distinguishability. The algorithm is as follows:

Step (1): Set up the connectivity matrix C for a given SEC graph without directed circuits. Let $\Omega = \{1, 2, \dots, n\}$ be a set of n vertices and $\mathcal{E} = \{e_{i,j}\}$ be a set of M edges. Here, '1' is the entry vertex and 'n' is the exit vertex. ($e_{i,j}$ is an edge from vertex i to vertex j).

Step (2): Generate the reachability matrix R , using the connectivity matrix C (Theorem 4.2).

$$R = \sum_{k=1}^L C^k + I \quad L \leq N$$

Step (3): Find elements (each of them being a pair of vertices) in the set V (Definition 4.4) from the reachability matrix R ;

$$(\alpha, \beta) \in V$$

if and only if

$$r_{\alpha, \beta} = 0 \quad \text{and} \quad r_{\beta, \alpha} = 0$$

Step (4): Since a non-zero entry in matrix C corresponds to a directed edge in graph G , select such an entry $c_{s,t}$. Then there is a directed edge $e_{s,t}$ from vertex s to vertex t .

Step (5): If a test point is placed on the edge $e_{s,t}$, we have two measurable sets, $\Omega(1 \times s)$ and $\Omega(t \times n)$. Their vertices are obtained as follows:

The vertex α is contained in $\Omega(1 \times s)$ if and only if $r_{\alpha, s} = 1$. Similarly, vertex $\alpha \in \Omega(t \times n)$ if and only if $r_{t, \alpha} = 1$, where $\alpha = 1, 2, \dots, n$.

Step (6): Any edge covers either one or two directed cut-sets. Let $S(e_{s,t})$ be a collection of the edges of directed cut-sets covered by edge $e_{s,t}$. These edges are obtained using measurable sets generated in step (4) and matrix C .

$$e_{\alpha, \beta} \in S(e_{s,t})$$

if and only if

$$c_{\alpha,\beta} \neq 0 \text{ for } \begin{cases} \alpha \in \Omega(1 \times s) \\ \text{and } \beta \notin \Omega(1 \times s) \end{cases}$$

Also,

$$e_{\alpha,\beta} \in S(e_{s,t})$$

if and only if

$$c_{\alpha,\beta} \neq 0 \text{ for } \begin{cases} \alpha \notin \Omega(t \times n) \\ \beta \in \Omega(t \times n) \end{cases}$$

Find all such $e_{\alpha,\beta}$.

Step (7): Rows of the final covering matrix Z and columns of its submatrix Q are edges of the graph. In submatrix Q , place a '1' in a row that corresponds to the edge $e_{s,t}$ at the column positions that correspond to edges of $S(e_{s,t})$. That is,

$$(q_{e_{s,t}, e_{\alpha,\beta}}) = \begin{cases} = 1 & \text{if } e_{\alpha,\beta} \in S(e_{s,t}) \\ = 0 & \text{otherwise.} \end{cases}$$

Step (8): Let a pair (α, β) be an element of V . Now, if $r_{\alpha,s} \neq r_{\beta,s}$ and $r_{t,\alpha} \neq r_{t,\beta}$, any one of the directed

cut-sets covered by the edge $e_{s,t}$ separates the above pair of vertices.

Columns of submatrix P of Z are elements of V (obtained in step (3)).

In submatrix P , place a '1' in the row that corresponds to the edge $e_{s,t}$, at the column position that corresponds to the pair of vertices (α, β) , i.e.,

$(p_{e_{s,t}, (\alpha, \beta)}) = 1$ if pair (α, β) is separated by any one of the directed cut-sets covered by edge $e_{s,t}$. Check for all elements of V .

The remaining entries in that row of P are filled with zeros.

Step (9): Find the next nonzero entry $c_{s,t}$ from C . Repeat the algorithm from step (4) onward. Continue until there are no more nonzero entries in matrix C .

Step (10): Solve the final covering matrix Z as a minimum cover problem, by a well-known method [27,56]. To obtain 1-distinguishability, test points are placed on the edges that correspond to the minimum set of rows that cover all the columns of Z .

For the last step, (10), three well-known reduction rules, which are based on row dominance, column dominance,

and row essentiality [38,56] are applied repeatedly to the final covering matrix Z . These reduction rules allow, in general, great simplification in the determination of the minimal cover. After these rules have been applied as many times as possible, a reduced matrix is obtained which is said to be 'cyclic'. This cyclic matrix can not be reduced further with the above three reduction rules.

Cyclic matrices can be solved by writing a boolean function [56], but this approach can be very tedious when the number of columns is large. In such a situation, a more systematic approach of linear integer programming is used [27].

In the linear integer programming approach, a binary variable x_i is associated with each row r_i of the cyclic matrix, which assumes the value of 1 if r_i is selected in the minimal cover, and the value 0 otherwise. Suppose the cyclic matrix is

$$\begin{array}{l} r_1 \\ r_2 \\ r_3 \end{array} \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

Then the linear integer program can be formulated as,

$$\text{Minimize } x_1 + x_2 + x_3$$

$$\text{Subject to } x_1 + x_2 \geq 1$$

$$x_2 + x_3 \geq 1$$

$$x_1 + x_3 \geq 1$$

$$x_1, x_2, x_3 = 0, 1.$$

For the large cyclic matrices, techniques developed by Luccio [27] and Gimpel [17] are valuable to reduce computation time.

Applying reduction techniques [56] to the rows and columns of the final covering matrix Z , as obtained from Examples 4.5 and 4.6, we have the following cyclic matrix:

	e_2	e_4	e_5	e_6	e_7	e_8	e_9	(2,4)
e_2	1	1	0	0	1	0	0	0
e_4	1	1	1	0	0	0	1	1
e_5	0	1	1	1	0	0	1	0
e_6	0	0	1	1	1	1	0	0
e_7	1	0	0	1	1	1	0	1
e_8	0	0	0	1	0	1	1	1
e_9	0	0	1	0	0	1	1	1

1-distinguishability of graph G can be achieved by using any one of the edge-sets $\{e_4, e_7\}$, $\{e_4, e_6\}$, or $\{e_5, e_7\}$. Thus, we have two test points, which is the minimum for a SEC graph of 7 vertices.

$$[N \geq \log_2(n+1) - 1] \text{ or } N \geq 2$$

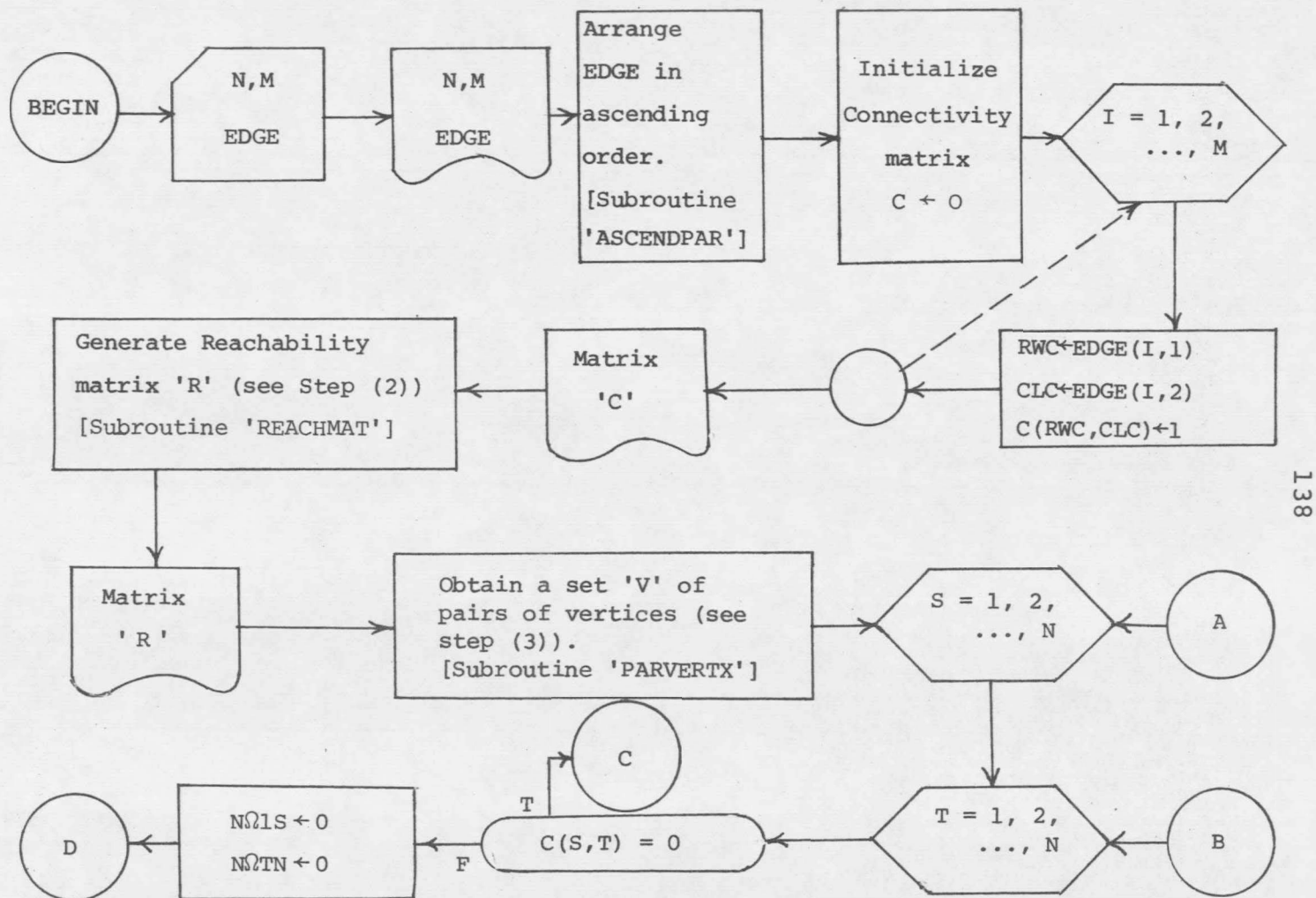


Figure 4.5(a) Flow diagram of MAIN PROGRAM.

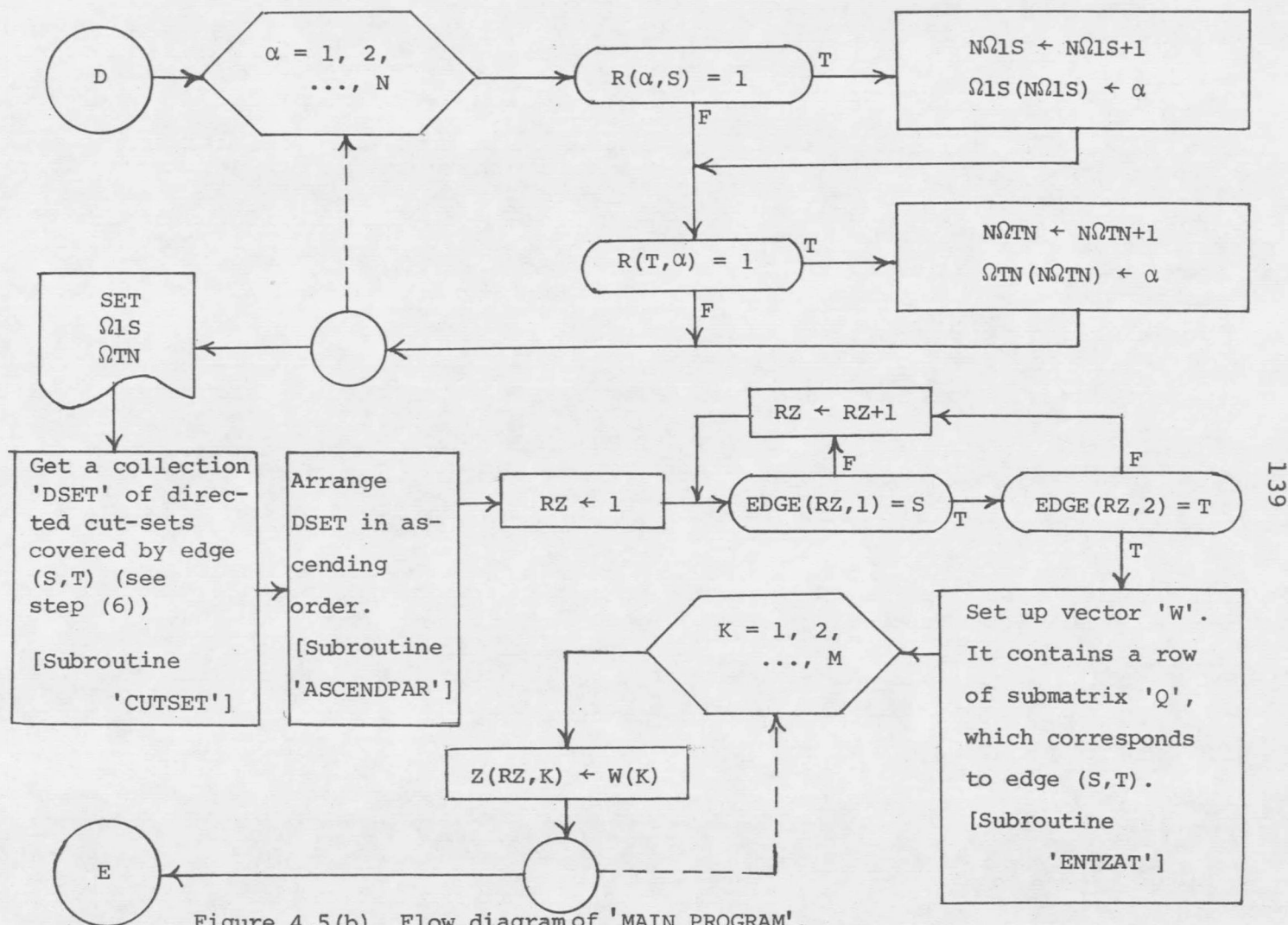


Figure 4.5(b) Flow diagram of 'MAIN PROGRAM'.

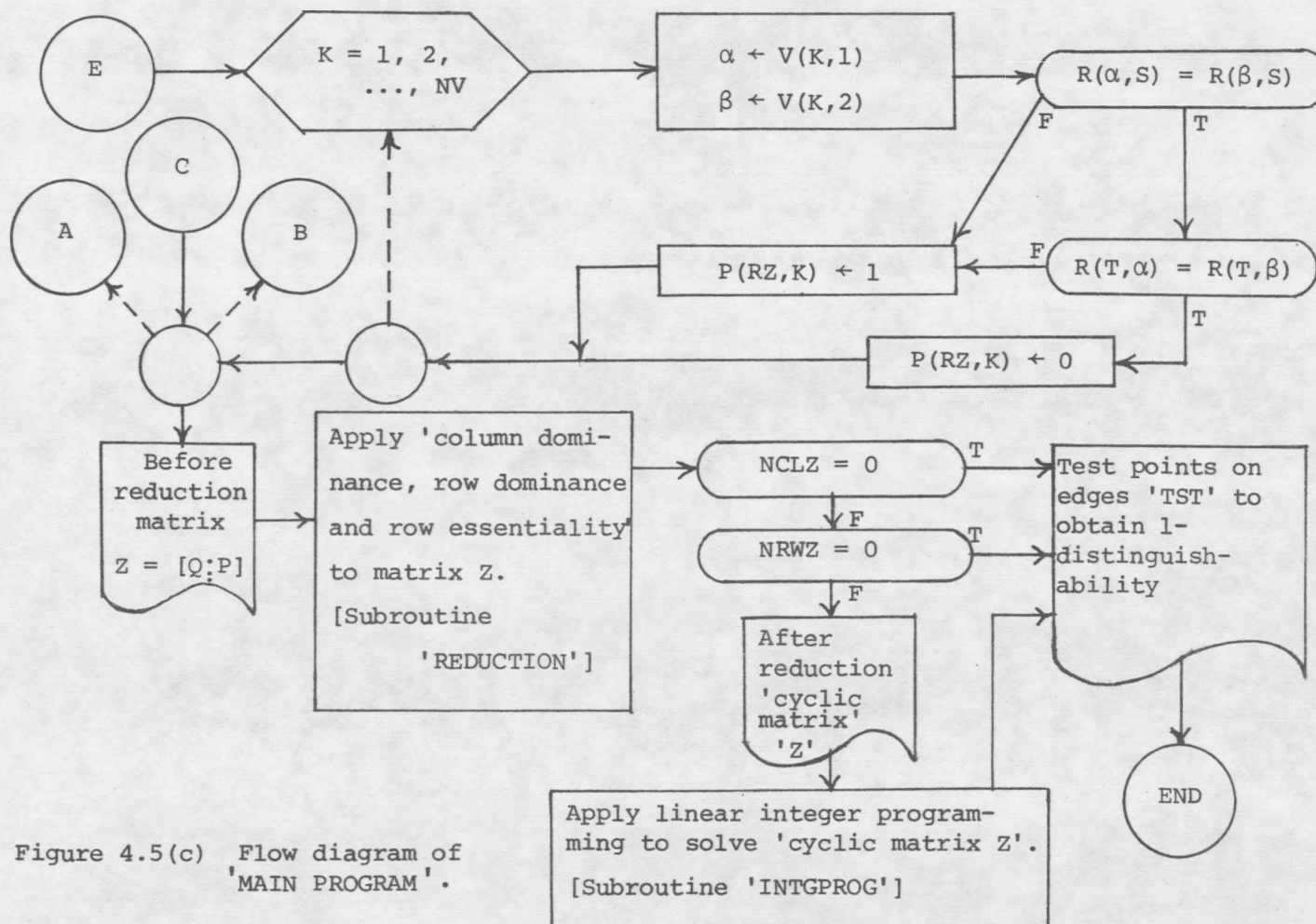


Figure 4.5(c) Flow diagram of 'MAIN PROGRAM'.

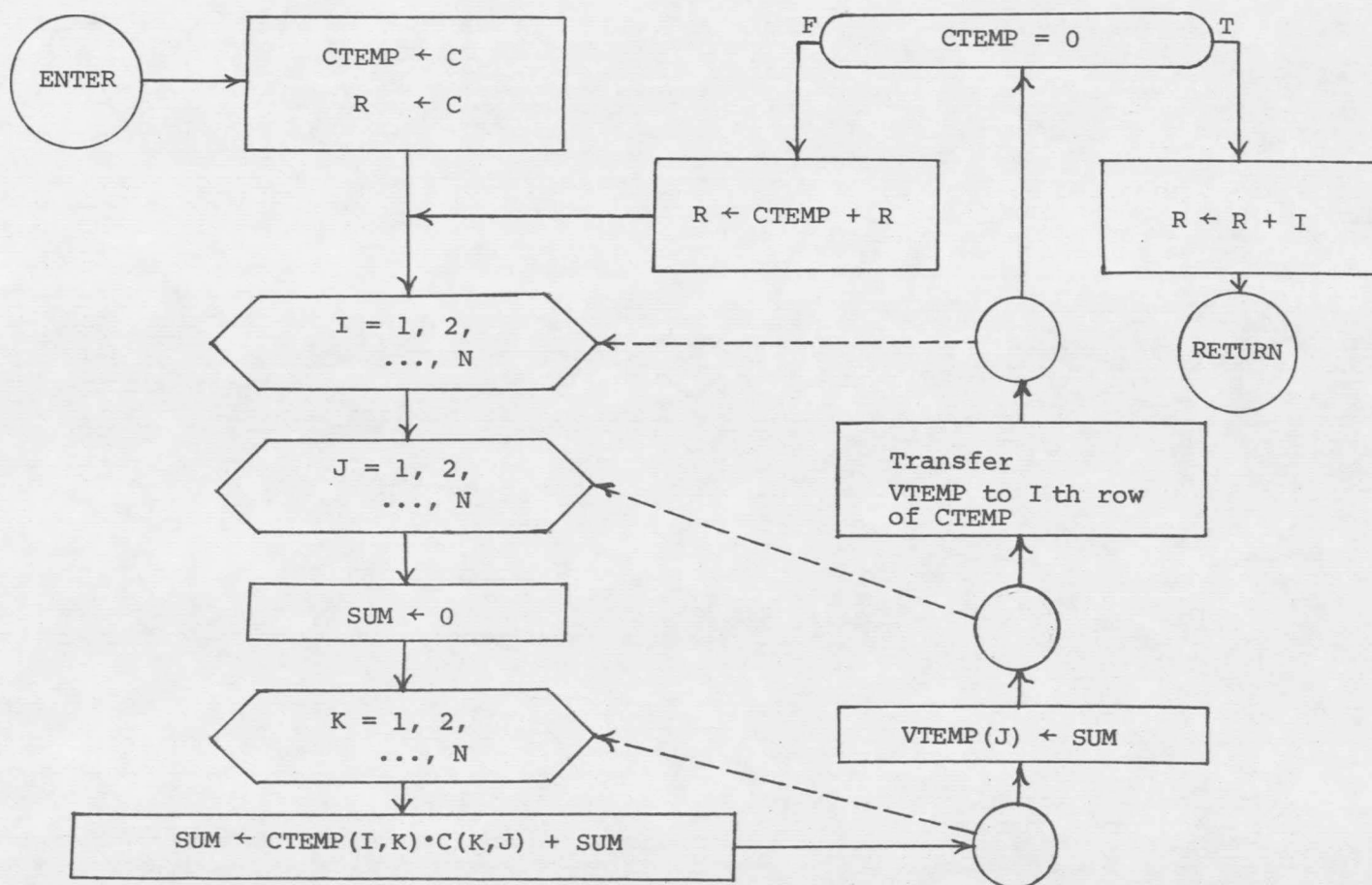


Figure 4.6 Flow diagram of subroutine 'REACHMAT'.

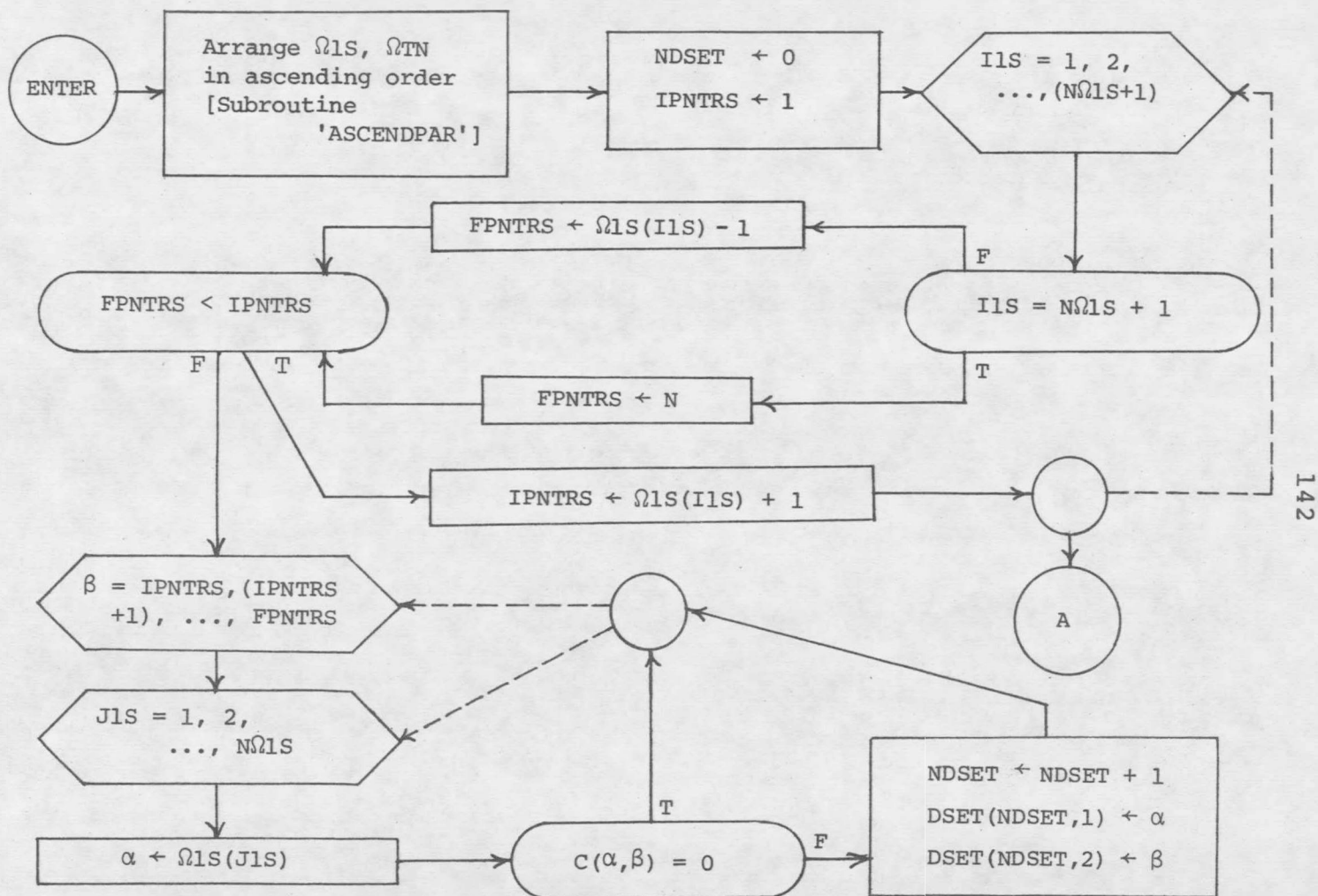
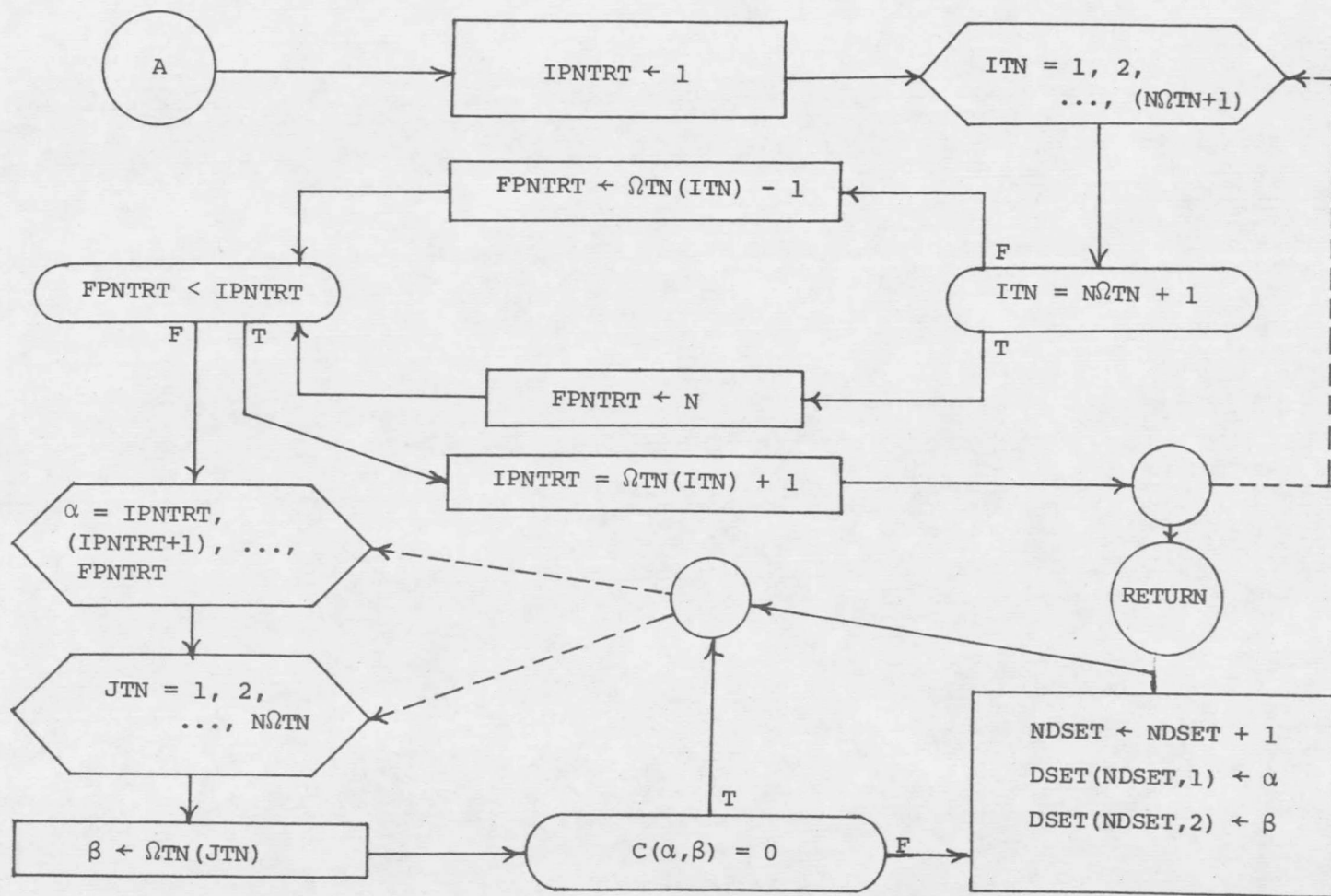


Figure 4.7(a) Flow diagram of subroutine 'CUTSET'.



143

Figure 4.7(b) Flow diagram of subroutine 'CUTSET'.

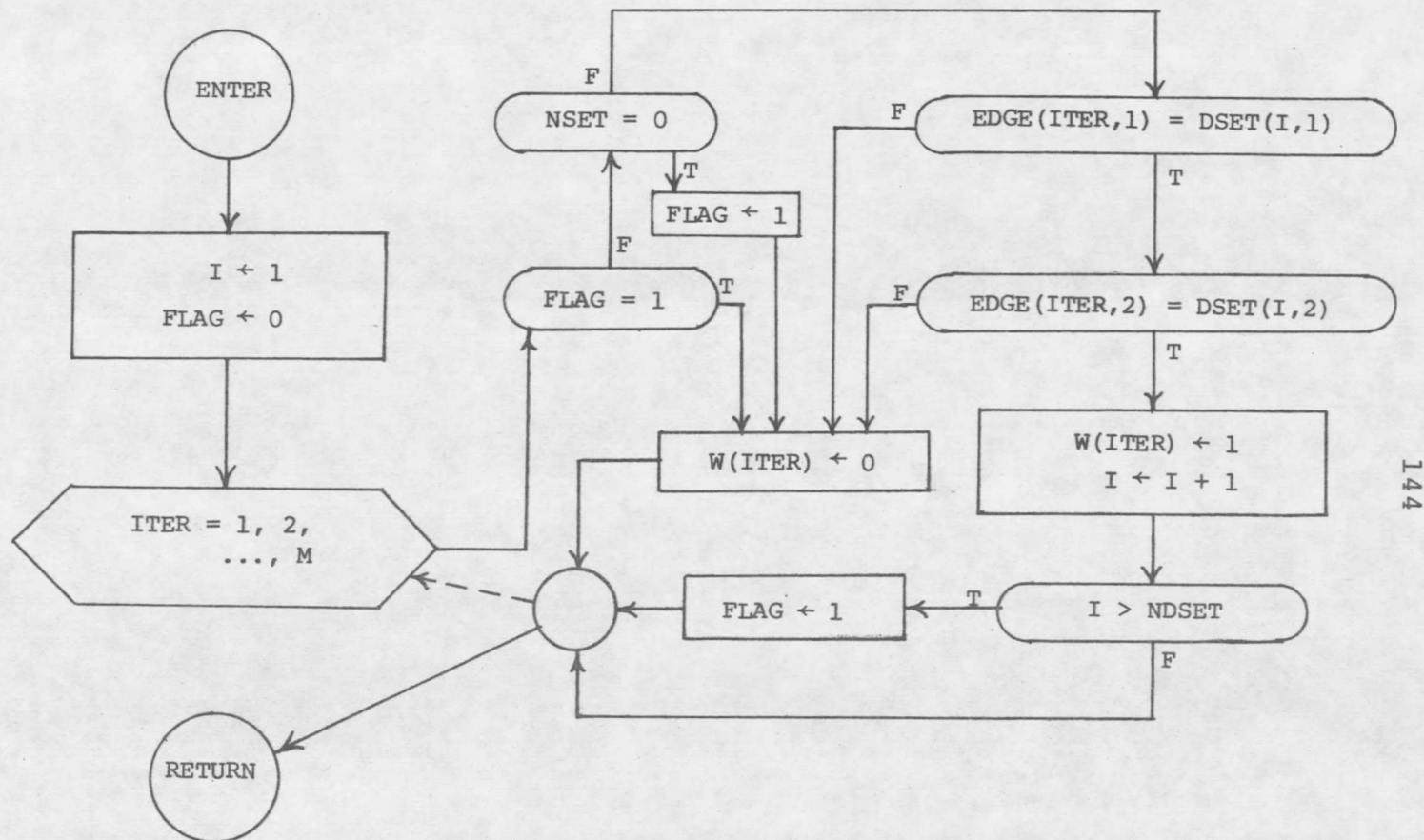


Figure 4.8 Flow diagram of subroutine 'ENTZMAT'.

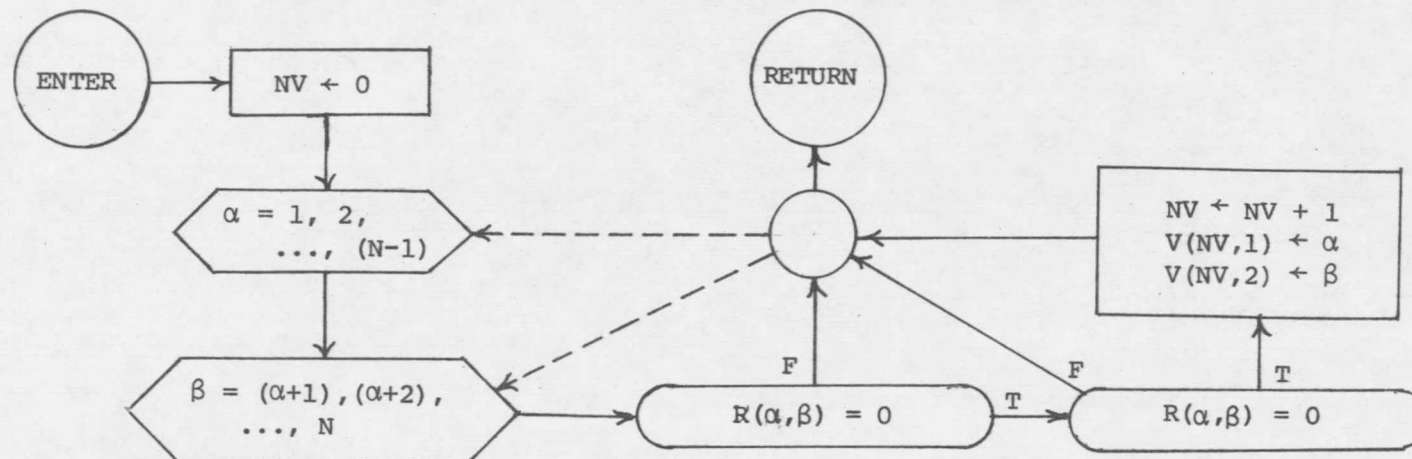


Figure 4.9 Flow diagram of subroutine 'PARVERTX'.

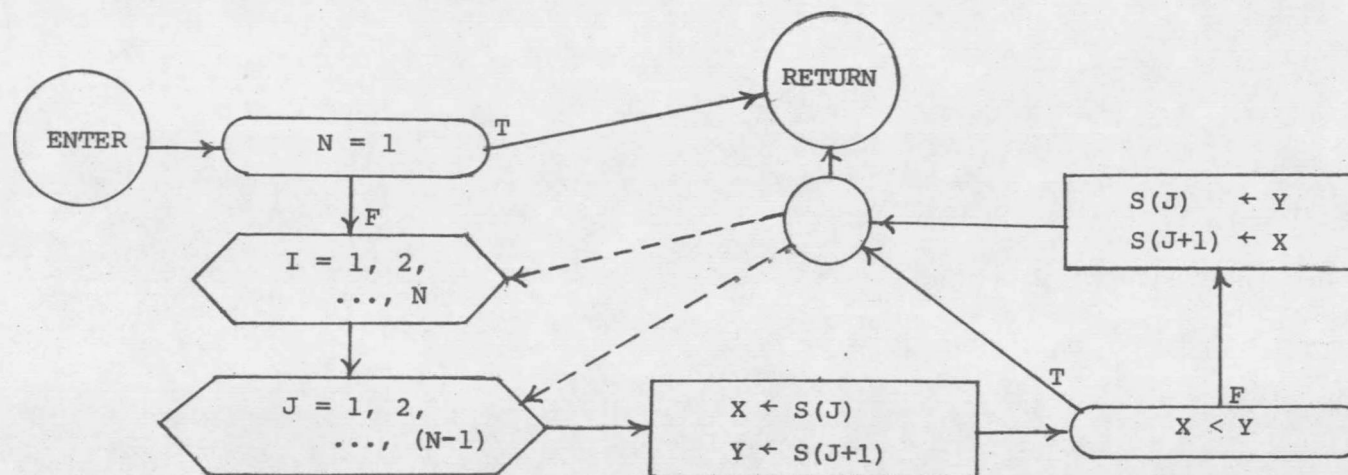


Figure 4.10 Flow diagram of subroutine 'ASCEND'.

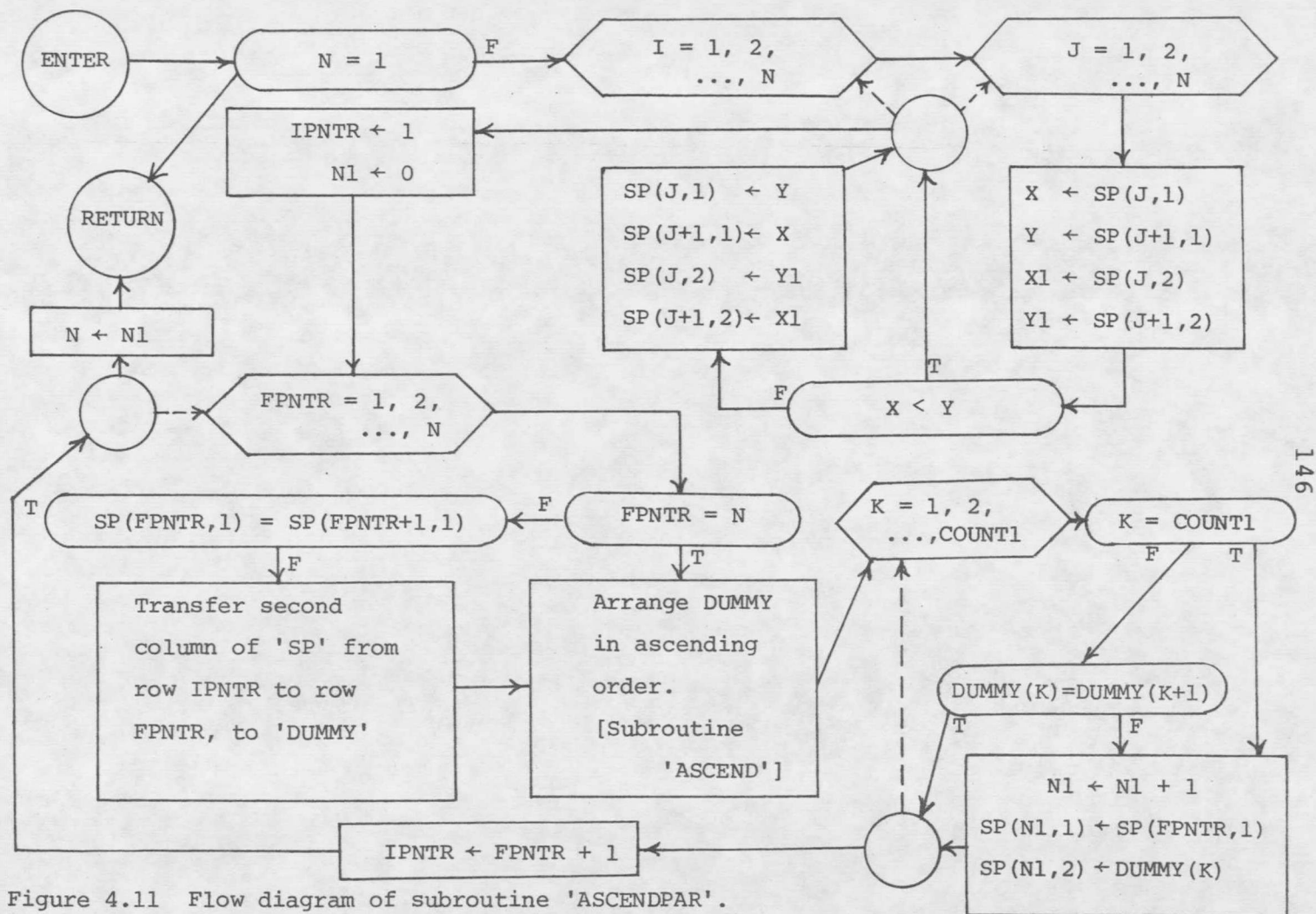


Figure 4.11 Flow diagram of subroutine 'ASCENDPAR'.

Chapter V

SUMMARY AND SUGGESTED FUTURE RESEARCH

5.1 Introduction

This chapter summarizes the results obtained and describes the conclusions drawn from the research conducted. The significance of the findings of this work is discussed, and possible directions for the extension of the work are mentioned.

5.2 Summary and Conclusions

This work is an extension of Ramamoorthy and Mayeda's [30,31,32,40,41] new concepts in the area of fault diagnosis. It deals with a study of structural models of logical circuits and/or computer systems, from a diagnostic viewpoint using graph theory. Considerable attention was given to this technique because it provides a better method of understanding, visualizing, and analyzing diagnostic problems where, in general, most other techniques fall short. This new concept eases the diagnostic problem by separating structural and behavior information of logical circuits and/or computer systems. Structural models are studied by graph theory and are

improved by inserting test points and/or blocking gates. On this improved structure, diagnostic tests are carried out by activating test points with test signals (from a prepared test dictionary) and/or controlling blocking gates. Hence, after structural analysis, behavior information is superimposed.

Theorems of Ramamoorthy and Mayeda on necessary and/or sufficient conditions for test points were particularly useful for checking structural resolution from a diagnostic viewpoint, when test points were pre-assigned on the graph. But there was no systematic method for inserting test points in a system graph to improve its diagnosability. Hence, in this work an algorithm was developed for inserting test points in the system graph. The single-entry, single-exit type of directed graph (SEC graph), without directed circuits, was considered.

Following are the significant points of the algorithm.

1. Test points obtained by the algorithm will achieve 1-distinguishability of the system graph. Hence, it is possible to isolate a fault to a single vertex in the system graph.

2. The algorithm provides the minimum number of test points.

3. Computation time is reasonable: an alternative is to find a collection of directed cut-sets that separates every pair of vertices and then to cover this collection of directed cut-sets by a set of edges to obtain 1-distinguishability, but this approach is more exhaustive than the method developed in the algorithm. An attempt is made in the algorithm to avoid consideration of all pairs of vertices and thereby reduce the computation time. The amount of reduction in computation depends on the graph under consideration and particularly, on the connection of edges between vertices. For graphs with directed paths from any vertex to any other vertex, maximum reduction in computation can be achieved. In general, the longer the directed paths, the greater the reduction in computation time.

The algorithm also reduces the computation required to search for directed cut-sets covered by any edge of the graph.

4. The algorithm relates the problem of inserting test points in the system graph to a well-known covering

problem. Hence, it is possible to use some of the already developed efficient algorithms [17,27,38,56] for solving the covering problem.

5. The algorithm is easy to implement on the computer.

6. The algorithm can be used for large system graphs.

We have established the proof of the theorem on the greatest lower bound on the number of test points to obtain 1-distinguishability, and have given proofs of other related theorems on D-partitions and test points.

5.3 Suggestions for Future Research

The following areas appear to be fruitful for future work:

1. Additional work on the design of test points and blocking gates in order to inject and monitor the test signals is required.

2. The cost of implementation of test points varies for different edges of system graphs. Modification of the present algorithm is required to account for this implementation cost at the time of test point selection.

3. Consideration should be given to development of automatic diagnostic procedures. These procedures should control certain blocking gates, inject and monitor required test signals at appropriate test points, and decide the sequence in which various test points should be activated.

4. Consideration should be given to development of heuristic methods for finding locations of test points. Such procedures may not lead to the minimum number of test points but may reduce computation considerably.

5. It would be desirable to extend this research work so that it covers any general system graph (with any number of entry and exit vertices, and with directed circuits). We can use blocking gates along with test points to obtain maximum distinguishability in this type of general graph. An algorithm for the location of both test points and blocking gates is a fruitful area for further research.

APPENDIX A

SUMMARY OF NOTATION FOR
PROGRAM LISTING

APPENDIX A

An algorithm for inserting the minimum number of test points on a SEC graph for 1-distinguishability was implemented using FORTRAN-IV. It was run at the Montana State University Computation Center using an XDS Sigma 7 computer. This program can be used for graphs of 30 vertices and 90 edges at most. By changing dimension statements, it can be used for larger graphs. The program is initiated by reading the total number of vertices, the total number of edges, and the connection (represented as ordered pairs of vertices) of all edges. Final output consists of the minimum set of edges (represented as ordered pairs of vertices) on which test points should be inserted to obtain 1-distinguishability of a given graph.

Subroutine 'INTGRPROG' is used in the program for solving cyclic matrices (Section 4.5). It uses a linear integer programming method. The program listing for INTGRPROG is given in reference [33], so it is not listed in Appendix B along with other listings.

A summary of notation used for the program is as follows:

(MAIN PROGRAM)

1. N : Total number of vertices of the graph.
2. M : Total number of edges of the graph.
3. EDGE : Edge matrix, whose rows contain ordered pairs (v, v') of the edges of the graph.
4. C : Connectivity matrix.
5. RWC, CLC : Row and column subscripts of matrix C.
6. R : Reachability matrix.
7. ALPHA, BETA : Row and column subscripts of matrix R.
8. S, T : Ordered pair of the edge $e_{s,t}$ on which a test point is assumed.
9. V : Vertex-pair matrix. Its rows contain pairs of vertices whose separation by a collection of directed-cut-sets $\{S\}$ is under consideration. Here $\{S\}$ contains all edges of the graph.
10. NV : Total number of pairs of vertices in V.
11. OMEG1S : Vector of measurable set $\Omega(1 \times s)$.
12. NOMEG1S : Total number of vertices in $\Omega(1 \times s)$.
13. OMEGTN : Vector of measurable set $\Omega(t \times n)$.

- 14. NOMEGTN : Total number of vertices in $\Omega(t \times n)$.
- 15. DSET : Directed cut-set matrix. Its rows contain ordered pairs (v, v') of the edges which are in the directed cut-sets $S(e_{s,t})$ covered by $e_{s,t}$.
- 16. NDSET : Total number of edges in DSET.
- 17. Z : Final covering matrix, which is solved as a minimum cover problem.
- 18. P : Submatrix of Z.
- 19. RZ : Row subscript of Z.
- 20. NCLZ : Total number of columns of Z.
- 21. I, J, K : Iteration counters.
- 22. VDMY : Dummy matrix, contains vertex-pairs that are separated by one of the directed cut-sets, $S(e_{s,t})$ covered by $e_{s,t}$.
- 23. W : Dummy vector contains a row of submatrix P, which corresponds to $e_{s,t}$.

(SUBROUTINE REACHMAT)

- 1. CTEMP : Matrix used for storing intermediate results during matrix multiplications.
- 2. VTEMP : Vector used for storing a new calculated row during matrix multiplication.
- 3. COUNT : Counts total number of zeros in matrix CTEMP to check whether series converges or not.

4. SUM : Logical variable used to store intermediate result during the calculation of

$$C^{\alpha+1} = \sum_{k=1}^N C_{i,k}^{\alpha} \cdot C_{k,j}$$

5. I, J, J1, K : Iteration counters.
 6. ALPHA : Temporary storage.
 7. L1, L2 : Dummy logical variable.

(SUBROUTINE PARVERTX)

1. ALPHA, BETA : Subscripts of matrix R.

(SUBROUTINE CUT-SET)

1. IPNTRS, FPNTRS : Initial and final pointers for a set of vertices which are in between the adjacent entries of OMEG1S. This set of vertices is not in measurable set $\Omega(1 \times S)$.
 2. IPNTRT, FPNTRT : Same as above but true for OMEGTN and $\Omega(1 \times S)$, respectively.
 3. I1S, J1S : Subscripts of OMEG1S.
 4. ITN, JTN : Subscripts of OMEGTN.
 5. ALPHA, BETA : Subscripts of matrix C.

(SUBROUTINE ENTZMAT)

1. FLAG : Flag. FLAG = 1, if all entries in DSET are entered in W, as '0' or '1'. FLAG = 0 otherwise.
 2. I : Subscript of DSET.

3. ITER : Iteration counter.

(SUBROUTINE REDUCTION)

1. TST : Vector to store the address of the edges on which test points should be placed.
2. ITST : Subscripts of TST.
3. SZR : Vector to store row locations that have zero entries for any column.
4. K : Subscripts of SZR.
5. SN : Vector to store column locations.
6. K1 : Subscripts of SN.
7. PTR : Pointer for row which should be deleted.
8. PTC : Pointer for column which should be deleted.
9. NRW : Total number of rows in matrix 'Z'.
10. NRW1, NCLZ1 : Total number of rows and columns, respectively, in matrix 'Z' at any instant before column dominance, row dominance and row essentiality are applied.
11. LNK : Vector to store address of all rows of matrix 'Z'.
12. FR : Flag, FR=1, if row deletion of Z is required.
13. FC : Flag, FC=1, if column deletion of Z is required.

14. I, I1, I2, I3, I4,
J, J1, J2, J3, J4 : Iteration counters.
NCLP, NRWP

(SUBROUTINE DELET)

1. I, I1, I3
J, J1 : Iteration counters.

(SUBROUTINE ASCEND)

1. S : Vector whose entries are required to be arranged in ascending order.
2. I, J : Subscripts of S.
3. N : Total number of entries in S.
4. X, Y : Temporary storage.

(SUBROUTINE ASCENDPAR)

1. SP : Matrix whose rows are required to be arranged in ascending order.
Let i and j be two rows of SP. Then i succeeds j if and only if,
 $SP(i,1) \geq SP(j,1)$ and then
 $SP(i,2) \geq SP(j,2)$.
2. I, J : Subscripts of SP.
3. N : Total number of rows in SP.
4. N1 : Total number of rows left in SP at the end of subroutine, after removing identical rows.
5. DUMMY : Vector whose entries are required to be arranged in ascending order. It contains entries of second column from IPNTR to FPNTR.

- 6. IPNTR,FPNTR : Initial and final pointers indicating initial and final locations of rows that have identical entries in first column.
- 7. COUNT1 : Total number of entries in DUMMY.
- 8. ITER : Iteration counter.
- 9. K : Subscript of DUMMY.
- 10. X,Y,X1,Y1 : Temporary storage.

APPENDIX B

PROGRAM LISTINGS FOR INSERTING THE
MINIMUM NUMBER OF TEST POINTS

A PROGRAM TO INSERT THE MINIMUM # OF TEST POINTS TO OBTAIN
1=DISTINGUISHABILITY OF DIRECTED 'SEC' GRAPHS

.....THIS ALGORITHM IS PREPARED FOR 'SINGLE ENTRY, SINGLE EXIT
CONNECTED=DIRECTED (SEC) GRAPHS WITHOUT DIRECTED CIRCUITS.'
BASICALLY IT RELATES THE PROBLEM TO A WELL-KNOWN 'COVERING
PROBLEM'.

THIS PROGRAM READS THE TOTAL NUMBER OF VERTICES (N) AND
ALL THE EDGES OF THE GRAPH; EACH EDGE IS REPRESENTED AS AN ORDERED
PAIR OF VERTICES (STARTING VERTEX, ENDING VERTEX). AN ENTRY VERTEX
AND EXIT VERTEX ARE ALWAYS LABELED AS '1' AND 'N' RESPECTIVELY.
THE CONNECTIVITY MATRIX 'C' IS SET UP AND THE REACHABILITY MATRIX
'R' IS GENERATED FROM 'C'. THE FINAL OUTPUT IS THE MINIMUM SET OF
EDGES FOR THE LOCATIONS OF TEST POINTS FOR 1=DISTINGUISHABILITY
OF THE GRAPH.

THERE ARE SOME PAIRS OF VERTICES. THEY MAY OR MAY NOT BE
SEPARATED BY A COLLECTION OF DIRECTED CUT-SETS 'S'. HERE 'S'
CONTAINS ALL EDGES OF THE GRAPH. LET 'V' BE A SET OF ALL SUCH
PAIRS OF VERTICES.

A FINAL COVERING MATRIX 'Z' IS PARTITIONED INTO TWO
SUBMATRICES 'Q' AND 'P' (IN THE PROGRAM 'Q' IS NOT USED. HERE IT
IS MENTIONED FOR EXPLANATION ONLY). THE ROWS OF 'Z' AND COLUMNS OF
'Q' REPRESENT ALL THE EDGES OF THE GRAPH, AND THE COLUMNS OF 'P'
REPRESENT THE ELEMENTS OF 'V'.

THE WHOLE PROGRAM IS DIVIDED INTO THREE MAJOR STEPS:

1. ASSUMING TEST POINTS ON AN EDGE, WHICH IS REPRESENTED BY
AN ORDERED-PAIR OF VERTICES (S,T), THE PROGRAM FINDS MEASURABLE
SETS 'OMEGA(1 X S)', 'OMEGA(T X N)' AND GETS A COLLECTION OF
DIRECTED CUT-SETS 'S(EDGE(S,T))' COVERED BY 'EDGE(S,T)'.

2. THE PROGRAM FINDS WHICH PAIR OF VERTICES IN 'V' ARE
SEPARATED BY 'S(EDGE(S,T))'.

3. IN SUBMATRIX 'Q', '1' IS PLACED IN A ROW THAT CORRESPONDS
TO THE 'EDGE(S,T)' AT COLUMN POSITIONS THAT CORRESPOND TO THE
EDGES OF 'S(EDGE(S,T))'.

SIMILARLY IN SUBMATRIX 'P', '1' IS PLACED IN THE SAME ROW
AT THE COLUMN-POSITIONS THAT CORRESPOND TO THE PAIR OF VERTICES
FOUND IN STEP (2). THE REMAINING ENTRIES ARE FILLED WITH ZEROS.

THE FINAL COVERING MATRIX 'Z' IS PREPARED BY REPEATING THE
ABOVE THREE STEPS FOR ALL THE EDGES. 'Z' IS SOLVED AS A MINIMUM
COVERING PROBLEM. TO OBTAIN 1=DISTINGUISHABILITY, TEST POINTS ARE
PLACED ON THE EDGES THAT CORRESPOND TO THE MINIMUM SET OF ROWS

WHICH COVER ALL THE COLUMNS OF 'Z' TO SOLVE MATRIX 'Z' AS A
 MINIMUM COVERING PROBLEM, INITIALLY 'ROW DOMINANCE', 'ROW
 ESSENTIALITY' AND 'COLUMN DOMINANCE' TECHNIQUES ARE APPLIED, AND
 FINALLY, THE 'CYCLIC MATRIX' (IF OBTAINED) IS SOLVED BY LINEAR
 INTEGER PROGRAMMING.

 DATA CARDS REQUIRED

FORMAT OF FIRST DATA CARD : (2I5)
 FIRST FIELD : 'N', THE TOTAL NUMBER OF VERTICES OF THE GRAPH.
 SECOND FIELD : 'M', THE TOTAL NUMBER OF EDGES OF THE GRAPH.
 FORMAT OF SECOND AND ONWARD DATA CARDS : (16I5)
 EIGHT ORDERED PAIRS OF VERTICES THAT CORRESPOND TO ANY EIGHT
 EDGES OF THE GRAPH ARE ENTERED ON ONE OF THESE DATA CARDS.
 ODD NUMBERED FIELD OF EACH DATA CARD : STARTING VERTEX OF AN
 EDGE.
 EVEN NUMBERED FIELD OF EACH DATA CARD: ENDING VERTEX OF THE
 EDGE WHICH HAS THE STARTING VERTEX IN THE FIELD PRECEDING THIS
 FIELD.

***** X *****
 INTEGER RWC,CLC,S,T,ALPHA,BETA,RZ,C,R,Z,EDGE(70,2),V(90,2),
 * OMEG1S(20),OMEGTN(20),DSET(70,2),P(30,90),
 * W(70),EXAMPLPE,TST
 COMMON / COM1 / C(30,30)
 COMMON / COM2 / R(30,30)
 COMMON / COM3 / Z(70,140)
 COMMON / COM4 / LNK(70),TST(15)
 EXAMPLE = 0
 B EXAMPLE = EXAMPLE + 1
READ DATA AND PRINT .
 READ (105,101,END = 9) N, M
 READ (105,102)((EDGE(I,J),J=1,2),I=1,M)
 WRITE (108,103) EXAMPLE
 WRITE (108,104) N, M
 CALL ASCENDPAR (EDGE,M)
 WRITE (108,105) (I,(EDGE(I,J),J=1,2),I=1,M)
INITIALIZATION OF MATRIX 'C' .

```

DO 14 I=1,N
DO 14 J=1,N
C ( I , J ) = 0
14 CONTINUE

....SET=UP CONNECTIVITY MATRIX 'C' .

DO 1 I=1,M
RWC = EDGE ( I,1 )
CLC = EDGE ( I,2 )
C ( RWC,CLC )=1
1 CONTINUE
WRITE (108,106) N, ( I, I=1,N )
DO 10 I = 1,N
WRITE (108,115) I,N, ( C ( I,J), J = 1,N )
10 CONTINUE

....CALL 'REACHMAT' TO GENERATE REACHABILITY MATRIX 'R' .

CALL REACHMAT ( N )
WRITE (108,108) N, ( I , I = 1,N )
DO 11 I = 1,N
WRITE (108,115) I,N, ( R ( I,J), J = 1,N )
11 CONTINUE

....CALL 'PARVERTX' TO OBTAIN A SET 'V' OF PAIRS OF VERTICES'

CALL PARVERTX ( NV,V,NV )
WRITE (108,109) NV, ( I, ( V ( I,J), J=1,2), I=1,NV )
WRITE (108,110)

....FIND MEASURABLE SETS 'OMEGA(1 X S)' AND 'OMEGA(T X N)',
....ASSUMING A TEST=POINT ON 'EDGE(S,T)'.

DO 2 S=1,N
DO 2 T=1,N
IF ( C ( S,T ) .EQ. 0 ) GO TO 2
NOMEG1S = 0
NOMEGTN = 0
DO 3 ALPHA =1,N
IF ( R ( ALPHA,S ) .EQ. 1 ) NOMEG1S=NOMEG1S+1 ; OMEG1S ( NOMEG1S )=ALPHA
IF ( R ( T,ALPHA ) .EQ. 1 ) NOMEGTN=NOMEGTN+1 ; OMEGTN ( NOMEGTN )=ALPHA
3 CONTINUE
WRITE (108,111) S,NOMEG1S, ( OMEG1S ( I ), I=1,NOMEG1S ), T,NOMEGTN, ( OMEGTN
*( J ), J=1,NOMEGTN )

....CALL 'CUTSET' TO FIND A COLLECTION OF DIRECTED CUT=SETS

```

```

C      ....COVERED BY 'EDGE(S,T)'.
C
C      CALL CUTSET (N,OMEG1S,NOMEG1S,OMEGTN,NOMEGTN,DSET,NDSET)
C      CALL ASCENDPAR(DSET,NDSET)
C
C      ....FILL THE ENTRIES OF 'Z' MATRIX AT THE ROW POSITION THAT CORRE-
C      ....SPONDS TO 'EDGE(S,T)'.
C
C      RZ=1
4      IF(EDGE(RZ,1).NE.S) RZ=RZ+1 ; GO TO 4
C      IF(EDGE(RZ,2).NE.T) RZ=RZ+1 ; GO TO 4
C      CALL ENTZMAT (DSET,NDSET,EDGE,M,W)
C      DO 6 K=1,M
C      Z(RZ,K)=W(K)
6      CONTINUE
C      DO 5 K=1,NV
C      ALPHA=V(K,1)
C      BETA=V(K,2)
C      IF(R(ALPHA,S).NE.R(BETA,S)) GO TO 7
C      IF(R(T,ALPHA).EQ.R(T,BETA)) P(RZ,K)=0 ; GO TO 5
7      P(RZ,K)=1
5      CONTINUE
2      CONTINUE

C      ....PRINT 'Z' MATRIX BEFORE REDUCTION .
C
C      WRITE (108,112) (1.5*M+15),(1.5*M+1.5*Nv-4),M,(I,I=1,M),NV,(J,J=1
C      *NV)
C      DO 12 I=1,M
C      WRITE(108,113) I,M,(Z(I,J),J=1,M),NV,(P(I,J),J=1,NV)
12      CONTINUE

C      ....CALL 'REDUCTION' TO REDUCE MATRIX TO A 'CYCLIC MATRIX'.
C
C      CALL REDUCTION (M,NV,NCLZ,NRWZ,ITST,P)
C      IF(NCLZ.EQ.0) GO TO 15
C      IF(NRWZ.EQ.0) GO TO 15
C      WRITE (108,114)
C      DO 13 I=1,NRWZ
C
C      ....PRINT 'CYCLIC MATRIX Z' AFTER REDUCTION.
C
C      K10 = LNK(I)
C      WRITE (108,107) K10,NCLZ,(Z(I,J),J=1,NCLZ)
13      CONTINUE
C      CALL INTGRPROG(NCLZ,NRWZ,ITST)
15      WRITE(108,116)

```

```

DO 16 K11=1,ITST
WRITE(108,105)TST(K11),EDGE(TST(K11),1),EDGE(TST(K11),2)
16 CONTINUE
GO TO 8
9 STOP
*****
.....FORMATS FOR INPUT AND OUTPUT STATEMENTS.....
*****
101 FORMAT (2I5)
102 FORMAT (16I5)
103 FORMAT(1H1,/,36X,/,.....EXAMPLE # (',I2,').....,/,26X, '***
$ *****
$ THE EDGES TO INSERT MINIMUM NUMBER OF',/,28X, 'TEST POINTS TO OBTAIN
$ 1-DISTINGUISHABILITY',/,28X, 'OF GIVEN DIRECTED SEC GRAPH FROM 1 TO
$ N',/,26X, '*****
$ )
104 FORMAT(33X, 'N = NUMBER OF VERTICES OF GRAPH = ',I3,/,33X, 'M = NUM
$ BER OF EDGES OF GRAPH = ',I3,/,43X, 'LISTING OF EDGES',/, 41
$ X, '-----',/,35X, 'EDGE # : FROM VERTEX : TO VERT
$ EX', /)
105 FORMAT (39X,I3,8X,I3,3X,I3,/)
106 FORMAT(1X,/,30X,':C(CONNECTIVITY MATRIX OF GRAPH) :',/,28X, '-----
$ -----
$ STRIX ARE VERTICES OF GRAPH )',/,29X, 'N(I2,2X),/)
107 FORMAT (24X,I2,4X,N(I1,2X),/)
108 FORMAT(1X,/,30X,':R ( REACHABILITY MATRIX OF GRAPH ):',/,28X, '-----
$ -----
$ F MATRIX ARE VERTICES OF GRAPH )',/,29X, 'N(I2,2X),/)
109 FORMAT(1X,/,30X,':V (SET OF PAIRS OF VERTICES) :',/,28X, '-----
$ -----
$ CONSIDERATION',/,16X, 'BY SET (S) OF DIRECTED CUT-SETS : HERE (
$ S) CONTAINS ALL EDGES OF GRAPH' ,
$ , N(43X, I2, 10X, I2, ', ', I2, ', ', /)
110 FORMAT (30X,': MEASURABLE SETS :',/,28X, '-----',/,
$ )
111 FORMAT (25X, 'OMEGA ( 1 X ',I2, ' ) = ',N(I2, ', ',1X),/,25X, 'OMEGA (
$ ,I2, ' X N ) = ',N(I2, ', ',1X),/)
112 FORMAT ( 30X,': Z ( FINAL COVERING MATRIX BEFORE REDUCTION ) :',/,
$ 28X, '-----
$ 25X, 'ROWS OF Z ARE EDGES OF GRAPH :',/,25X, 'COLUMNS OF SUBMATR
$ IX Q ARE EDGES OF GRAPH :',/,25X, 'COLUMNS OF SUBMATRIX P ARE P
$ AIR OF VERTICES FROM SET V :',/,NX, 'SUBMATRIX Q', NX, 'SUBMA
$ TRIX P',/,19X, N(I2,1X),2X,': ',3X, N(I2,1X),/)
113 FORMAT (14X,I2,4X, N(I1,2X) ,1X, ': ', 4X, N(I1,2X) ,/)

```

```

114  FORMAT(30X,'CYCLIC MATRIX AFTER REDUCTION',//,28X,'=====
$=====')
115  FORMAT(24X,I2,4X,N(I1,3X),/)
116  FORMAT(30X,'INSERT TEST POINTS (THE MINIMUM NUMBER) ON THE FOLLOWI
$NG',//,30X,'EDGES TO OBTAIN 1-DISTINGUISHABILITY OF GIVEN GRAPH',
$//, 28X,'=====')
$//,35X,'EDGE # :FROM VERTEX : TO VERTEX',/)
END

```



```

      IF (COUNT .EQ. (N*N)) GO TO 7
C     ....ALGEBRIC SUMMATION OF THE SERIES OBTAINED SO FAR.
      DO 5 I=1,N
        DO 5 J=1,N
          IF (CTEMP(I,J) .EQ. 0) L1= .FALSE. ; GO TO 12
          L1= .TRUE.
12         IF (R(I,J) .EQ. 0) L2= .FALSE. ; GO TO 13
          L2= .TRUE.
13        IF (L1 .OR. L2) R(I,J)=1 ; GO TO 5
          R(I,J) = 0
5       CONTINUE
        GO TO 6
C     ....ADDING UNITY MATRIX TO THE SERIES .
7      DO 8 I=1,N
        R(I,I) = 1
8      CONTINUE
      RETURN
      END

```

C
C
C
C
C

SUBROUTINE PARVERTX (N,V,NV)

....THIS SUBROUTINE IS USED TO FIND A SET 'V' OF PAIRS OF VERTICES
THAT MAY OR MAY NOT BE SEPARATED BY A COLLECTION OF DIRECTED
CUT-SETS. HERE THE COLLECTION OF DIRECTED CUT-SETS CONTAINS
ALL EDGES OF THE GRAPH.

INTEGER ALPHA , BETA , V(90,2) , R
COMMON / COM2 / R(30,30)

NV = 0

DO 1 ALPHA = 1 , (N-1)

DO 1 BETA = (ALPHA + 1) , N

IF (R(ALPHA,BETA) .NE. 0) GO TO 1

IF (R(BETA,ALPHA) .NE. 0) GO TO 1

NV = NV + 1

V (NV,1) = ALPHA

V (NV,2) = BETA

1 CONTINUE

RETURN

END

```

C      SUBROUTINE CUTSET ( N,OMEG1S,NOMEG1S,OMEGTN,NOMEGTN,DSET,NDSET )
C
C      THIS SUBROUTINE FINDS A COLLECTION OF DIRECTED CUT=SETS COVER=
C      ED BY THE 'EDGE(S,T)' USING MEASURABLE SETS 'OMEGA(1 X S)'
C      AND 'OMEGA(T X N)' .
C
C      INTEGER ALPHA,BETA,C,OMEG1S(20),OMEGTN(20),DSET(70,2),FPNTRS,FPNTR
C      *T
C      COMMON / COM1 / C(30,30)
C      CALL ASCEND ( OMEG1S,NOMEG1S )
C      CALL ASCEND ( OMEGTN ,NOMEGTN )
C      NDSET = 0
C      ....TO FIND THE DIRECTED CUT=SET I E(OMEGA(1 X S) X (COMPLEMENT
C      ....(OMEGA(1 X S))))' .
C      IPNTRS = 1
C      DO 1 I1S = 1,(NOMEG1S+1)
C      IF ( I1S .EQ. (NOMEG1S+1)) FPNTRS=N; GO TO 2
C      FPNTRS = OMEG1S(I1S) = 1
C      2 IF(FPNTRS .LT. IPNTRS ) GO TO 3
C      DO 4 BETA = IPNTRS,FPNTRS
C      DO 5 J1S = 1,NOMEG1S
C      ....'OMEG1S(J1S)' IS 'ALPHA' .
C      IF(C(OMEG1S(J1S),BETA) .NE. 0 ) GO TO 6
C      5 CONTINUE
C      GO TO 4
C      6 NDSET = NDSET + 1
C      DSET(NDSET,1) = OMEG1S(J1S)
C      DSET(NDSET,2) = BETA
C      GO TO 5
C      4 CONTINUE
C      3 IPNTRS = OMEG1S(I1S) + 1
C      1 CONTINUE
C      ....TO FIND THE DIRECTED CUT=SET I E((COMPLEMENT(OMEGA(T X N)))
C      ....X OMEGA(T X N)))' .
C      IPNTRT = 1
C      DO 7 ITN = 1,(NOMEGTN+ 1)
C      IF (ITN .EQ. (NOMEGTN +1)) FPNTRT = N ; GO TO 8
C      FPNTRT = OMEGTN (ITN) = 1
C      8 IF(FPNTRT .LT. IPNTRT ) GO TO 9
C      DO 10 ALPHA = IPNTRT,FPNTRT
C      DO 11 JTN = 1,NOMEGTN
C      ....'OMEGTN(JTN)' IS 'BETA' .
C      IF(C(ALPHA,OMEGTN(JTN)) .NE. 0 ) GO TO 12
C      11 CONTINUE
C      GO TO 10
C      12 NDSET =NDSET +1
C      DSET(NDSET,1)=ALPHA

```

```
      DSET(NDSET,2) = OMEGTN(JTN)  
      GO TO 11  
10  CONTINUE  
9   IPNTRT = OMEGTN(IITN) + 1  
7   CONTINUE  
    RETURN  
    END
```

C
C
C
C
C
C

SUBROUTINE ENTZMAT (DSET,NDSET,EDGE,M,W)

....VECTOR 'W' CONTAINS THE ROW CORRESPONDING TO 'EDGE(S,T)' OF
SUBMATRIX 'P' AT THE END OF THIS SUBROUTINE. IT IS OBTAINED
BY PLACING '1' IN 'W' WHEN WE HAVE IDENTICAL ENTRIES IN
THE ROW OF 'EDGE' AND 'DSET'. PLACE '0' OTHERWISE.

INTEGER W(70),EDGE(70,2), DSET(70,2),FLAG

I = 1

FLAG = 0

DO 1 ITER = 1, M

IF (FLAG .EQ. 1) GO TO 2

IF (NDSET .EQ. 0) FLAG = 1 ; GO TO 2

IF (EDGE(ITER,1).NE. DSET(I,1)) GO TO 2

IF (EDGE(ITER,2).NE. DSET(I,2)) GO TO 2

W (ITER) = 1

I = I + 1

IF (I.GT.NDSET) FLAG = 1 ; GO TO 1

GO TO 1

2 W (ITER) = 0

1 CONTINUE

RETURN

END

```

C      SUBROUTINE REDUCTION(M,NV,NCLZ,NRWZ,ITST,P)
C
C      ....THIS SUBROUTINE IS USED TO APPLY 'ROW DOMINANCE', 'COLUMN
C      DOMINANCE' AND 'ROW ESSENTIALITY' TO FINAL COVERING MATRIX 'Z'.
C
      INTEGER P(30,90),FC,FR,Z,TST,COUNT1,PTR,PTC,SZR(70),SN(140)
      COMMON / COM3 / Z(70,140)
      COMMON / COM4 / LNK(70),TST(15)
      NCLZ = M
      IF(NV.EQ.0) GO TO 22
      DO 1 NCLP = 1,NV
        NCLZ = NCLZ + 1
        DO 2 NRWP = 1,M
          Z(NRWP,NCLZ) = P(NRWP,NCLP)
        2 CONTINUE
      1 CONTINUE
      C      ....FINDING ESSENTIAL ROW AND DELETING ALL COLUMNS THAT HAVE '1'
      C      ....ENTRY IN ESSENTIAL ROW.
      22 ITST = 0
      FC = 0
      FR = 0
      NRWZ = M
      DO 3 I1 = 1, NRWZ
        LNK(I1) = I1
      3 CONTINUE
      21 NCLZ1 = NCLZ
      NRWZ1 = NRWZ
      10 COUNT1 = 0
      DO 4 J = 1,NCLZ
        DO 5 I = 1,NRWZ
          IF(Z(I,J).NE.1) GO TO 5
          COUNT1 = COUNT1 + 1
          PTR = I
        5 CONTINUE
        IF(COUNT1.EQ.1) GO TO 6
        COUNT1 = 0
      4 CONTINUE
      GO TO 7
      6 ITST = ITST + 1
      TST(ITST) = LNK(PTR)
      J1 = 1
      9 IF(Z(PTR,J1).NE.1) GO TO 8
      FC = 1
      CALL DELET(FR,FC,NRWZ,NCLZ,PTR,J1,ITST)
      IF (NCLZ.EQ.0) GO TO 23
      GO TO 9
      8 J1 = J1 + 1
      23

```

```

      IF(J1.LE.NCLZ) GO TO 9
      FR=1
      CALL DELET(FR,FC,NRWZ,NCLZ,PTR,PTC,ITST)
      IF(NRWZ.EQ.0) GO TO 23
      GO TO 10
C   7   J2= 1
      ....CHECKING 'COLUMN DOMINANCE' AND DELETING DOMINATING COLUMN.
C   15   K=0
      DO 12 I2=1,NRWZ
         IF(Z(I2,J2).NE.0) GO TO 12
         K=K+1
         SZR(K) = I2
C   12   CONTINUE
      DO 13 J3= 1,NCLZ
         IF(J3.EQ.J2) GO TO 13
         DO 14 I3=1,K
            IF(Z(SZR(I3),J3).EQ.1) GO TO 13
C   14   CONTINUE
      FC=1
C   ....SUBROUTINE 'DELET' FOR DELETING DOMINATING COLUMN.
      CALL DELET(FR,FC,NRWZ,NCLZ,PTR,J2,ITST)
      IF(NCLZ.EQ.0) GO TO 23
      GO TO 101
C   13   CONTINUE
      J2=J2+1
C   101  IF(J2.LE.NCLZ) GO TO 15
      I3=1
C   ....CHECKING 'ROW DOMINANCE' AND DELETING DOMINATED ROW.
C   19   K1=0
      DO 16 J3=1,NCLZ
         IF(Z(I3,J3).NE.1) GO TO 16
         K1=K1+1
         SN(K1)=J3
C   16   CONTINUE
      IF(K1.NE.0) GO TO 20
      FR=1
C   ....SUBROUTINE 'DELET' FOR DELETING DOMINATED ROW
      CALL DELET(FR,FC,NRWZ,NCLZ,I3,PTC,ITST)
      IF(NRWZ.EQ.0) GO TO 23
      GO TO 100
C   20   DO 17 I4= 1,NRWZ
         IF(I4.EQ.I3) GO TO 17
         DO 18 J4=1,K1
            IF(Z(I4,SN(J4)).EQ.0) GO TO 17
C   18   CONTINUE
      FR=1
      CALL DELET(FR,FC,NRWZ,NCLZ,I3,PTC,ITST)

```

```

      IF(NRWZ.EQ.0) GO TO 23
      GO TO 100
17  CONTINUE
      I3=I3+1
C 100 IF(I3.LE.NRWZ) GO TO 19
      ....CHECK FOR A CYCLIC MATRIX.
      IF(NCLZ.NE.NCLZ1) GO TO 21
      IF(NRWZ.NE.NRWZ1) GO TO 21
23  RETURN
      END

```


* RETURN
END

C
C
C
C

SUBROUTINE ASCEND (S,N)

....THIS SUBROUTINE ARRANGES THE ENTRIES OF VECTOR 'S' IN
....ASCENDING ORDER.

```
INTEGER X,Y,S(1)
IF(N.EQ.1) GO TO 3
DO 1 I=1,N
  DO 2 J=1,(N-1)
    X= S(J)
    Y= S(J+1)
    IF(X.LT.Y) GO TO 2
    S(J)=Y
    S(J+1)=X
  2 CONTINUE
1 CONTINUE
3 RETURN
END
```

```

C      SUBROUTINE ASCENDPAR (SP,N)
C
C      ....THIS SUBROUTINE ARRANGES THE ENTRIES OF 'SP' IN ASCENDING
C      ORDER.
C
C      INTEGER ALPHA,X,Y,X1,Y1,COUNT1,SP(70,2),DUMMY(70),FPNTR
C      IF (N.EQ. 1) GO TO 7
C
C      ....ARRANGE FIRST COLUMN OF 'SP' IN ASCENDING ORDER .
C
C      DO 1 I=1,N
C      DO 2 J=1,(N-I)
C      ALPHA = J+1
C      X=SP(J,1)
C      Y=SP(ALPHA,1)
C      X1= SP(J,2)
C      Y1= SP(ALPHA,2)
C      IF(X.LT.Y) GO TO 2
C      SP(J,1) = Y
C      SP (ALPHA,1)=X
C      SP(J,2)=Y1
C      SP (ALPHA,2) =X1
C      2 CONTINUE
C      1 CONTINUE
C      ....ARRANGE SECOND COLUMN OF 'SP' IN ASCENDING ORDER .
C      IPNTR = 1
C      N1 = 0
C      DO 3 FPNTR = 1,N
C      IF ( FPNTR.EQ. N ) GO TO 4
C      ....COUNT THE IDENTICAL ENTRIES IN A FIRST COLUMN OF 'SP' .
C      IF (SP (FPNTR,1) .EQ. SP(FPNTR+1,1)) GO TO 3
C      4 COUNT1=0
C      DO 5 ITER = IPNTR , FPNTR
C      COUNT1 = COUNT1 + 1
C      DUMMY ( COUNT1 ) = SP(ITER,2)
C      5 CONTINUE
C      CALL ASCEND ( DUMMY , COUNT1)
C      DO 6 K = 1,COUNT1
C      IF (K .EQ. COUNT1 ) GO TO 8
C      ....DELETE IDENTICAL ROWS OF 'SP' .
C      IF ( DUMMY(K) .EQ. DUMMY(K+1)) GO TO 6
C      8 N1 = N1 + 1
C      SP (N1,1) = SP (FPNTR, 1)
C      SP (N1,2) = DUMMY(K)
C      6 CONTINUE
C      IPNTR = FPNTR + 1
C      3 CONTINUE

```

180

N 2 N1
7 RETURN
END

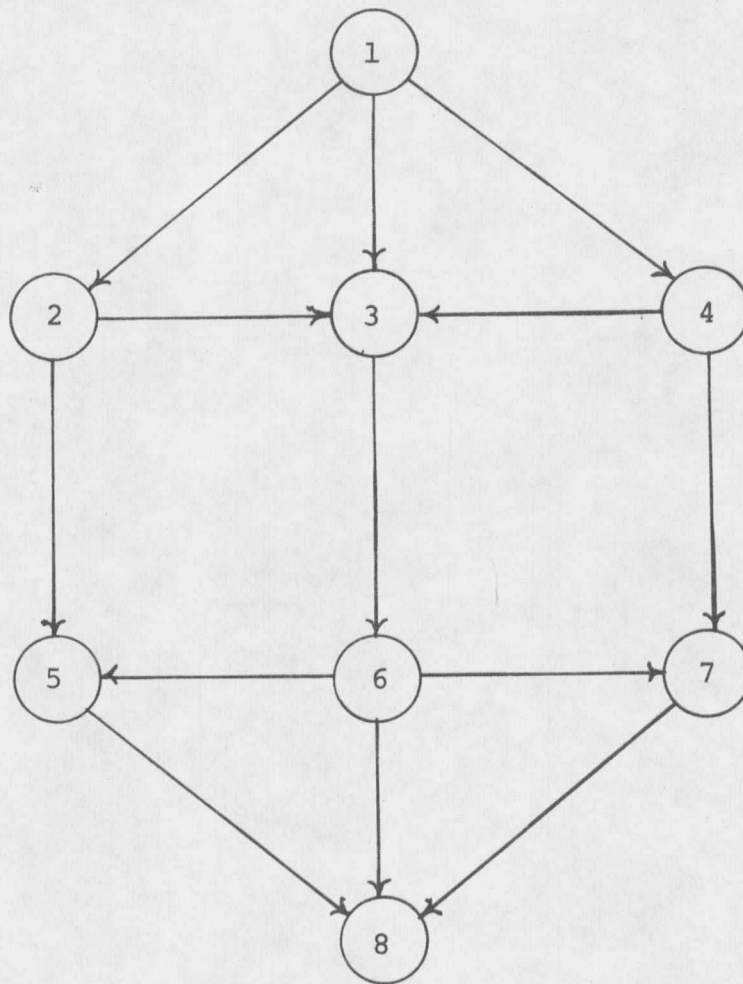


Figure B.1 A SEC graph used in the sample program.

.....EXAMPLE # (1).....

TO FIND THE EDGES TO INSERT MINIMUM NUMBER OF
TEST POINTS TO OBTAIN 1-DISTINGUISHABILITY
OF GIVEN DIRECTED SEC GRAPH FROM 1 TO N

N = NUMBER OF VERTICES OF GRAPH = 8

M = NUMBER OF EDGES OF GRAPH = 13

LISTING OF EDGES

.....

EDGE # : FROM VERTEX : TO VERTEX

1	1	2
2	1	3
3	1	4
4	2	3
5	2	5
6	3	6
7	4	3
8	4	7
9	5	8
10	6	5
11	6	7
12	6	8
13	7	8

!C(CONNECTIVITY MATRIX OF GRAPH) !
 =====

(ROWS AND COLUMNS OF MATRIX ARE VERTICES OF GRAPH)

	1	2	3	4	5	6	7	8
1	0	1	1	1	0	0	0	0
2	0	0	1	0	1	0	0	0
3	0	0	0	0	0	1	0	0
4	0	0	1	0	0	0	1	0
5	0	0	0	0	0	0	0	1
6	0	0	0	0	1	0	1	1
7	0	0	0	0	0	0	0	1
8	0	0	0	0	0	0	0	0

!R (REACHABILITY MATRIX OF GRAPH) !
 =====

(ROWS AND COLUMNS OF MATRIX ARE VERTICES OF GRAPH)

	1	2	3	4	5	6	7	8
1	1	1	1	1	1	1	1	1
2	0	1	1	0	1	1	1	1
3	0	0	1	0	1	1	1	1
4	0	0	1	1	1	1	1	1
5	0	0	0	0	1	0	0	1
6	0	0	0	0	1	1	1	1
7	0	0	0	0	0	0	1	1

8 0 0 0 0 0 0 0 1

:V (SET OF PAIRS OF VERTICES) :

=====

(WHOSE SEPARATION IS UNDER CONSIDERATION

BY SET (S) OF DIRECTED CUT-SETS . HERE (S) CONTAINS ALL EDGES OF GRAPH

VERTEX PAIR # : VERTEX PAIR

1 2, 4
2 5, 7

: MEASURABLE SETS :

=====

OMEGA (1 X 1) = 1,
OMEGA (2 X N) = 2, 3, 5, 6, 7, 8,

OMEGA (1 X 1) = 1,
OMEGA (3 X N) = 3, 5, 6, 7, 8,

OMEGA (1 X 1) = 1,
OMEGA (4 X N) = 3, 4, 5, 6, 7, 8,

OMEGA (1 X 2) = 1, 2,
OMEGA (3 X N) = 3, 5, 6, 7, 8,

OMEGA (1 X 2) = 1, 2,
OMEGA (5 X N) = 5, 8,

OMEGA (1 X 3) = 1, 2, 3, 4,
OMEGA (6 X N) = 5, 6, 7, 8,

4	0	1	1	1	1	0	1	1	0	0	0	0	0	:	1	0
5	0	1	1	1	1	0	0	0	0	1	0	1	1	:	1	1
6	0	0	0	0	1	1	0	1	0	0	0	0	0	:	0	0
7	1	1	0	1	1	0	1	1	0	0	0	0	0	:	1	0
8	1	1	0	0	0	0	1	1	1	0	1	1	0	:	1	1
9	0	0	0	0	0	0	0	1	1	0	1	1	1	:	0	1
10	0	0	0	0	1	0	0	1	0	1	1	1	1	:	0	1
11	0	0	0	0	1	0	0	1	1	1	1	1	0	:	0	1
12	0	0	0	0	1	0	0	1	1	1	1	1	1	:	0	0
13	0	0	0	0	1	0	0	0	1	1	0	1	1	:	0	1

CYCLIC MATRIX AFTER REDUCTION
 =====

1	1	1	0	1	0	0	0	0	1	0
2	1	1	1	1	0	0	0	0	0	0
3	1	1	1	0	0	0	0	0	1	0
4	0	1	1	1	0	0	0	0	1	0
5	0	1	1	0	0	1	0	1	1	1
7	1	0	1	1	0	0	0	0	1	0
8	1	0	0	1	1	0	1	0	1	1
9	0	0	0	0	1	0	1	1	0	1
10	0	0	0	0	0	1	1	1	0	1
11	0	0	0	0	1	1	1	0	0	1
12	0	0	0	0	1	1	1	1	0	0
13	0	0	0	0	1	1	0	1	0	1

INSERT TEST POINTS (THE MINIMUM NUMBER) ON THE FOLLOWING
EDGES TO OBTAIN 1-DISTINGUISHABILITY OF GIVEN GRAPH

=====

EDGE # :FROM VERTEX : TO VERTEX

6	3	6
5	2	5
8	4	7

*STOP# 0

APPENDIX C

LIST OF DEFINITIONS

	Definition No.	Page
Blocking gate	2.24 . .	63
Collection of generators $\{\Omega^i\}$	3.3 . .	80
Complement of $\Omega(i \times j)$	2.7 . .	36
Connected graph	1.2 (a) . .	21
Connectivity matrix C	4.1 . .	106
Cut-set and directed cut-set of a directed graph	2.15 . .	45
Directed cut-sets covered by an edge . .	2.18 . .	47
Directed cut-set separating i and j . .	2.17 . .	46
Directed edge	2.2 . .	34
Directed (linear) graph	2.1 . .	33
Directed path and directed circuit . . .	2.5 . .	35
D-partition	2.8 . .	38
Entry vertex	1.6 . .	24
Essential vertex and separate graph . .	1.1 . .	21
Exit vertex	1.7 . .	24
Fault	2.3 . .	34
Fault vertex	2.4 . .	34
Final covering matrix Z	4.8 . .	129
k-distinguishability	2.9 . .	41
Linearly independent sets	2.23 . .	60
Link subgraph	1.5 . .	23
Matrix C^k	4.3 . .	107
Matrix P	4.7 . .	128
Matrix Q	4.6 . .	127
Maximal connected subgraphs	2.16 . .	46
Maximal strongly connected (M.S.C.) subgraph	1.4 . .	23

	Definition No.	Page
Measurable set $\Omega(i \times j)$	2.6	36
Measurable sets by gate-set E	2.26	64
Measurable set under E	2.19	51
Minimum set for $\Omega(i \times j)_E$	2.27	66
1-distinguishable under set E of edges	2.20	51
1-(i,j)-distinguishable	2.28	67
Ordering of two vertices	2.21	59
Proper subgraph	3.2	75
Reachability matrix R	4.2	106
Ring sum ($\oplus$) of two sets	2.22	60
SEC graph	1.8	27
Set $T = \{S\}$	3.5	89
Set $V = \{(v_i, v_j)\}$	4.4	122
Strongly connected graph	1.2(b)	23
Subgraph of a graph	1.3	23
Symbol $\mathcal{G}(\Omega_1 \times \overline{\Omega}_1)$	2.14	45
Symbol $G(\Omega(i \times j))$ or $G(i, j)$	3.1	75
Symbol $\Omega(e^\pm \times j)$	2.12	44
Symbol $\Omega(G_i)$	2.13	45
Symbol $\Omega(i \times j)$	2.25	64
Symbols $v(e^+)$ and $v(e^-)$	2.11	44
Test point	2.10	44
Total graph	4.5	123
2N-tuple S	3.4	86

REFERENCES

REFERENCES

- [1] Armstrong, D.B., "On Finding a Nearly Minimal Set of Fault Detection Tests of Combinational Logic Nets", IEEE Trans. on Electronic Computers, Vol. EC-15, February 1966, pp. 66-73.
- [2] Avižienis, A., "An Experimental Self-Repairing Computer", Proc. IFIP Congress, Vol. E29-23, 1968.
- [3] Avondo-Bodio, G., Economic Applications of the Theory of Graphs, (Gordon and Breach, New York, 1962).
- [4] Balas, E., "An Additive Algorithm for Solving Linear Programs with Zero-One Variables", Operations Research, Vol. 13, 1965, pp. 517-546.
- [5] Bell, C.G., and Newell, A., Computer Structures: Readings and Examples, (McGraw-Hill Book Co., New York, 1971).
- [6] Berztiss, A.T., Data Structures: Theory and Practice, (Academic Press, New York, 1971).
- [7] Breuer, M.A., Design Automation of Digital Systems: Theory and Techniques, Vol-1, (Prentice-Hall, Inc., New Jersey, 1972).
- [8] Buchholz, W., "Design Objectives for the IBM STRETCH Computer, New Computers, Rept. from the Manufacturers", ACM Conf., 1957, pp. 59-104.
- [9] Busackers, R.G., and Saaty, T.L., Finite Graphs and Networks: An Introduction with Applications, (McGraw-Hill Book Company, New York, 1965).
- [10] Carnahan, B., Luther, H.A., and Wilkes, J.O., Applied Numerical Methods (John Wiley & Sons, Inc., New York, 1969).
- [11] Cayley, A., "A Theorem on Trees", Quart. J. Math., Vol. 23, 1889, pp. 379-378.

- [12] Chang, H.Y., Manning, E.G., and Metze, G., Fault Diagnosis in Digital Systems, (Wiley-Interscience, New York, 1970).
- [13] Chen, Wai-Kai, Applied Graph Theory, (American Elsevier Publishing Company, Inc., New York, 1971).
- [14] Eldred, R.D., "Test Routines Based on Symbolic Logic Statements", Journal of ACM, Vol. 6(1), January 1959, pp. 33-36.
- [15] Forbes, R.E., Rutherford, D.H., Stieglitz, C.B., and Tung, L.H., "A Self-diagnosable Computer", Fall Joint Computer Conf., AFIPS Proc., Vol. 27, 1965, pp. 1073-1086.
- [16] Friedman, A.D., and Menon, R.R., Fault Detection in Digital Circuits, (Prentice-Hall, Inc., New Jersey, 1971).
- [17] Gimpel, J.F., "A Reduction Technique for Prime Implicant Tables", IEEE Trans. on Electronic Computers, Vol. EC-14, August 1966, pp. 535-541.
- [18] Glove, F., "A Multiphase-Dual Algorithm for the Zero-One Integer Programming Problem", Operations Research, Vol. 13, No. 6, November-December 1965, pp. 879-919.
- [19] Harary, F., Graph Theory and Theoretical Physics, (Academic Press, New York, 1967).
- [20] _____, and Norman, R.Z., Graph Theory as a Mathematical Model in Social Science, (Institute for Social Research, University of Michigan, Ann Arbor, 1953).
- [21] _____, and _____, and Cartwright, D., Structural Models: An Introduction to the Theory of Directed Graphs, (John Wiley & Sons, Inc., New York, 1965).

- [22] Hennie, F.C., "Fault-Detecting Experiments for Sequential Circuits", Proc. Fifth Annual Symposium on Switching Circuit Theory and Logical Design, 1964, pp. 95-110.
- [23] Huffman, D.A., "The Synthesis of Sequential Switching Circuits", J. Franklin Inst., 1954, pp. 257, 161-190 and pp. 275-303.
- [24] Kaufmann, A., Graphs, Dynamic Programming, and Finite Games, (Academic Press, 1967).
- [25] Kernighan, B.W., and Lin, S., "An Efficient Heuristic Procedure for Partitioning Graphs", The Bell System Technical Journal, February 1970, pp. 291-307.
- [26] Lowe, C.T., "Analysis of Boolean Program Models for Time-Shared, Paged Environments", Communications of the ACM, Vol. 12, No. 4, April 1969, pp. 199-205.
- [27] Luccio, F., "A Method for the Selection of Prime Implicants", IEEE Trans. Electronic Computers, Vol. EC-15, April 1966, pp. 205-212.
- [28] Manning, E., "On Computer Self-diagnosis, Part I: Experimental Study of a Processor, Part II: Generalizations and Design Principles", IEEE Trans. on Electronic Computers, Vol. EC-15, December 1966, pp. 873-881 and pp. 882-890.
- [29] Marshall, C.W., Applied Graph Theory, (Wiley-Interscience, New York, 1971).
- [30] Mayeda, W., Graph Theory, (Wiley-Interscience, New York, 1972).
- [31] _____, and Ramamoorthy, C.V., "Distinguishability Criteria in Oriented Graphs and Their Application to Computer Diagnosis-I", IEEE Trans. On Circuit Theory, Vol. CT-16, No. 4, November 1969, pp. 448-454.

- [32] _____, and _____, "Distinguishability Criteria in Oriented Graphs and Their Application to Computer Diagnosis-II", Proc. of IEEE International Symposium on Circuit Theory, Georgia, 1970.
- [33] McMillan, C., Mathematical Programming, (John Wiley & Sons, Inc., New York, 1970).
- [34] Pierce, W.H., Failure-Tolerant Computer Design, (Academic Press, New York, 1965).
- [35] Poage, J.F., and McCluskey, E.J., "Derivation of Optimum Test Sequences for Sequential Machines", Proc. Fifth Annual Symposium on Switching Circuit Theory and Logical Design", 1964, pp. 121-132.
- [36] Preparata, F.P., Metze, G., and Chien, R.T., "On the Connection Assignment Problem of Diagnosable Systems", IEEE Trans. on Electronic Computers, Vol. EC-16, December 1967, pp. 848-854.
- [37] Prosser, R., "Applications of Boolean Matrices to the Analysis of Flow-Diagrams", Proc. Eastern Joint Computer Conf., 1959.
- [38] Pyne, I.B., and McCluskey, E.J., "An Essay on Prime Implicant Tables", J. Soc. Ind. Appl. Math., Vol. 9, December 1961, pp. 604-631.
- [39] Ramamoorthy, C.V., "Fault-Tolerant-Computing: An Introduction and an Overview", IEEE Trans. on Computers, Vol. C-20, No. 11, November 1971, pp. 1241-1244.
- [40] _____, "A Structural Theory of Machine Diagnosis", Spring Joint Computer Conf., AFIPS Conf. Proc., Vol. 30, Washington, D.C., 1967, pp. 743-756.
- [41] _____, and Mayeda, W., "Computer Diagnosis using the Blocking Gates Approach", IEEE Trans. on Computers, Vol. C-20, No. 11, November 1971, pp. 1294-1299.

- [42] _____, and Chang, Lih-Chung, "System Segmentation for the Parallel Diagnosis of Computers", IEEE Trans. on Computers, Vol. C-20, No. 3, March 1971, pp. 261-270.
- [43] _____, "Analysis of Graphs by Connectivity Considerations", Journal of the ACM, Vol. 13, No. 2, March 1971, pp. 261-270.
- [44] _____, "The Analytic Design of a Dynamic Look-Ahead and Program Segmentation System for Multi-programmed Computers", Proc. National Conference of ACM, 1966.
- [45] _____, Generating Functions of Abstract Graphs with Systems Applications, (Ph.D. Thesis, Harvard University, May 1964).
- [46] Roberts, L.G., "Multiple Computer Networks and Intercomputer Communication", ACM Symp. on Operating System Principles, Gatlinburg, Tenn., October 1967.
- [47] Rosenblatt, D., "Aggregation in Matrix Models of Resource Flows II.--Boolean Remation Matrix Methods", The American Statistician, June 1967, pp. 32-37.
- [48] Roth, J.P., "Diagnosis of Automata Failures: a Calculus and a Method", IBM J. of Res. and Dev., July 1966, pp. 278-291.
- [49] Russell, J.D., and Charles, R.K., "Structural Factors in the Fault Diagnosis of Combinational Network", IEEE Trans. on Computers, Vol. C-20, No. 11, November 1971, pp. 1276-1285.
- [50] Sellers, F.F., Hsiao, M.Y., and Bearnson, L.W., "Analyzing Errors with the Boolean Difference", IEEE Trans. on Computers, Vol. C-17, July 1968, pp. 676-683.
- [51] Seshu, S., The Logic Organizer and Diagnosis Programs, (Co-ordinated Science Laboratory).

- [52] _____, "On the Improved Diagnosis Program",
IEEE Trans. on Electronic Computers, Vol. EC-14,
February 1965, pp. 76-79.
- [53] _____, and Freeman, D.N., "The Diagnosis of
Asynchronous Sequential Switching Systems",
IRE Trans. on Electronic Computers, Vol. EC-11,
August 1962, pp. 459-465.
- [54] _____, and Reed, M.B., Linear Graph and Electrical
Networks, (Addison-Wesley, Mass., 1961).
- [55] Stephen, Y.H. Su, and Yun-Chung Cho, "A New Approach
to the Fault Location of Combinational Circuits",
IEEE Trans. on Computers, Vol. C-21, No. 1,
January 1972, pp. 21-30.
- [56] Trong, H.C., Switching Circuits, Theory and Logic
Design, (Addison-Wesley Publishing Company,
Malopark, California, 1972).
- [57] Wilcox, R.H., and Mann, W.C., Redundancy Techniques
for Computing System, (SPARTAN Books, Washington,
D.C., 1962).
- [58] Mathur, F.P., "On Reliability Modeling and Analysis
of Ultrareliable Fault-Tolerant Digital Systems",
IEEE Trans. on Computers, Vol. C-20, No. 11,
November 1971, pp. 1376-1381.
- [59] Breuer, M.A., "Testing for Intermittent Faults in
Digital Circuits", IEEE Trans. on Computers,
Vol. C-22, No. 3, March 1973.

MONTANA STATE UNIVERSITY LIBRARIES

3 1762 100111

3 1762 10011142 4

[REDACTED]
D379

[illegible]

D378
P272
Cap 2

PLACE BINDERY