To the Graduate Council:

I am submitting herewith a thesis written by Mary Karen Lyle entitled "The Calculation Of Reliability For Complex Systems." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Industrial Engineering.

D.H. Hutchinson, Major Professor

We have read this thesis
and recommend its acceptance:

K. E. Kirby

C. W. Aitken

Accepted for the Council:

Vice Provost
and Dean of The Graduate School

THE CALCULATION OF RELIABILITY FOR COMPLEX SYSTEMS

A Thesis

Presented In Partial Fulfillment of the

Master of Science

Degree

The University of Tennessee, Knoxville

Mary Karen Lyle

December 1987

ACKNOWLEDGEMENTS

The author wishes to thank her mother and father, Elizabeth D. and Benjamin F. Lyle, for their support and their inspiration in her pursuit of this degree. She also wishes to express her deepest gratitude to her major professor Dr. D.H. Hutchinson for all of his patience and his intellectual guidance.

ABSTRACT

This study develops two programs for the calculation of the reliability of small complex systems. Small is defined as 10 or less components, 10 or less paths or 10 or less cuts. A method for reducing large systems is presented with an application.

Both programs are written in VAX/VMS Basic. Program ESPACE enumerates all possible system states. The user inputs the components, their reliabilities and the min cut set. The program determines which states result in system failure from the cut set and sums the reliabilities of those states to find system reliability.

For program ADDON the user inputs component reliabilities and either the min path set or the min cut set. The program manipulates string expressions to produce the terms of the addition theorem, applies the idempotency law (A intersection A is A) and calculates the signed terms which are summed to give system reliability.

Systems with common cause failure are not considered.

TABLE OF CONTENTS

CHAPTER		PAGE
I.	INTRODUCTION	1
II.	LITERATURE SURVEY	8
III.	ESPACE	20
IV.	ADDON	27
V.	SYSTEM REDUCTION	34
VI.	CONCLUSIONS	50
LIST OF REFERENCES		52
APPENDIXES		56
APPENDIX A.	PROGRAM ESPACE	57
APPENDIX B.	PROGRAM ADDON	70
VITA		82

LIST OF FIGURES

FIGURE		PAGE
1.	SERIES SYSTEM	2
2.	PARALLEL SYSTEM	2
3.	SERIES-PARALLEL SYSTEM	4
4.	5-COMPONENT SYSTEM	4
5.	7-COMPONENT SYSTEM	4
6.	5-COMPONENT COMPLEX SYSTEM	20
7.	OUTPUT FROM ESPACE RUN OF COMPLEX SYSTEM Figure 6	23
8.	FLOW DIAGRAM FOR PROGRAM ESPACE	24
9.	5-COMPONENT COMPLEX SYSTEM	28
10.	OUTPUT FROM ADDON RUN OF COMPLEX SYSTEM Figure 9	29
11.	7-COMPONENT COMPLEX SYSTEM	30
12.	OUTPUT FROM ADDON RUN OF COMPLEX SYSTEM Figure 11	30
13.	FLOW DIAGRAM FOR PROGRAM ADDON	31
14.	FAULT TREE FOR AUTOMOTIVE BRAKE FAILURE	35
15.	FAULT TREE FOR AUTOMOTIVE REAR BRAKE SYSTEM FAILURE	36
16.	STATISTICS ABOUT COMPONENTS FOR REAR BRAKE FAULT TREE Figure 15	36
17.	OUTPUT FROM ADDON RUN OF REAR BRAKE (Figure 16) STATISTICS	37
18.	REDUCED FAULT TREE FOR AUTOMOTIVE BRAKE FAILURE	38
19.	STATISTICS ABOUT THE COMPONENTS OF REDUCED TREE (Figure 18)	38

FIGURE PAGE

20. OUTPUT FROM ADDON RUN OF THE REDUCED TREE STATISTICS OF Figure 19 39

21. STATISTICS FOR THE UNREDUCED FAULT TREE OF Figure 14 43

22. OUTPUT FROM ADDON RUN OF THE STATISTICS FOR Figure 21 44

23. FAULT TREE FOR AUTOMOBILE FAILURE TO START . 44

24. FAULT TREE FOR AUTOMOBILE BATTERY FAILURE .. 45

25. REDUCED FAULT TREE FOR AUTOMOBILE FAILURE TO START 45

26. STATISTICS ABOUT THE COMPONENTS BATTERY FAILURE IN Figure 24 46

27. STATISTICS ABOUT THE COMPONENTS REDUCED TREE OF Figure 25 46

28. OUTPUT FROM ESPACE RUN OF BATTERY FAILURE STATISTICS IN Figure 26 47

29. OUTPUT FROM ADDON RUN OF REDUCED TREE STATISTICS IN Figure 27 48

30. STATISTICS ABOUT THE COMPONENTS OF NON REDUCED TREE IN Figure 14 AND THE ADDON PROGRAM OUTPUT FOR THIS SYSTEM 49

CHAPTER I

INTRODUCTION

The determination of the reliability of a system consisting of several independent components is based upon the representation of the system in the form of a series, parallel or series-parallel configuration. This representation is an expression of the logical relationship among the system components. Calculation of the reliability uses the addition and multiplication theorems of basic probability according to the logic of the configuration.

For example, assume a water pipe with four independent valves. It is represented as a series system shown in Figure 1. The configuration implies that all four valves must be opened in order for water to flow thru the system. The upper case letter A represents the probability component A works.

Or suppose an airplane has four independent engines, this would be drawn as the parallel system shown in Figure 2. This configuration implies that the aiplane can fly as long as any one or more of the four engines are working.

The multiplication theorem is applied to the series system of Figure 1 since flow through the system occurs if A works and B works and C works and D works and this



Figure 1

SERIES SYSTEM

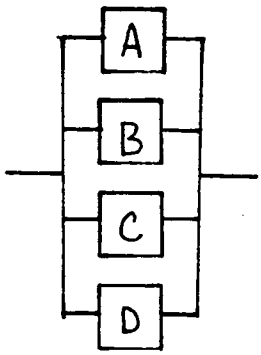


Figure 2

PARALLEL SYSTEM

is represented by the following equation:

$$R_s = R_A \times R_B \times R_C \times R_D$$

The addition theorem applies to the system of Figure 2 since the system works if A works or B works or C works or D works, or;

$$R_s = R_A+R_B+R_C+R_D-R_{AB}-R_{AC}-R_{AD}-R_{BC}-R_{BD}-R_{CD}+R_{ABC}+R_{ABD}+R_{ACD}+R_{BCD}-R_{ABCD}$$

If valves B and C in Figure 1 have relatively low probabilities of working, additional components can be inserted at B and C to give a series-parallel representation as shown in Figure 3. The corresponding reliability expression is:

$$R_s = R_A \times (R_{B_1}+R_{B_2}+R_{B_3}-R_{B_1B_2}-R_{B_1B_3}-R_{B_2B_3}+R_{B_1B_2B_3}) \times (R_{C_1}+R_{C_2}-R_{C_1C_2}) \times R_D$$

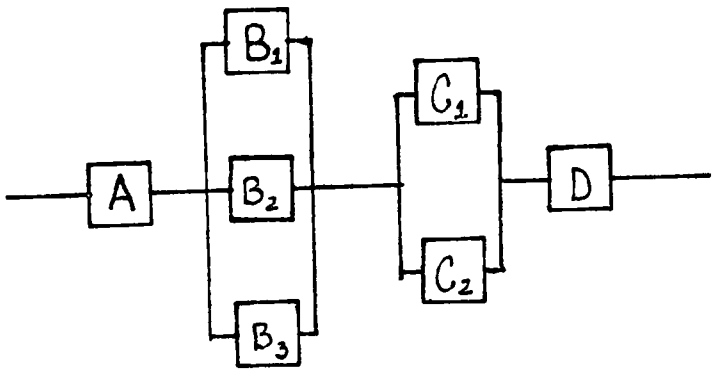


Figure 3

SERIES-PARALLEL SYSTEM

However, some systems can not be represented as series, parallel or series-parallel. These are called complex systems. Their evaluation involves more complicated analysis methods such as event-space, decomposition and min path/ min cut. Event-space is a method which enumeratates all possilble combinations of component failures. The combinations which result in system failure are noted and the sum of the probabilities of the failure combinations is the probability of system failure. Each possible failure combination is considered logically as one point in a sample space. With 4 components the sample space will contain 2^4 = 16 points. In the Decomposition method, a key element is selected

and removed from the system. Apparently this is done by "intuition." Because this method has a subjective structure, it is not widely used and not further mentioned here. The min cut/min path method seems to be the most widely applied because the others become very awkward to use as the number of independent components increases.

The cut sets for a complex system (shown as Figures 4 & 5) are those basic events whose simultaneos occurrence results in system failure. If AB is a cut set and ABD is also a cut set, the set ABD can be "reduced" to AB and still insure system failure. A cut set is minimal (min cut) if it cannot be reduced.

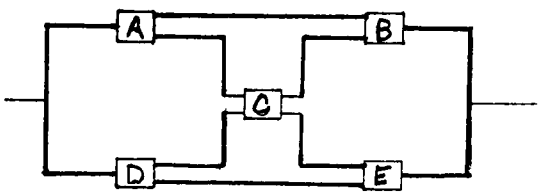


Figure 4

5-COMPONENT SYSTEM

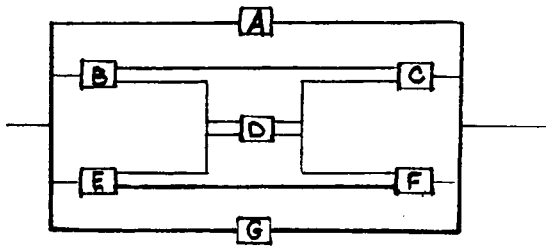


Figure 5

7-COMPONENT SYSTEM

A path set is a set of components whose occurrence assures that the system works. A path set is minimal if it cannot be reduced and still remain a path set. A minimal path set (min path) is a set of elements whose functioning insures the functioning of the system.

For Figure 4 the min cuts are AD,BE,ACE,DCB and the min paths are AB,DE,DCB and ACE. Path DCE results in a working system but it can be reduced to DE.

One reason min path/min cut is important is the development of Fault Tree Analysis (FTA). A Fault Tree is a diagram showing the relationship of the system components to a particular event which is called the top event. The top event in the water flow system of Figure 3 might be the failure of water to flow thru the system. Fault tree analysis begins with the definition of an event (desirable or undesirable) and traces this event down through the system to the basic identifiable events. This top down approach hopefully includes every possible reason - component failure, operator error, power failure, etc. - for the occurrence of the top event. One value of FTA is that it requires a total system view. The engineers who designed the system can prepare a fault tree to show how the failure of a given component or combination of components can lead to the occurrence of the top event. Fussel and Vessely developed the min cut algorithm for a fault tree. This algorithm, which has

been computerized, starts with the fault tree and produces the set of all min cuts.

Recently the obvious application of min cut network theory to FTA has been developed. The logic of the system is represented in the fault tree. From the fault tree a network of nodes and arcs is drawn and algorithms to calculate system reliability from the network have been developed.

This study focuses upon an intermediate point in the problem of calculating system reliability. It looks at the system at the time the min paths or min cuts have been determined. No network ideas have been considered. If the system is relatively simple, the min cuts or paths can be determined directly. If a fault tree has been constructed by the system designers, it is assumed that the well known min cut algorithm has been applied to produce either the min cuts from the original fault tree or the min paths from the dual of the fault tree. The dual of a fault tree is the tree produced when the "and" and "or" gates of the original tree are interchanged.

Two approachs to the determination of system reliability are developed and compared. The first is a computer calculation which uses the enumeration of the events leading to system failure and the second is a computer calculation using directly the min cuts or min paths. The obvious limitation for calculations of this

type are system size. Both types of calculations are applied to a reduced system. Instead of an algorithm which deals with very large systems, groups of components in either a series-parallel diagram of a fault tree are reduced to a single value and successive system reductions lead to a reduced system, small in size and to which the small system computer programs can be applied.

CHAPTER II

LITERATURE SURVEY

The literature contains a structured set of articles on the computation of the reliability of complex systems beginning with a 1969 article by P. Jensen and M. Bellmore (1). They are listed in the bibliography and considered here. The methods of analysis include the reduction of minimal cut sets, splitting the min cuts into subsets which will cut down on the number of cancelling terms in the reliability expression, a computer program that will reduce the core requirements and computer time by determining the success paths of a system and then modifying them to be mutually disjoint, and a method that will reduce of the number of terms in the reliability expression by a symbolic terminal-pair reliability analysis. There are also methods in which Monte Carlo Simulation is used to model the time to failure reliability expression.

Jensen and Bellmore (1) present a technique by which the minimal cuts can be found. After finding these cuts, they apply Esary and Proschan's lower bound approximation to reliability that requires the enumeration of these smaller sets of cuts through the system. This method is referred to as the minimal cut approximation to reliability.

Boolean algebra is a tool many of the authors use to determine their algorithms for reliability analysis and several programs have been created that will reduce the amount of computation required in the reliability analysis. In 1973 Fratta and Montanari (3) state in their findings that;

> Given the set of all simple paths between two nodes in a network, the terminal reliability can be symbolically computed by transforming a Boolean sum of products into an equivalent form in which all terms are disjoint. They represent a communication network by an oriented graph with weighted arcs and unweighted nodes. The nodes represent stations, assumed to be completely reliable; the weighted arcs represent direct unreliable connections, that is, connections which are available on the average only for a given percentage of time. The weight assigned to each arc is its probability of existance and is also known as the arc reliability.
>
> And also:
>
> Boolean algebra has been used to find the probability of communication between a pair of nodes in a network starting with a Boolean product corresponding to simple paths between the pair of nodes and making them disjoint (mutually exclusive).

Reliability theory and its foundation in terms of coherent structures is "success-event-oriented." In event tree theory the system event constituting the top event may be system success or system failure or any other system event which can be analyzed.

A coherent system is one that meets the following criteria: (this is taken from P.A. Jensen and M.

Bellmore, "An Algorithm to Determine the Reliability of a Complex System")

> 1). when a group of components in the system is failed causing the system to be failed, the occurrence of any additional failure or failures will not return the system to a successful condition;
>
> 2). when a group of components in the system is successful, the system will not fail if some of the failed components are returned to the successful condition;
>
> 3). when all the components in a system are successful the system is successful;
>
> 4). when all the components in a system are failed the system is failed;

Singh and Billington (4) developed in 1974 a new method to determine the failure frequency of a complex system in which they "consider the reliability evaluation of a complex maintainable system using a cut set approach." They use a reliability block diagram and make no reference to the number or configuration of the components.

In 1975, a fast algorithm for reliability evaluation was given by Aggarwal, Misra and Gupta (5).

> An algorithm is developed to obtain a simplified reliability expression for a general network. All the success paths of the network are determined; then they are modified to be mutually disjoint. The reliability expression follows directly from the disjoint paths. The algorithm is easy and computationally economical. The reliability expression involves fewer terms and the arithmetic operations than any of the existing methods.

This paper makes no reference to the size of the system or number of components.

"Determination of Reliability Using Event Based Monte Carlo Simulation," written in 1975 by Kamat and Riley (6), combines reliability flow graph representation and Monte Carlo simulation. Their method does not generate failure times for the system but instead generates the failure for the individual components and uses these failures to trip signal a system failure. The distribution approximation is at the component level, thus reducing the possible error in the resulting reliability estimate.

In 1976 Lin, Leon and Huang (7) developed a new algoritm for symbolic reliability analysis. Twenty or fewer paths are evaluated. This paper makes no attempt to generate mutually exclusive events from the set of paths or cut sets but uses a technique to greatly reduce the number of terms in the reliability expression. The method applies to system graphs with unreliable branches or nodes. Each branch is directed or undirected.

Kumanto, Tanaka and Inoue (8) developed in 1977 a Monte Carlo Simulation method by applying variance-reducing techniques. The method can be applied to large complex systems consisting of reliability block diagrams or fault trees.

> Two binary functions are introduced: one dominates the system structure and the other is dominated by the structure function. These functions can be constucted easily by using parts of path sets and cut sets of the system. Through the use of these binary functions, two variance reducing techniques (control variate and importance sampling) are applied to the Monte Carlo evaluation of the system reliability.

This method gives a reliability estimate with a smaller variance than the crude Monte Carlo method.

A top down algorithm for obtaining prime implicant sets of non-coherent fault trees was given in 1978 by H. Kumamoto and E.J. Henley (11). A non-coherent fault tree is one that contains other gates than the AND or OR as in coherent trees. These are gates such as EOR (either this component or that component but not both) and the NOT gate. The sets for such a tree are referred to as the prime implicant sets.

> A complete set of prime implicant sets gives all the nodes of system failure, and a qualitative and/or quantitative system analysis becomes possible. For example, a system is improved by removing the sets containing only one or two primary events. The system reliability and availability can be evaluated by the inclusion-exclusion formulas when a stochastic property of primary events is given. The exact system reliability can be calculated by converting the prime implicant sets to disjoint probability algebraic product terms.

A core efficient cut set algorithm, FATRAM, was developed in 1978 by D.M. Rasmuson and N.H. Marshall (12). They present a program that will efficiently

produce the cuts sets required for fault tree analysis. This program is intended to make a higher utilization of computer memory. However, this program does not evaluate the system reliability.

In 1979 J. A. Abraham (13) developed an algorithm which will rapidly produce, using "boolean algebra to find the probability of communication between a pair of nodes in a network by starting with a boolean product corresponding to simple paths between the pair of nodes and making them mutually exclusive," the disjoint products. Abraham considers the failures on the edges of the network and the algorithm can be used to derive the failures at the nodes. Abraham does not account for the the actual number of components in the complex system. In his paper entitled "An improved algorithm for network reliability", he states;

> The n product terms corresponding to the simple paths from the source node to the termination node are presumed known. At step k of the algorithm, the product Sk is made disjoint with each of the products S1,S2,S3,...,Sk-1. In this process, Sk expands to a set of disjoint products PDk, each product in this set being disjoint with S1,S2,S3,...,Sk-1. Thus the crux of the problem is the manner in which some product Pi is made disjoint with a product Sj.

"Recursive disjoint products, inclusion-exclusion and min-cut approximations," written in 1980 by M.O. Locks (14), references the paper by J.A.Abraham (13), and Locks says;

> Abraham presented an algorithm which drastically reduces the amount of computation needed to obtain the disjoint sums. Given the complete list of minimal success paths for an s-coherent system, the algorithm recursively decomposes the Boolean polynomial for system success into disjoint terms stepping one minimal path at a time; the systems reliability R is then computed by summing the probability of the terms.

Locks points out that recursive inclusion-exclusion (called IE), also known as Poincares Theorem, gives the same result as Abraham, however, the probability polynomial is different.

> The system success probability R is derived recursively from the Boolean logic polynomial whose terms are the minimal paths by mixing and matching 1-valued indicators stepping one min path at a time.

These terms are between alternating plus and minus signs instead of the additive approach of Abraham. Hence, the accumulated probability is the summation of the probabilities of the terms. Disjoint Products (called DP) builds the system reliability function for R by accumulating the probabilities of the "m" minimal paths one min path at a time.

The IE method delivers the upper and lower bounds on a system reliability which converge to the exact system reliability.

For a very large complex system, it is not practical to use exact methods with full information of all minimal states in the system. This requires a very large number

of terms in the polynomial. Because they provide a good standard of comparison, exact methods are quite attractive. Locks did not conduct comparative computer experiments for recursive IE verses the recursive DP due to the difficulty of comparing matters such as size of the polynomial, number of comparisons, ease of programming or computer time. He states that to do a more complete story on the advantages and disadvantages of IE verses DP would require more work and computer experimentation.

"A Compilation Technique for Exact System Reliability" is given by Keun K. Lee (15) in 1981.

> An algorithm is presented for evaluating exact system reliability without using a Boolean polynomial equation. A Boolean expression is converted to polish post fix notation. System reliablility is computed by converting the post fix notation into a sequence of arithmetic and stack operations.

The system reliability is the result of the success or failure of the components and their structure throughout the system. There are 2^m additions and subtractions and $(n \times 2^m)$ multiplications to perform (m being the paths and n being the number of components). He will find either the failure or the success (from the dual of the fault tree) depending on whether there are more OR gates or if there are more AND gates. This paper addresses the question of number of computations and computer error

magnitude in the evaluation of a fault tree with, say, 1000 events.

There is a considerable amount of attention given to the subject of complex system reliability in a review by F. A. Tillman, C.L. Hwang and M.H. Lee (16) in 1981 on reliability evaluation techniques for complex systems. They refer to several articles in which computer programs are used in determining system reliability.

> All the evaluation techniques used in the papers surveyed have only limited success in solving some large/complex system reliability evaluation problems. Few techniques have been completely effective when applied to large system reliability problems. Algorithms of effective and time saving origin are a common approach in computer aided reliability approximations. They are useful when systems become so large and complex that the computation time has limited feasability.

Minset splitting is a method introduced in 1986 by K.D. Heidtmann (18). This method is designed to improve the current reliability computations by reducing all minimal sets into two subsets. The use of this method avoids the task of inclusions or exclusions in the addition theorem and results in a formula without cancelling terms.

> Because systems with such splitting do not belong to the class of p-graphs, a new subset of coherent structures (those that are solely "and" and "or" gates) and a new algorithm is found so that system reliability computation avoids cancelling terms.

This paper looks at a way to calculate the system

reliability which is more accurate than the standard addition theorem.

In 1986 S.P. Jain and Krishna Gopal (19) developed Network Augmentation for improving the reliability of the network. This is a method which involves improving the reliability of existing links in the network or the addition of new links at precisely the most beneficial stage of the network.

An algorithm for reducing the cut sets in a fault tree was presented by N. Limnios and R. Ziani (20) in 1986. The Mocus computer program contains an algorithm in which the cut sets of a fault tree are minimized. Using a fault tree to arrive at all possible minimal cut sets is a major step. Many algorithms have been developed to produce the min cut sets and removal of the non-minimal sets is quite time consumming.

An algorithm for exact fault tree probabilities without the use of cut sets was developed by L.B. Page and J.E. Perry (21) in 1986. TPDD is a top down recursive algorithm and declares that fault tree probabilities do not depend upon pre-determined cut sets. TDPP handles non-coherent fault trees with replicated/repeated inputs. This applies to moderate size trees (50 to 100 nodes,i.e., 50 to 100 components).

> For in an exact top-down approach to cut-set determination, both the sizes of individual cut sets and the length of the list of cut sets grow

> as one works down through the tree. By contrast, in top-down probability calculations no such dynamic memory requirement is present. No lists need be maintained. Their algorithm works directly with probabilities, which are numbers. No symbolic manipulation is required. The only dynamic memory allocation is for the recursive stack, and this is proportional only to the depth of the fault tree and presents small demands even for a microcomputer.

"Reliability analysis of networks with capacity-constraints and failures at branches and nodes," by W.J. Rueger (22) in 1986, declares that the minimal paths or cuts of a system are not clearly stated. A new method for reliability analysis of stochastic flow networks is proposed. Nodes as well as branches can have limited capacities and can fail.

> The method builds a branching tree. The tree nodes are the disjoint terms of a symbolic reliability expression. Ruegers method shows how "to find the probability function and mean of the network performance."

The system must be monotonic. When the elements of a set are failed then the result is also a complete system failure. The elements are statistically independent.

Finally, in 1986, Schneeweiss (23) covers the subject of failure frequency of systems with dependent components.

> The conventional approach to determining system failure does not allow for dependent components. A novel approach presented (in his paper) shows clearly which information is needed to treat the case of independence. The information is more

complicated than that needed for system availability determination, yet it is the same type, viz, joint probability of higher moments of joint binary processes. The main result is: system failure frequency can be found as the limiting value for expectation of the product of the negated fault tree output at t and the usual output at (t + Δt), all divided by Δ t.

With Schneeweiss's paper the editor noted that it was not feasible to reconcile all the differences between the author and referees. The differences centered on the novelty of his approach and not the correctness.

Summary of the literature

The literature prior to about 1970 seems to have established methods for the determination of complex system reliability. For the last 15-20 years almost all of the efforts directed at the problem are more efficient algorithms for computation. The basic probability problem is usually converted to a network problem or boolean algebra problem for a new algorithm.

Several articles are directed at attempting to compute very large system reliabilities at one pass. In practice, large systems of, say, 1000 components, exist but are analyzed in terms of subsystems for which a reliability has been calculated. Not one of the articles considered here were found to consider the problem of common cause failures.

CHAPTER III

ESPACE

Any system can be evaluated by the Event-Space method. This method uses combinatorial methods to produce a list of all possible "failed" states of the system: beginning with 1 failure up to n failures. The event space method is an exhaustive listing of all possible states of the n components in the system. Consider Figure 6 shown below, without any consideration of the paths between components, it is clear that $2^5 = 32$ possible states exist. The empty set, ABCDE, is one of the 32 but cannot result in system failure so that 32 - 1 = 31 states exist which could result in a system failure.

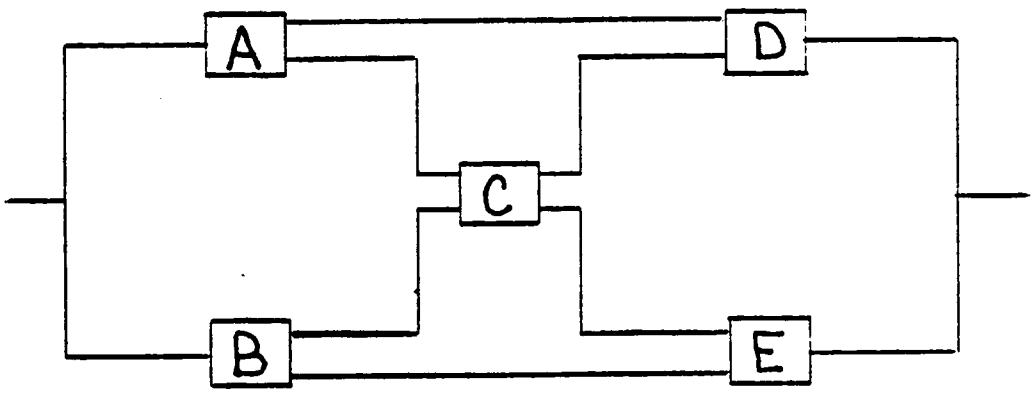


Figure 6

5-COMPONENT COMPLEX SYSTEM

The Espace program has to be told which states result in system failure. This is done by giving the program the min cut sets. The program determines each state and compares it to the min cut sets. If the state contains any min cut, the program stores that state (event) into a matrix and calculates the probability of its occurrence.

The first step is to produce the combinations of all possible failures, for this example there are five components and therefore the event space is $2^5 = 32$. If any one of these events contains a minimal cut set then this event is a system failure state and its probability of occurrence is (using the multiplication theorem for n independent components) the probability each component is either failed or working.

For Figure 6, the minimal cuts are those cuts that upon occurrence, will result in system failure. For this example the min cut sets are AB, DE, ACE and BCD. If cut set AB occurs the probability of its occurrence will be equal to the failure of component A times the failure of component B times the product of the probability of successes of the components C,D and E. Table 1 is a listing of all possible failure combinations for the components. For easier notation, this listing produces the state of each component for each event, this is done by denoting a "failed" componet with a capital letter and a "functioning" component with a lower case letter (e.g.

event AbcDe implies components A and D are failed, while components B,C and E are functioning). The probability of the occurrence AbcDe is A x b x c x D x e, where an upper case letter has the probability 1 -p (p is the probability the component works).

After all possible combinations are checked for cut sets, those that contain a minimal cut set are denoted with an asterik(*) and stored into an array for the future calculation of the probability of occurrence and system reliability. In Table 1, those events containing cut sets are marked by an *. Note that the computer uses only upper case characters and only the asterik indicates events resulting in system failure.

Figure 7 is a listing of the statistics about the data for the components shown as Figure 6. Figure 7 also lists the computer printout of the results for this data run. The output is given to be a listing of the asterik marked combinations and there corresponding probability of occurence. Note that the computer will only list the failed components and not the working ones.

Program ESPACE will produce the combinations of failed components and compute their probability of occurrence and the resulting reliability of the system. Figure 8 is a logic flow diagram for program ESPACE and the corresponding computer commands are given at each flow block.

1 Failure	2 Failures	3 Failures	4 Failures	5 Failures
Abcde	ABcde*	ABCde*	ABCDe*	ABCDE*
aBcde	AbCde	ABcDe*	ABCdE*	
abCde	AbcDe	ABcdE*	ABcDE*	
abcDe	AbcdE	AbCDe	AbCDE*	
abcdE	aBCde	AbCdE*	aBCDE*	
	aBcDe	AbcDE*		
	aBcdE	aBCDe*		
	abCDe	aBCdE		
	abCdE	aBcDE*		
	abcDE*	abCDE*		

Table 1

EVENT SPACE FOR 5-COMPONENT SYSTEM (Figure 6)

NUMBER	COMPONENT	PROBABILITY OF SUCCESS p	PROBABILITY OF FAILURE q
1	A	.9	.1
2	B	.98	.02
3	C	.97	.03
4	D	.95	.05
5	E	.97	.03

THE CUT SETS ARE

AB
DE
ACE
BCD

COMBINATION	PROBABILITY OF OCCURENCE
AB	.00178771
DE	.00128331
ABC	.00005529
ABD	.00009409
ABE	.00005529
ACE	.00008379
ADE	.00014259
BCD	.00003619
BDE	.00002629
CDE	.00003969
ABCD	.00000291
ABCE	.00000171
ABDE	.00000291
ACDE	.00000441
BCDE	.00000081
ABCDE	.00000009
TOTAL	.00360698

RELIABILITY OF SYSTEM = .99639302

Figure 7

OUTPUT FROM ESPACE RUN OF COMPLEX SYSTEM Figure 6

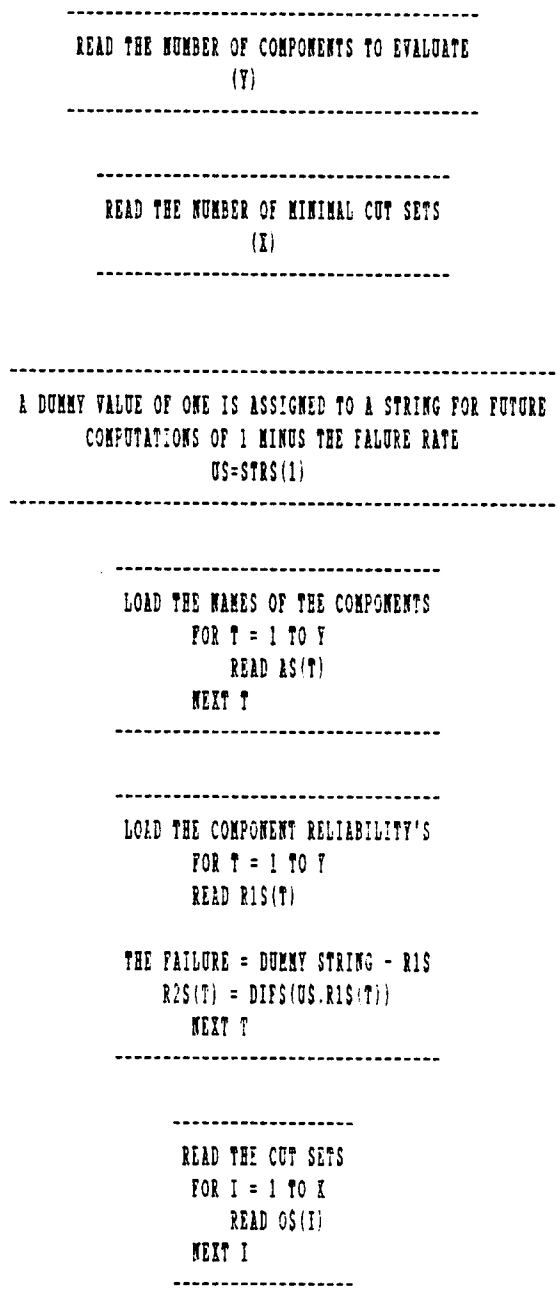


Figure 8

FLOW DIAGRAM FOR PROGRAM ESPACE

```
-----------------------------------------------
PRODUCE THE EVENTS IN WHICH ONE COMPONENT FAILS
              A1 = 1
              FOR A = A1 TO Y
                  K = K + 1
                  BS(K) = AS(A)
                  GOSUB TO SEE IF THIS
                       EVENT CONTAINS MIN CUT
              NEXT A
-----------------------------------------------

-----------------------------------------------
PRODUCE THE EVENTS IN WHICH TWO COMPONENTS FAIL
           B1 = 1
           A1 = 2
           FOR B = B1 TO Y-1
              FOR A = A1 TO Y
                  K = K + 1
                  BS(K) = AS(B)+AS(A)
                  GOSUB TO SEE IF THIS
                       EVENT CONTAINS MIN CUT
              NEXT A
              A1 = A1 + 1
           NEXT B
-----------------------------------------------

-----------------------------------------------
PRODUCE THE EVENTS IN WHICH THREE COMPONENTS FAIL
        C1 = 1
        B1 = 2
        A1 = 3
          FOR C = C1 TO Y-2
             FOR B = B1 TO Y-1
                FOR A = A1 TO Y
                   K = K + 1
                   BS(K) = AS(C)+AS(B)+AS(A)
                   GOSUB TO SEE IF THIS
                        EVENT CONTAINS MIN CUT
                NEXT A
                A1 = A1 + 1
             NEXT B
             A1 = B1 + 2
             B1 = B1 + 1
          NEXT C
-----------------------------------------------
```

Figure 8 (Continued)

```
SUBROUTINE WHICH WILL CHECK THE GIVEN ESPACE COMBINATION
FOR CONTAINING A MINIMAL CUT SET
                S = S + 1
                FOR I = 1 TO X
                    FOR T = 1 TO Y
                        Z=POS(OS(I),AS(T),1)
                        IF Z = 0 THEN RETURN
                        ZS = SEGS(OS(I),Z,Z)
                        M =POS(BS(E),ZS,1)
                        IF M = 0 THEN RETURN
                    NEXT T
                    LS(S) = CHRS(42)
                    GO TO NEXT I
                    LS(S) = SPACES(1)
                    NEXT I
```

```
TEST TO SEE IF A GIVEN EVENT IS MARKED WITH AN *. IF SO THEN COMPUTE
THE PROBABILITY OF THE OCCURRENCE FOR THE EVENT AND TOTAL RELIABILITY
OF SYSTEM
                    FOR I = 1 TO S
                       RELS(I) = US
                       IF LS(I) = CHRS(42) THEN 1690
                       GO TO 1780
                       FOR M = 1 TO Y
                          Z = POS(BS(I),AS(M),1)
                          IF Z = 0 THEN 1740
                          RELS(I) = PRODS(RELS(I),R2S(M),10)
                          GO TO 1750
                          RELS(I) = PRODS(RELS(I),R1S(M),10)
                       NEXT M
                       RS=SUMS(RS,RELS(I))
                     NEXT I
```

```
THIS PROCESS WILL CONTINUE UNTIL THE EVENT OCCURS THAT COMPONENTS
1 THROUGH Y ARE FAILED.
```

Figure 8 (Continued)

CHAPTER IV

ADDON

ADDON is another computer program for reliability evaluations. This program applies the addition theorem of probability to system reliability evaluations. Given the min cut sets, the system unreliability is calculated or given the min path sets the system reliability is calculated. Unlike ESPACE, program ADDON does not have to be told which failures result in system failure because the set of min cuts or min paths contains this information.

If the set of min cuts is known, the occurrence of any one or any combination of these min cuts result in system failure. If any one or combination of the min paths occurs the system must work. The cuts or paths are not mutually exclusive. That is, the system is examined at, say, 200 hours of service and cut ACD has occurred. It is also possible that cuts ADF and BCD have occurred.

System failure can be calculated using the addition theorem. Suppose the cuts are;

AE, ACD, ADF and BCD,

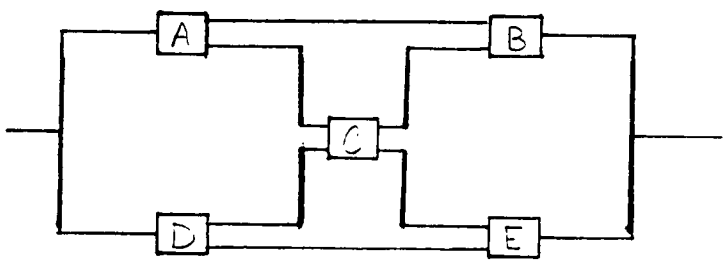
the addition theorem for 4 events is used to calculate F_s, the probability of system failure. The logical principal is that if any one or any combination of the cuts has occured, the system will fail (F_s) and system

reliability (R_S) is;

$$R_S = 1 - F_S$$

Consider the complex system shown as Figure 9. The path sets are easily determined to be AB, DE, ACE and DCB. The program output is shown in Figure 10 giving each term for the addition theorem and a resulting reliability of 0.967023. A more complicated system is shown as Figure 11. The path sets are A,G,EF,EDC,BC and BDF. This system has seven components. The size of an ESPACE program run is determined by the number of components. There would be $2^7 - 1 = 127$ events to be evaluated. ADDON, however, has a program run size determined by the number of paths or cuts. In this case, with 6 cuts, there are $2^6 - 1 = 63$ terms to evaluate. The output is given in Figure 12 with a resulting system reliability of 0.999949.

Figure 13 is a flow diagram for program ADDON. The listing includes the computer commands at each stage of the diagram.



5-COMPONENT COMPLEX SYSTEM

Figure 9

NUMBER	COMPONENT	PROBABILITY OF SUCCESS p	PROBABILITY OF FAILURE q
1	A	.9	.1
2	B	.8	.2
3	C	.95	.05
4	D	.9	.1
5	E	.89	.11

THE MINIMAL PATHS ARE

AB
DE
ACE
BCD

NUMBER	TERM	RELIABILITY OF TERM
1	AB	.72
2	DE	.801
3	ACE	.76095
4	BCD	.684
5	ABDE	-.57672
6	ABCE	-.60876
7	ABCD	-.6156
8	ACDE	-.684855
9	BCDE	-.60876
10	ABCDE	-.547884
11	ABCDE	.547884
12	ABCDE	.547884
13	ABCDE	.547884
14	ABCDE	.547884
15	ABCDE	-.547884
TOTAL		.967023

RELIABILITY EXPRESSION = + AB + DE + ACE + BCD - ABDE - ABCE - ABCD - ACDE - BCDE - ABCDE + ABCDE + ABCDE + ABCDE + ABCDE - ABCDE

RELIABILITY OF SYSTEM = .967023

Figure 10

OUTPUT FROM ADDON RUN OF COMPLEX SYSTEM Figure 9

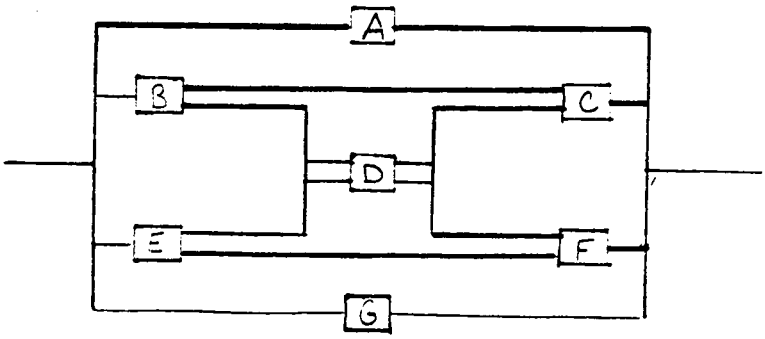


Figure 11

7-COMPONENT COMPLEX SYSTEM

NUMBER	COMPONENT	PROBABILITY OF SUCCESS p	PROBABILITY OF FAILURE q
1	A	.9	.1
2	B	.8	.2
3	C	.7	.3
4	D	.95	.05
5	E	.96	.04
6	F	.86	.14
7	G	.99	.01

THE MINIMAL CUTS ARE

ABEG
ACFG

NUMBER	TERM	UNRELIABILITY OF TERM
1	ABEG	.000008
2	ACFG	.000042
3	ABCEFG	-.000000336
TOTAL		.000049664

UNRELIABILITY EXPRESSION = + ABEG + ACFG - ABCEFG

RELIABILITY OF SYSTEM = .999950336

Figure 12

OUTPUT FROM ADDON RUN OF COMPLEX SYSTEM Figure 11

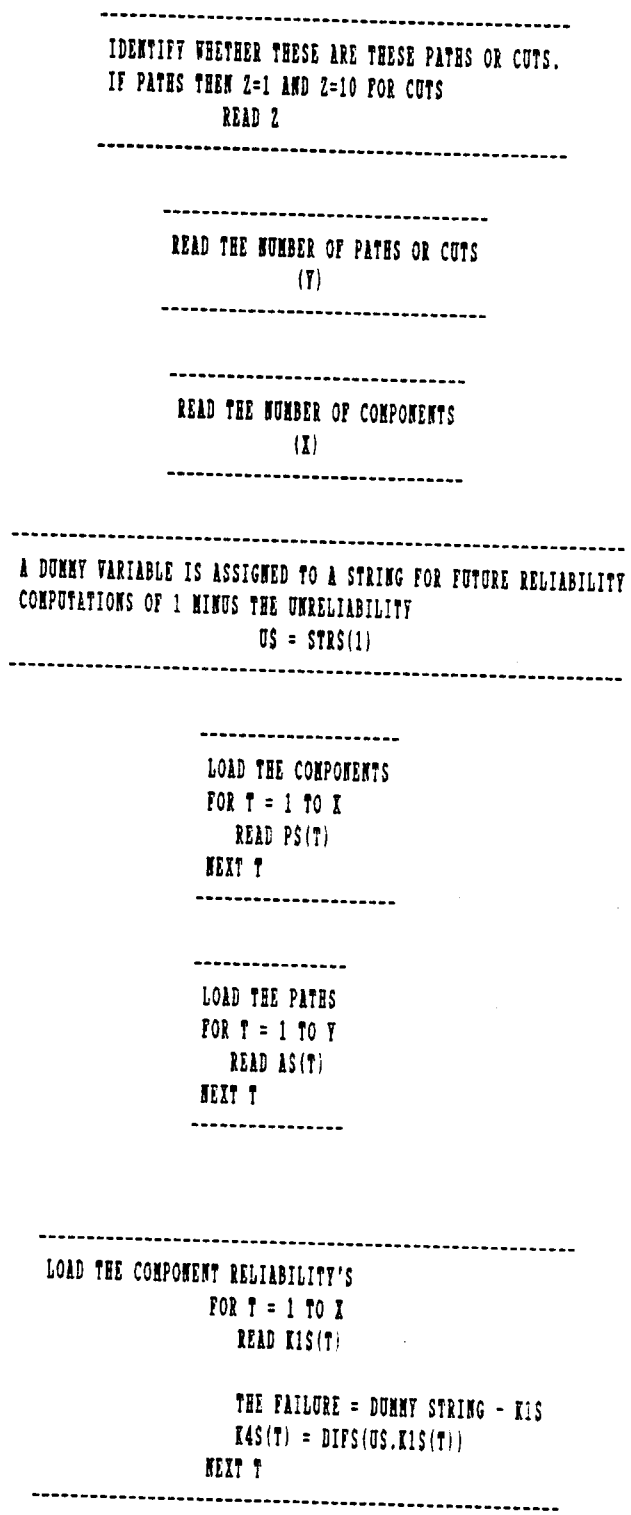


Figure 13

FLOW DIAGRAM FOR PROGRAM ADDON

```
------------------------------------------------------------------
PRODUCE THE TERMS OF THE EXPRESSION IN WHICH PATHS OR CUTS ARE SINGULAR
              A1 = 1
              FOR A = A1 TO Y
                 T9 = 1
                 K = K + 1
                 BS(K) = AS(A)
                 GOSUB TO REMOVE DUPLICATE COMPONENTS
              NEXT A
------------------------------------------------------------------


------------------------------------------------------------------
PRODUCE THE TERMS OF THE EXPRESSION IN WHICH THE PATHS OR CUTS ARE
SUMMED IN SETS OF TWO
              B1 = 1
              A1 = 2
              FOR B = B1 TO Y-1
                 FOR A = A1 TO Y
                    T9 = -1
                    K = K + 1
                    BS(K) = AS(B)+AS(A)
                    GOSUB TO REMOVE DUPLICATE COMPONENTS
                 NEXT A
                 A1 = A1 + 1
              NEXT B
------------------------------------------------------------------


------------------------------------------------------------------
PRODUCE THE TERMS OF THE EXPRESSION IN WHICH THE PATHS OR CUTS ARE
SUMMED IN SETS OF THREE
                C1 = 1
                B1 = 2
                A1 = 3
                FOR C = C1 TO Y-2
                   FOR B = B1 TO Y-1
                      FOR A = A1 TO Y
                         T9 = 1
                         K = K + 1
                         BS(K) = AS(C)+AS(B)+AS(A)
                         GOSUB TO REMOVE DUPLICATE COMPONENTS
                      NEXT A
                      A1 = A1 + 1
                   NEXT B
                   A1 = B1 + 2
                   B1 = B1 + 1
                NEXT C
------------------------------------------------------------------
```

Figure 13 (Continued)

```
COMPUTE THE RESEPECTIVE RELIABILITIES OF THE TERMS OF THE EXPRESSION
AND THE RELIABILITY OF THE SYSTEM
    IF T9 = 1 THEN THE TERM OF THE EXPRESSION IS PRECEEDED BY A +
            SIGN AND A - SIGN IF T9 = -1

            IF T9 = 1 GO TO BOX
            SIGN$= CHR$(45)
            L5=-1
            GO TO CASE
       BOX  L5=1
            SIGN$=CHR$(43)
       CASE L5$=STR$(L5)
            K7$=STR$(1)

            FOR T = 1 TO X
              L = POS(S$(U),P$(T),1)
              IF L = 0 THEN 1080
              IF Z = 1 THEN 1060
              G3$(T) = K4$(T)
              GO TO UNRELIABILITY COMPUTATIONS
              G3$(T) = K1$(T)
              K7$ = PROD$(K7$,G3$(T),10)
            NEXT T
            N5 = N5 + 1
            K7$(N5) = PROD$(K7$,L5$,10)
            SPCE$ = SPACE$(1)
            REL$=SUM$(REL$,K7$(N5))
            EXPRESSION$ = EXPRESSION$
                        + SPCE$ + SIGN$ + SPCE$ + S$(U)
```

```
SUBROUTINE WHICH WILL SEARCH THE TERM OF EXPRESSION FOR
DUPLICATE VARIABLES
            U = U + 1
            FOR T = 1 TO X
              F = POS(B$(K),P$(T),1)
              F = 0  THEN 760
              T$ = SEG$(B$(K),F,F)
              S$(U) = S$(U) + T$
            NEXT T
```

```
THIS PROCESS WILL CONTINUE UNTIL THE TERM IN WHICH ALL THE PATHS
OR CUTS ARE COMBINED.
```

Figure 13 (Continued)

CHAPTER V

SYSTEM REDUCTION

For large systems, say those which contain more than 10 min cuts or min paths and more than 10 components, it may not be practical to evaluate the system as a whole. Because the number of terms in the reliability expression for min cuts or paths grows at a rate of 2 to the n^{th} power (where n is the number of min cuts or paths) the reliability expression jumps from 1023 terms to 2047 terms as the number of cuts or paths moves from 10 to 11. With the ESPACE method the number of possible points in the event space increases at the same rate as the ADDON method. ESPACE is based on number of components while ADDON depends upon number of paths or cuts. Therefore, a reduction in system size would be an appropriate evaluation technique for large systems. This reduction will involve isolating a section of the system and making its ADDON or ESPACE reliability computation and then inserting it in the system as a component. This is best explained through an illustration. Assume the fault tree given in Figure 14. This system contains 25 min cuts, 3 min paths and 10 components. The following procedure could be applied:

1). Select any gate on the fault tree or dual fault tree that contains a state of the system for which

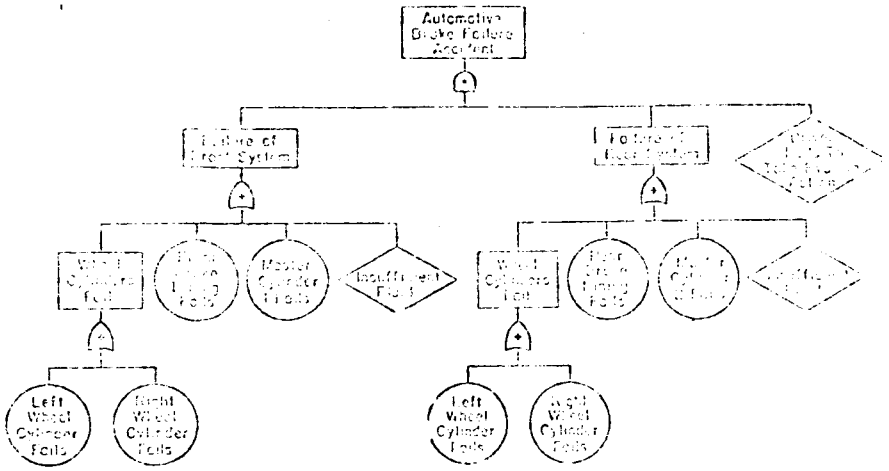


Figure 14

FAULT TREE FOR AUTOMOTIVE BRAKE FAILURE

a reliability calculation is desired. For this example use "failure of rear system", as shown in Figure 15.

2). Treat this gate as a "top event" and calculate its respective reliability or unreliablity using either the ESPACE or ADDON program. Figure 16 is a listing of the statistics about the data. The results of the ADDON run are given in Figure 17.

3). Insert the reliability or unreliability into the diagram and treat this as a new element or component. Figures 18 and 19 are the reduced system fault tree diagram and statistics.

4). Compute the total system reliability. The results of the ADDON program for the reduced system is shown as Figure 20.

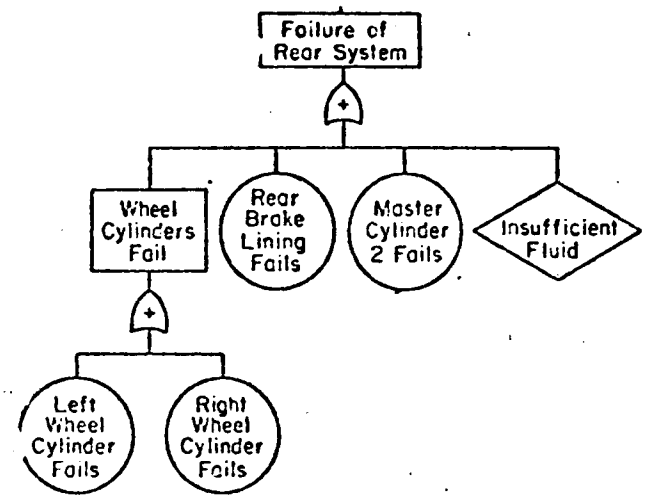


Figure 15

FAULT TREE FOR AUTOMOTIVE REAR BRAKE SYSTEM FAILURE

NUMBER	COMPONENT	COMPONENT IDENTIFIER	PROB. of SUCCESS p	PROB. of FAILURE q
1	Rear Brake Lining Fails	A	.95	.05
2	Master Cylinder 2 Fails	B	.98	.02
3	Insufficient Fluid	C	.97	.03
4	Left Wheel Cylinder Fails	D	.94	.06
5	Right Wheel Cylinder Fails	E	.96	.04

THE CUT SETS ARE

A
B
C
D
E

Figure 16

STATISTICS ABOUT COMPONENTS FOR REAR BRAKE FAULT TREE Figure 15

NUMBER	TERM	UNRELIABILITY OF TERM
######	####	###################
1	A	.05
2	B	.02
3	C	.03
4	D	.06
5	E	.04
6	AB	-.001
7	AC	-.0015
8	AD	-.003
9	AE	-.002
10	BC	-.0006
11	BD	-.0012
12	BE	-.0008
13	CD	-.0018
14	CE	-.0012
15	DE	-.0024
16	ABC	.00003
17	ABD	.00006
18	ABE	.00004
19	ACD	.00009
20	ACE	.00006
21	ADE	.00012
22	BCD	.000036
23	BCE	.000024
24	BDE	.000048
25	CDE	.000072
26	ABCD	-.0000018
27	ABCE	-.0000012
28	ABDE	-.0000024
29	ACDE	-.0000036
30	BCDE	-.00000144
31	ABCDE	.000000072

TOTAL		.185069632

UNRELIABILITY EXPRESSION = + A + B + C + D + E - AB - AC - AD - AE - BC - BD - BE - CD - CE - DE + ABC + ABD + ABE + ACD + ACE + ADE + BCD + BCE + BDE + CDE - ABCD - ABCE - ABDE - ACDE - BCDE + ABCDE

RELIABILITY OF SYSTEM = .814930368

Figure 17

OUTPUT FROM ADDON RUN OF REAR BRAKE (Figure 16) STATISTICS

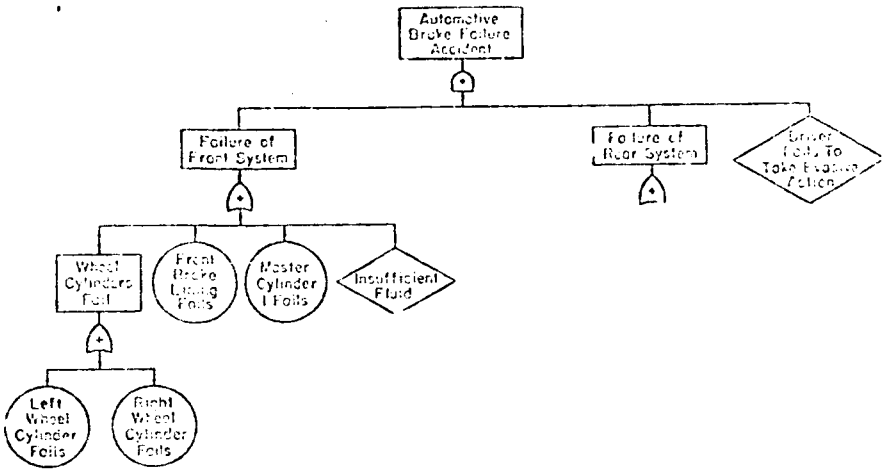


Figure 18

REDUCED FAULT TREE FOR AUTOMOTIVE BRAKE FAILURE

NUMBER	COMPONENT	COMPONENT IDENTIFIER	PROB. of SUCCESS p	PROB. of FAILURE q
1	Failure of Rear System	A	.8149304	.1850696
2	Driver Action Failure	B	.80	.2
3	Front Brake Lining Fails	C	.75	.25
4	Master Cylinder 1 Fails	D	.95	.05
5	Insufficient Fluid	E	.97	.03
6	Left Wheel Cylinder Fails	F	.94	.06
7	Right Wheel Cylinder Fails	G	.96	.04

THE MINIMAL PATHS ARE

A
B
FGCDE

Figure 19

STATISTICS ABOUT THE COMPONENTS OF REDUCED TREE (Figure 18)

NUMBER	TERM	RELIABILITY OF TERM
1	A	.8149304
2	B	.8
3	CDEFG	.5236712
4	AB	-.65194432
5	ACDEFG	-.5082486204
6	BCDEFG	-.49893696
7	ABCDEFG	.4065988964
TOTAL		.986070596

RELIABILITY EXPRESSION = + A + B + CDEFG - AB - ACDEFG - BCDEFG + ABCDEFG

RELIABILITY OF SYSTEM = .986070596

Figure 20

OUTPUT FROM ADDON RUN OF THE REDUCED TREE STATISTICS OF Figure 19

Figures 21 and 22 are the statistics about the data for the unreduced fault tree shown in figure 14 and the output from the ADDON program run of these statistics. Note the variation in the two outputs is only 0.0000000026.

The idea of system reduction might be best explained by the following hypothetical example.

Suppose that 40,000 mile data is accumulated on automobile components and the probability the automobile fails to start at 40,000 miles is to be determined. The fault tree for this situation is drawn by system designers and is shown as Figure 23.

NUMBER	COMPONENT	COMPONENT IDENTIFIER	PROB. of SUCCESS p	PROB. of FAILURE q
1	Driver Action Failure	A	.80	.2
2	Front Brake Lining Fails	B	.75	.25
3	Master Cylinder 1 Fails	C	.95	.05
4	Front Insufficient Fluid	D	.97	.03
5	Rear Brake Lining Fails	E	.95	.05
6	Master Cylinder 2 Fails	F	.98	.02
7	Rear Insufficient Fliud	G	.97	.03
8	Front Left Cylinder Fails	H	.94	.06
9	Front Right Cylinder Fails	I	.96	.04
10	Rear Left Cylinder Fails	J	.94	.06
11	Rear Right Cylinder Fails	K	.96	.04

Figure 21

STATISTICS FOR THE UNREDUCED FAULT TREE OF Figure 14

NUMBER	TERM	RELIABILITY OF TERM	THE MINIMAL PATHS ARE
1	A	.8	A
2	BCDHI	.6236712	HIBCD
3	EFGJK	.814930368	JKEFG
4	ABCDHI	-.49893696	
5	AEFGJK	-.6519442944	
6	BCDEFGHIJK	-.5082486007	
7	ABCDEFGHIJK	.4065988804	
		.9860705933	
TOTAL			

RELIABILITY EXPRESSION = + A + BCDHI + EFGJK - ABCDHI - AEFGJK - BCDEFGHIJK + ABCDEFGHIJK

RELIABILITY OF SYSTEM = .9860705933

Figure 22

OUTPUT FROM ADDON RUN OF THE STATISTICS FOR Figure 21

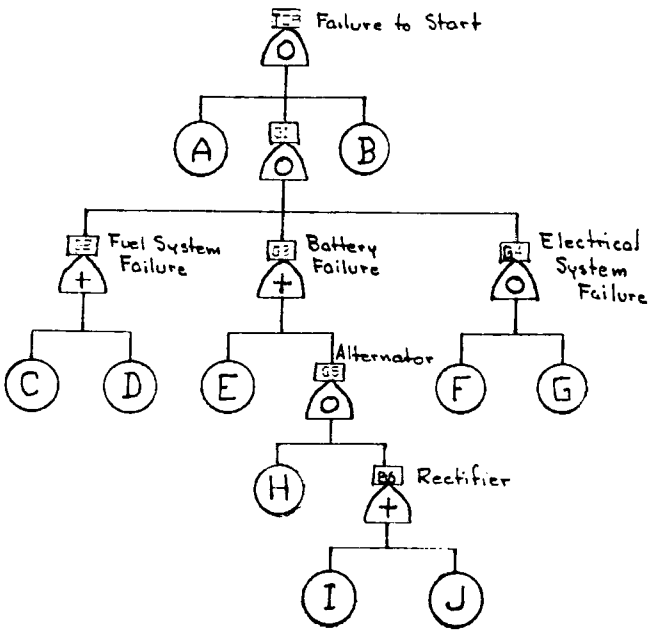


Figure 23

FAULT TREE FOR AUTOMOBILE FAILURE TO START

There are 10 components for which 40,000 mile failure data has been accumulated. The top event is failure to start. Gate 6 (G6) represents the failure of the rectifier in the alternator to operate with components I and J. The rectifier fails if either I or J fails. Gate 5 represents the alternator failure with various components collectively shown as H and alternator failure resulting from rectifier failure as G6. Both H and G6 must fail before the alternator fails. Gate 3 (G3) represents battery failure which occurs if either its various components, shown as E, fail or if an alternator failure shorts out the battery. Gate 2 (G2) represents the fuel system with components C and D being collections of various fuel system components. Component C might represent the various carburator systems considered collectively. Gate 4 (G4) represents the electrical system failure. Both components F and G must fail to cause a G4 failure. At the top is the "and" gate stating that G1 (representing the failure of the fuel system or battery or electrical system) must occur in conjunction with events A and B for the start failure to take place.

This example is hypothetical but generally representative. It is possible that the complete fault tree for the top event would be drawn and one system failure probability calculated. It is more likely that the system designers want to know the probability of

alternator failure at 40,000 milies, of battery failure at 40,000 miles and electrical failure at 40,000 miles. That is, there is a normal desire to determine failure information on system components. This implies a need for system reduction. The cut sets for the complete system of Figure 23 are determined by the min cut algorithm to be :

ABC,ABD,ABE,ABFG,ABHI,ABHJ

Suppose Gate G6 is evaluated and replaced in the fault tree by G6. A reapplication of the min cut algorithm would give cuts:

ABC,ABD,ABE,ABFG,ABH(G6)

Because the algorithm converts ABH(G6) to ABHI and ABHJ. Similarly, the fault tree with Gate 4 represented by component (G4) results in cuts:

ABC,ABD,ABE,ABHI,ABHJ,AB(G4)

and the algorithm converts AB(G4) to ABFG.

If the battery failure probability has been determined and is represented in the fault tree by (G3), the cut sets are:

ABC,ABD,ABFG,AB(G3)

Again the algorithm converts AB(G3) to ABE, ABHI and ABHJ.

The purpose of this example is to show that there is no question about equivalence when system reductions occur. Any gate in the fault tree can be replaced by an

equivalent representative component with a known probability of failure.

This means that either program ESPACE or ADDON can be applied to a gate such as G3 (shown as Figure 24) with 4 components to determine the probability of battery failure at 40,000 miles. This would result in a reduced fault tree (shown as Figure 25) with seven components which can, in turn, be evaluated using ESPACE or ADDON.

Figures 26 and 27 are the data statistics for Gate (G3) and for the reduced fault tree respectively. Note that G3 is represented in the reduced tree data as component Z.

Figures 28 and 29 are the respective program outputs for the data runs with the ADDON program.

Note the reliability of Gate (G3) is found to be .92245 and is used as the component reliability for Z in the reduced system. The resulting total system reliability is found to be 0.9980892851.

Given in Figure 30 are the statistics for the total system without reduction. The ADDON program is used and the result is a reliability of 0.9980892850 as also shown in Figure 30. The difference is only .0000000001.

It should be mentionioned that this system reduction procedure should not be applied to systems with common cause failures. At no time does this analysis or the developed programs consider the common cause problem.

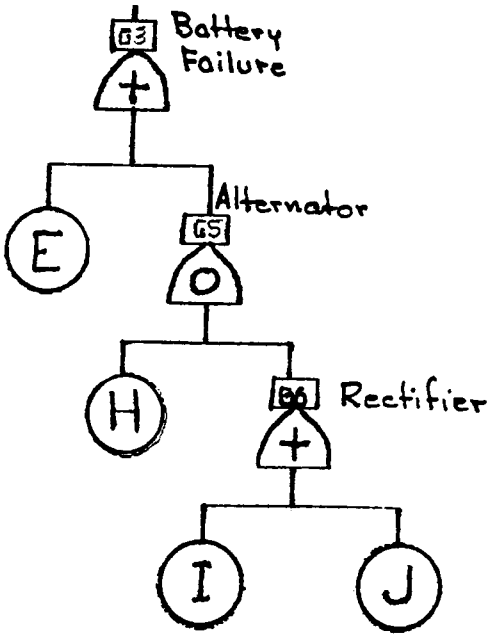


Figure 24

FAULT TREE FOR AUTOMOBILE BATTERY FAILURE

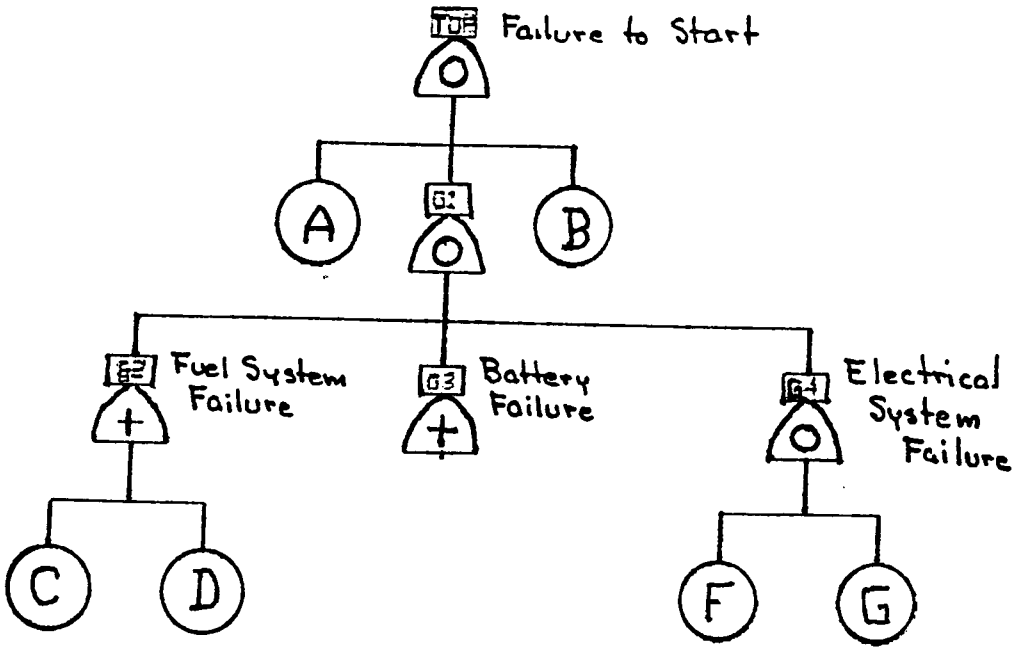


Figure 25

REDUCED FAULT TREE FOR AUTOMOBILE FAILURE TO START

NUMBER	COMPONENT	PROBABILITY OF SUCCESS p	PROBABILITY OF FAILURE q
1	E	.95	.05
2	H	.80	.2
3	I	.95	.05
4	J	.90	.1

THE CUT SETS ARE

E
HI
HJ

Figure 26

STATISTICS ABOUT THE COMPONENTS BATTERY FAILURE IN Figure 24

NUMBER	COMPONENT	PROBABILITY OF SUCCESS p	PROBABILITY OF FAILURE q
1	A	.95	.05
2	B	.90	.1
3	C	.85	.15
4	D	.80	.2
5	F	.90	.1
6	G	.85	.15
7	Z	.92245	.07755

THE MINIMAL CUTS ARE

ABC
ABD
ABFG
ABZ

Figure 27

STATISTICS ABOUT THE COMPONENTS REDUCED TREE OF Figure 25

POSSIBLE FAILURES WITH THOSE RESULTING IN
TOTAL SYSTEM FAILURE DENOTED BY AN *

1 FAILS	2 FAIL	3 FAIL	4 FAIL
E*	EH*	EHI*	EHIJ*
H	EI*	EHJ*	
I	EJ*	EIJ*	
J	HI*	HIJ*	
	HJ*		
	IJ		

THOSE COMBINATIONS WHICH CONTAIN A CUT SET ARE
DENOTED WITH AN * AND HAVE THE FOLLOWING
PROBABILITIES OF OCCURANCE

COMBINATION	PROBABILITY OF OCCURENCE
**********	*************************
E	.0342
EH	.00855
EI	.0018
EJ	.0038
HI	.00855
HJ	.01805
EHI	.00045
EHJ	.00095
EIJ	.0002
HIJ	.00095
EHIJ	.00005
TOTAL	.07755

RELIABILITY OF SYSTEM = .92245

Figure 28

OUTPUT FROM ESPACE RUN OF BATTERY FAILURE STATISTICS IN Figure 26

NUMBER	TERM	UNRELIABILITY OF TERM
######	####	###################
1	ABC	.00075
2	ABD	.001
3	ABFG	.000075
4	ABZ	.00038775
5	ABCD	-.00015
6	ABCFG	-.00001125
7	ABCZ	-.0000581625
8	ABDFG	-.000015
9	ABDZ	-.00007755
10	ABFGZ	-.0000058163
11	ABCDFG	.00000225
12	ABCDZ	.0000116325
13	ABCFGZ	.0000008724
14	ABDFGZ	.0000011633
15	ABCDFGZ	-.0000001745

TOTAL		.0019107149

UNRELIABILITY EXPRESSION = + ABC + ABD + ABFG + ABZ - ABCD - ABCFG - ABCZ - ABDFG - ABDZ - ABFGZ + ABCDFG + ABCDZ + ABCFGZ + ABDFGZ - ABCDFGZ

RELIABILITY OF SYSTEM = .9980892851

Figure 29

OUTPUT FROM ADDON RUN OF REDUCED TREE STATISTICS IN Figure 27

NUMBER	COMPONENT	COMPONENT IDENTIFIER	PROB. of SUCCESS p	PROB. of FAILURE q
1	Driver Action Failure	A	.80	.2
2	Front Brake Lining Fails	B	.75	.25
3	Master Cylinder 1 Fails	C	.95	.05
4	Front Insufficient Fluid	D	.97	.03
5	Rear Brake Lining Fails	E	.95	.05
6	Master Cylinder 2 Fails	F	.98	.02
7	Rear Insufficient Fliud	G	.97	.03
8	Front Left Cylinder Fails	H	.94	.06
9	Front Right Cylinder Fails	I	.96	.04
10	Rear Left Cylinder Fails	J	.94	.06
11	Rear Right Cylinder Fails	K	.96	.04

THE MINIMAL PATHS ARE

A
HIBCD
JKEFG

NUMBER	TERM	RELIABILITY OF TERM
1	A	.8
2	BCDHI	.6236712
3	EFGJK	.814930368
4	ABCDHI	-.49893696
5	AEFGJK	-.6519442944
6	BCDEFGHIJK	-.5082486007
7	ABCDEFGHIJK	.4065988804
TOTAL		.9860705933

RELIABILITY EXPRESSION = + A + BCDHI + EFGJK - ABCDHI - AEFGJK - BCDEFGHIJK + ABCDEFGHIJK

RELIABILITY OF SYSTEM = .9860705933

Figure 30

STATISTICS ABOUT THE COMPONENTS OF NON REDUCED TREE IN Figure 14 AND THE ADDON PROGRAM OUTPUT FOR THIS SYSTEM

CHAPTER VI

CONCLUSIONS

The objective of developing a computerized calculation for the reliability of a complex system has been met with two basic programs. The first, principally involving the manipulation of strings, produces all subsets for a given set of components, notes those which result in system failure and calculates the system reliability. The second, again manipulating string expressions, applies the addition theorem to a known set of either min paths or min cuts.

The problem always begins with a complex system with a given number of components whose reliabilities are known. Also known are the min path and/or min cut sets. A proper choice of programs is then made; that is, if there are few components and many cut sets, it would be ESPACE. If there are many components but either a few paths or a few cuts, it would be ADDON. Then the program would be applied.

The question of dealing with a system having more than 10 components, 10 paths and 10 cuts is dealt with using system reduction. The result is a relatively simple procedure applied to basic ideas widely used and understood in practice. No translation of the system into a network is required.

The attempts during this study to apply the programs to systems with common cause failure were not satisfactory. The methods needed to deal with the "common cause" problem is beyond the scope of this study. The only recommendation would be to develop programs which are small and easily applied for the "common cause" problem.

Expanding ESPACE and ADDON to more than 10 events is not recommended. This could easily be done from say, 10 to 20. But there is an obvious limit which might be 30, or 50, or 100, to an acceptable running time or print time or print list for what is supposed to be an easy-to-use program.

LIST OF REFERENCES

LIST OF REFERENCES

1). P.A. Jensen, M. Bellmore, "An Algorithm to Determine the Reliability of a Complex System", IEEE Trans. on Reliability, vol. R-18, no. 4, 1969 November, pp 169-174.

2). J.R. Batts, "Computer Program for Approximating System Reliability -- Part II", IEEE Trans. on Reliability, vol. 20, no. 2, 1971 May, pp 88-90.

3). L. Fratta, U.G. Montanari, "A Boolean Algebra Method for Computing the Terminal Reliability in a Communication Network", IEEE Trans. on Circuit Theory, vol. CT-20, no. 3, 1973 May, pp 203-211.

4). C. Singh, R. Billington, "A New Method to Determine the Failure Frequency of a Complex System", IEEE Trans. on Reliability, vol. R-23, no. 4, 1974 October, pp 231-234.

5). K.K. Aggarwal, K.B. Misra, J.S. Gupta, "A Fast Algorithm for Reliability Evaluation", IEEE Trans. on Reliability, vol. R-24, no. 1, 1975 April, pp 83-85.

6). S.J. Kamat, M.W. Riley, "Determination of Reliability Using Event-Based Monte Carlo Simulation", IEEE Trans. on Reliability, vol. R-24, no. 1, 1975 April, pp 73-75.

7). P.M. Lin, B.J. Leon, T.C. Huang, "A New Algorithm for Symbolic System Reliability Analysis", IEEE Trans. on Reliability, vol. R-25, no. 1, 1976 april, pp 2-15.

8). H. Kumamoto, K. Tanaka, K. Inoue, "Efficient Evaluation of System Reliability by Monte Carlo Method", IEEE Trans. on Reliability, vol. R-26, no. 5, 1977 December, pp 311-315.

9). A. Satyanarayana, A. Prabhakar, "New Topological Formula and Rapid Algorithm for Reliability Analysis of Complex Networks", IEEE Trans. on Reliability, vol. R-27, no. 2, 1978 June, pp 82-100.

10). C. Singh, "On s-Independence in 'A new Method to Determine the Failure Frequency of A Complex System'", IEEE Trans. on Reliability, vol. R-27, no. 2, 1978 June, pp 147-148.

11). H. Kumamoto, E.J. Henley, "Top-down Algorithm for Obtaining Prime Implicant Sets of Non-Coherent Fault Trees", IEEE Trans. on Reliability, vol. R-27, no. 4 1978 October, pp 242-249.

12). D.M. Ramuson, N.H. Marshall, "FATRAM-A Core Efficient Cut-Set Algorithm", IEEE Trans. on Reliability, vol. R-27, no. 4, 1978 October, pp 250-253.

13). J.A. Abraham, "An Improved Algorithm for Network Reliability", IEEE Trans. on Reliability, vol. R-28, no. 1, 1979 April, pp 58-61.

14). M.O. Locks, "Recursive Disjoint Products, Inclusion-Exclusion, and Min-Cut Approximations", IEEE Trans. on Reliability, vol. R-29, no. 5, 1980 December, pp 368-371.

15). K.K. Lee, "A Compilation Technique for Exact System Reliability", IEEE Trans. on Reliabilty, vol. R-30, no. 3, 1981 August, pp 284-288.

16). C.L. Hwang, F.A. Tillman, M.H. lee, "System-Reliability Evaluation Techniques for Complex/Large Systems -- A Review", IEEE Trans. on Reliability, vol. R-30, no. 5, 1981 December, pp 416-422.

17). W.S. Lee, D.L. Grosh, F.A. Tillman, C.H. Lie, "Fault Tree Analysis, Methods, and Applications -- A Review", IEEE Trans. on Reliability, vol. R-34, no. 3, 1985 August, pp 194-203.

18). K.D. Heidtmann, "Minset Splitting for Improved Reliability Computation", IEEE Trans. on Reliability, vol. R-35, no. 5, 1986 December, pp 563-565.

19). S.P. Jain, K. Gopal, "On Network Augmentation", IEEE Trans. on Reliability, vol. R-35, no. 5, 1986 December, pp 541-543.

20). N. Limnios, R. Ziani, "An Algorithm For Reducing Cut Sets In Fault-Tree Analysis", IEEE Trans. on Reliability, vol. R-35, no. 5, 1986 December, pp 559-562.

21). L.B. Page, J.E. Perry, "An Algorithm for Exact Fault-Tree Probabilities without Cut Sets", IEEE Tran. on Reliability, vol. R-35, no. 5, 1986 December, pp 544-558.

22). W.J. Rueger, "Reliability Analysis of Networks with Capacity-Constraints and Failures at Branches & Nodes", IEEE Trans. on Reliability, vol. R-35, no. 5, 1986 December, pp 523-528.

23). W.G. Schneeweiss, "The Failure Frequency of Systems with Dependent Components", IEEE Trans. on Reliability, vol. R-35, no. 5, 1986 December, pp 512-517.

APPENDIXES

Appendix A

Program ESPACE

PROGRAM ESPACE

```
100     !       THIS PROGRAM IS DESIGNED TO HANDLE RELIABILITY  !
110     !       COMPUTATIONS OF NO MORE THAN TEN COMPONENTS      !
120     !       THE COMPONENTS ARE FEED INTO THE COMPUTER BY     !
130     !       IDENTIFICATIONS A THROUGH J                      !
140     !!
150     !!
160     ! FIRST READ THE NUMBER OF COMPONETS TO EVALUATE        !
170     READ Y
180     !!
190     ! NEXT THE NUMBER OF CUT SETS ARE READ                  !
200     READ X
210     !!
220     !!
230     !   THE ARRAYS MUST BE DIMENSIONED                      !
240     DIM A$(10)
250     DIM R1$(10)
260     DIM R2$(10)
270     DIM B$(5000)
280     DIM B1$(252)
290     DIM B2$(252)
300     DIM B3$(252)
310     DIM B4$(252)
320     DIM B5$(252)
330     DIM B6$(252)
340     DIM B7$(252)
350     DIM B8$(252)
360     DIM B9$(252)
370     DIM B10$(252)
380     DIM R3$(10)
390     DIM RELS(5000)
400     DIM L$(5000)
410     !!
420     !!
430     !!
440     !  A DUMMY VALUE OF ONE IS ASSIGNED TO A STRING         !
450     !  FOR FUTURE COMPUTATIONS OF 1 MINUS THE FAILURE RATE  !
460     U$=STR$(1)
470     !!
480     !!
490     !!
500     !!
510     ! LOAD THE NAMES OF THE COMPONENTS !
520     FOR T = 1 TO Y
530         READ A$(T)
540     NEXT T
550     !!
560     !!
570     !!
580     !!
590     !!
600   ! LOAD THE COMPONENT RELIABILITY'S !
610   FOR T = 1 TO Y
620       READ R1$(T)
630       R2$(T) = DIF$(U$,R1$(T))
640   NEXT T
```

```
650     !!
660     !!
670      !   READ THE CUT SETS   !
680      FOR I = 1 TO X
690         READ O$(I)
700      NEXT I
710   !!
720   !!
730   !!
740   !  COMPUTE THE COMBINATION OF TERMS FOR ESPACE !
750      GO TO 960
760  !!
770  !!
780  !!
790  !!
800  !   SUBROUTINE WHICH WILL DECIDE IF THE GIVEN ESPACE COMBINATION !
810  !   CONTAINS A CUT SET THAT WILL RESULT IN A SYSTEM FAILURE      !
820      S = S + 1
830      FOR I = 1 TO X
840              FOR T = 1 TO Y
850              Z=POS(O$(I),A$(T),1)
860              IF Z = 0 THEN 900
870              Z$ = SEG$(O$(I),Z,Z)
880              M =POS(B$(K),Z$,1)
890              IF M = 0 THEN 930
900      NEXT T
910     L$(S) = CHR$(42)
920     GO TO 950
930     L$(S) = SPACE$(1)
940     NEXT I
950    RETURN
960    PRINT
970    PRINT
980    PRINT "      ","         ","PROBABILITY","PROBABILITY"
990    PRINT "      ","         ","OF SUCCESS ","OF FAILURE "
1000    PRINT "NUMBER","COMPONENT","     p     ","     q     "
1010    PRINT "------","---------","-----------","-----------"
1020    FOR I = 1 TO Y
1030       PRINT "  ";I,"    ";A$(I),"    ";R1$(I),"    ";R2$(I)
1040    NEXT I
1050    PRINT
1060    PRINT
1070    PRINT
1080    PRINT " THE CUT SETS ARE "
1090    PRINT " ---------------- "
1100    PRINT
1110    FOR I = 1 TO X
1120       PRINT " "; O$(I)
1130    NEXT I
1140    PRINT
1150    PRINT
1160    GO TO 1890
1170    PRINT
1180    IF Y = 1 THEN Z = K1
1190    IF Y = 2 THEN Z = K1
```

```
1200    IF Y = 3 THEN Z = K2
1210    IF Y = 4 THEN Z = K2
1220    IF Y = 5 THEN Z = K3
1230    IF Y = 6 THEN Z = K3
1240    IF Y = 7 THEN Z = K4
1250    IF Y = 8 THEN Z = K4
1260    IF Y = 9 THEN Z = K5
1270    IF Y = 10 THEN Z = K5
1280    PRINT " POSSIBLE FAILURES WITH THOSE RESULTING IN "
1290    PRINT " TOTAL SYSTEM FAILURE DENOTED BY AN *      "
1300    PRINT
1310    PRINT
1320    PRINT C1$,C2$,C3$,C4$,C5$
1330    PRINT D1$,D2$,D3$,D4$,D5$
1340    PRINT
1350    FOR I = 1 TO Z
1360        PRINT "    ";B1$(I)," ";B2$(I)," ";B3$(I),B4$(I),B5$(I)
1370    NEXT I
1380    PRINT
1390    PRINT
1400    PRINT
1410    IF Y > 6 THEN 1430
1420    GO TO 1550
1430    PRINT
1440    PRINT
1450    PRINT C6$,C7$,C8$,C9$,C10$
1460    PRINT D6$,D7$,D8$,D9$,D10$
1470    PRINT
1480    FOR I = 1 TO K6
1490        PRINT B6$(I),B7$(I),B8$(I),B9$(I),B10$(I)
1500    NEXT I
1510    PRINT
1520    PRINT
1530    PRINT
1540    PRINT
1550    PRINT
1560    PRINT
1570    PRINT " THOSE COMBINATIONS WHICH CONTAIN A CUT SET ARE "
1580    PRINT " DENOTED WITH AN * AND HAVE THE FOLLOWING       "
1590    PRINT " PROBABILITIES OF OCCURANCE                     "
1600    PRINT
1610    PRINT
1620    PRINT " COMBINATION "," PROBABILITY OF OCCURENCE "
1630    PRINT " *********** "," ********************** "
1640    PRINT
1650    FOR I = 1 TO S
1660        REL$(I) = U$
1670        IF L$(I) = CHR$(42) THEN 1690
1680        GO TO 1780
1690        FOR N = 1 TO Y
1700            Z = POS(B$(I),A$(N),1)
1710            IF Z = 0 THEN 1740
1720            REL$(I) = PROD$(REL$(I),R2$(N),10)
1730            GO TO 1750
1740        REL$(I) = PROD$(REL$(I),R1$(N),10)
```

```
1750        NEXT N
1760        PRINT "   ";B$(I),"         ";REL$(I)
1770    R$=SUM$(R$,REL$(I))
1780    NEXT I
1790    PRINT ,"         ";"----------"
1800    PRINT " TOTAL ","         ";R$
1810    PRINT
1820    PRINT
1830    PRINT "RELIABILITY OF SYSTEM = ";;DIF$(U$,R$)
1840    PRINT
1850    PRINT
1860    GO TO 6250
1870    !!
1880    !!
1890     A1 = 1
1900     C1$="1 FAILS"
1910     D1$="-------"
1920     FOR A = A1 TO Y
1930         K = K + 1
1940         B$(K) = A$(A)
1950         GOSUB 820
1960         B1$(K) = B$(K) + L$(S)
1970     NEXT A
1980     IF Y > 1 THEN 2020
1990     GO TO 1170
2000     !!
2010     !!
2020     B1 = 1
2030     A1 = 2
2040     C2$="2 FAIL"
2050     D2$="------"
2060     FOR B = B1 TO Y-1
2070     FOR A = A1 TO Y
2080         K2 = K2 + 1
2090         K = K + 1
2100         B$(K) = A$(B)+A$(A)
2110         GOSUB 820
2120         B2$(K2) = B$(K) + L$(S)
2130     NEXT A
2140     A1 = A1 + 1
2150     NEXT B
2160     IF Y > 2 THEN 2200
2170     GO TO 1170
2180     !!
2190     !!
2200     C1 = 1
2210     B1 = 2
2220     A1 = 3
2230     C3$="3 FAIL"
2240     D3$="------"
2250     FOR C = C1 TO Y-2
2260     FOR B = B1 TO Y-1
2270     FOR A = A1 TO Y
2280         K3 = K3 + 1
2290         K = K + 1
```

```
2300        B$(K) = A$(C)+A$(B)+A$(A)
2310        GOSUB 820
2320        B3$(K3) = B$(K) + L$(S)
2330    NEXT A
2340    A1 = A1 + 1
2350    NEXT B
2360    A1 = B1 + 2
2370    B1 = B1 + 1
2380    NEXT C
2390    IF Y > 3 THEN 2430
2400    GO TO 1170
2410    !!
2420    !!
2430    D1 = 1
2440    C1 = 2
2450    B1 = 3
2460    A1 = 4
2470    C4$="4 FAIL"
2480    D4$="————"
2490    FOR D = D1 TO Y-3
2500    FOR C = C1 TO Y-2
2510    FOR B = B1 TO Y-1
2520    FOR A = A1 TO Y
2530        K4 = K4 + 1
2540        K = K + 1
2550        B$(K) = A$(D)+A$(C)+A$(B)+A$(A)
2560        GOSUB 820
2570        B4$(K4) = B$(K) + L$(S)
2580    NEXT A
2590    A1 = A1 + 1
2600    NEXT B
2610    A1 = B1 + 2
2620    B1 = B1 + 1
2630    NEXT C
2640    A1 = C1 + 3
2650    B1 = C1 + 2
2660    C1 = C1 + 1
2670    NEXT D
2680    IF Y > 4 THEN 2720
2690    GO TO 1170
2700    !!
2710    !!
2720    E1 = 1
2730    D1 = 2
2740    C1 = 3
2750    B1 = 4
2760    A1 = 5
2770    C5$="5 FAIL"
2780    D5$="————"
2790    FOR E = E1 TO Y-4
2800    FOR D = D1 TO Y-3
2810    FOR C = C1 TO Y-2
2820    FOR B = B1 TO Y-1
2830    FOR A = A1 TO Y
2840        K5 = K5 + 1
```

```
2850        K = K + 1
2860        B$(K) = A$(E)+A$(D)+A$(C)+A$(B)+A$(A)
2870        GOSUB 820
2880        B5$(K5) = B$(K) + L$(S)
2890    NEXT A
2900    A1 = A1 + 1
2910    NEXT B
2920    A1 = B1 + 2
2930    B1 = B1 + 1
2940    NEXT C
2950    A1 = C1 + 3
2960    B1 = C1 + 2
2970    C1 = C1 + 1
2980    NEXT D
2990    A1 = D1 + 4
3000    B1 = D1 + 3
3010    C1 = D1 + 2
3020    D1 = D1 + 1
3030    NEXT E
3040    IF Y > 5 THEN 3080
3050    GO TO 1170
3060    !!
3070    !!
3080    F1 = 1
3090    E1 = 2
3100    D1 = 3
3110    C1 = 4
3120    B1 = 5
3130    A1 = 6
3140    C6$="6 FAIL"
3150    D6$="------"
3160    FOR F = F1 TO Y-5
3170    FOR E = E1 TO Y-4
3180    FOR D = D1 TO Y-3
3190    FOR C = C1 TO Y-2
3200    FOR B = B1 TO Y-1
3210    FOR A = A1 TO Y
3220        K6 = K6 + 1
3230        K = K + 1
3240    B$(K) = A$(F)+A$(E)+A$(D)+A$(C)+A$(B)+A$(A)
3250    GOSUB 820
3260    B6$(K6) = B$(K) + L$(S)
3270    NEXT A
3280    A1 = A1 + 1
3290    NEXT B
3300    A1 = B1 + 2
3310    B1 = B1 + 1
3320    NEXT C
3330    A1 = C1 + 3
3340    B1 = C1 + 2
3350    C1 = C1 + 1
3360    NEXT D
3370    A1 = D1 + 4
3380    B1 = D1 + 3
3390    C1 = D1 + 2
```

```
3400     D1 = D1 + 1
3410     NEXT E
3420     A1 = E1 + 5
3430     B1 = E1 + 4
3440     C1 = E1 + 3
3450     D1 = E1 + 2
3460     E1 = E1 + 1
3470     NEXT F
3480     IF Y > 6 THEN 3520
3490     GO TO 1170
3500     !!
3510     !!
3520     G1 = 1
3530     F1 = 2
3540     E1 = 3
3550     D1 = 4
3560     C1 = 5
3570     B1 = 6
3580     A1 = 7
3590     C7$ =" 7 FAIL "
3600     D7$ ="————————"
3610     FOR G = G1 TO Y-6
3620     FOR F = F1 TO Y-5
3630     FOR E = E1 TO Y-4
3640     FOR D = D1 TO Y-3
3650     FOR C = C1 TO Y-2
3660     FOR B = B1 TO Y-1
3670     FOR A = A1 TO Y
3680         K7 = K7 + 1
3690         K = K + 1
3700     B$(K) = A$(G)+A$(F)+A$(E)+A$(D)+A$(C)+A$(B)+A$(A)
3710     GOSUB 820
3720     B7$(K7) = B$(K) + L$(S)
3730     NEXT A
3740     A1 = A1 + 1
3750     NEXT B
3760     A1 = B1 + 2
3770     B1 = B1 + 1
3780     NEXT C
3790     A1 = C1 + 3
3800     B1 = C1 + 2
3810     C1 = C1 + 1
3820     NEXT D
3830     A1 = D1 + 4
3840     B1 = D1 + 3
3850     C1 = D1 + 2
3860     D1 = D1 + 1
3870     NEXT E
3880     A1 = E1 + 5
3890     B1 = E1 + 4
3900     C1 = E1 + 3
3910     D1 = E1 + 2
3920     E1 = E1 + 1
3930     NEXT F
3940     A1 = F1 + 6
```

```
3950     B1 = F1 + 5
3960     C1 = F1 + 4
3970     D1 = F1 + 3
3980     E1 = F1 + 2
3990     F1 = F1 + 1
4000     NEXT G
4010     IF Y > 7 THEN 4050
4020     GO TO 1170
4030     !!
4040     !!
4050     H1 = 1
4060     G1 = 2
4070     F1 = 3
4080     E1 = 4
4090     D1 = 5
4100     C1 = 6
4110     B1 = 7
4120     A1 = 8
4130     C8$=" 8 FAIL "
4140     D9$="--------"
4150     FOR H = H1 TO Y-7
4160     FOR G = G1 TO Y-6
4170     FOR F = F1 TO Y-5
4180     FOR E = E1 TO Y-4
4190     FOR D = D1 TO Y-3
4200     FOR C = C1 TO Y-2
4210     FOR B = B1 TO Y-1
4220     FOR A = A1 TO Y
4230         K8 = K8 + 1
4240         K = K + 1
4250 B$(K)=A$(H)+A$(G)+A$(F)+A$(E)+A$(D)+A$(C)+A$(B)+A$(A)
4260     GOSUB 820
4270     B8$(K8) = B$(K) + L$(S)
4280     NEXT A
4290     A1 = A1 + 1
4300     NEXT B
4310     A1 = B1 + 2
4320     B1 = B1 + 1
4330     NEXT C
4340     A1 = C1 + 3
4350     B1 = C1 + 2
4360     C1 = C1 + 1
4370     NEXT D
4380     A1 = D1 + 4
4390     B1 = D1 + 3
4400     C1 = D1 + 2
4410     D1 = D1 + 1
4420     NEXT E
4430     A1 = E1 + 5
4440     B1 = E1 + 4
4450     C1 = E1 + 3
4460     D1 = E1 + 2
4470     E1 = E1 + 1
4480     NEXT F
4490     A1 = F1 + 6
```

```
4500      B1 = F1 + 5
4510      C1 = F1 + 4
4520      D1 = F1 + 3
4530      E1 = F1 + 2
4540      F1 = F1 + 1
4550      NEXT G
4560      A1 = G1 + 7
4570      B1 = G1 + 6
4580      C1 = G1 + 5
4590      D1 = G1 + 4
4600      E1 = G1 + 3
4610      F1 = G1 + 2
4620      G1 = G1 + 1
4630      NEXT H
4640      IF Y > 8 THEN 4680
4650      GO TO 1170
4660      !!
4670      !!
4680      I1 = 1
4690      H1 = 2
4700      G1 = 3
4710      F1 = 4
4720      E1 = 5
4730      D1 = 6
4740      C1 = 7
4750      B1 = 8
4760      A1 = 9
4770      C9$="  9 FAIL  "
4780      D9$="----------"
4790      FOR I = I1 TO Y-8
4800      FOR H = H1 TO Y-7
4810      FOR G = G1 TO Y-6
4820      FOR F = F1 TO Y-5
4830      FOR E = E1 TO Y-4
4840      FOR D = D1 TO Y-3
4850      FOR C = C1 TO Y-2
4860      FOR B = B1 TO Y-1
4870      FOR A = A1 TO Y
4880          K = K + 1
4890          K9 = K9 + 1
4900  B$(K)=A$(I)+A$(H)+A$(G)+A$(F)+A$(E)+A$(D)+A$(C)+A$(B)+A$(A)
4910      GOSUB 820
4920      B9$(K9) = B$(K) + L$(S)
4930      NEXT A
4940      A1 = A1 + 1
4950      NEXT B
4960      A1 = B1 + 2
4970      B1 = B1 + 1
4980      NEXT C
4990      A1 = C1 + 3
5000      B1 = C1 + 2
5010      C1 = C1 + 1
5020      NEXT D
5030      A1 = D1 + 4
5040      B1 = D1 + 3
```

```
5050      C1 = D1 + 2
5060      D1 = D1 + 1
5070      NEXT E
5080      A1 = E1 + 5
5090      B1 = E1 + 4
5100      C1 = E1 + 3
5110      D1 = E1 + 2
5120      E1 = E1 + 1
5130      NEXT F
5140      A1 = F1 + 6
5150      B1 = F1 + 5
5160      C1 = F1 + 4
5170      D1 = F1 + 3
5180      E1 = F1 + 2
5190      F1 = F1 + 1
5200      NEXT G
5210      A1 = G1 + 7
5220      B1 = G1 + 6
5230      C1 = G1 + 5
5240      D1 = G1 + 4
5250      E1 = G1 + 3
5260      F1 = G1 + 2
5270      G1 = G1 + 1
5280      NEXT H
5290      A1 = H1 + 8
5300      B1 = H1 + 7
5310      C1 = H1 + 6
5320      D1 = H1 + 5
5330      E1 = H1 + 4
5340      F1 = H1 + 3
5350      G1 = H1 + 2
5360      H1 = H1 + 1
5370      NEXT I
5380      IF Y > 9 THEN 5420
5390      GO TO 1170
5400      !!
5410      !!
5420      J1 = 1
5430      I1 = 2
5440      H1 = 3
5450      G1 = 4
5460      F1 = 5
5470      E1 = 6
5480      D1 = 7
5490      C1 = 8
5500      B1 = 9
5510      A1 = 10
5520      C10$="  10 FAIL  "
5530      D10$="------------"
5540      FOR J = J1 TO Y-9
5550      FOR I = I1 TO Y-8
5560      FOR H = H1 TO Y-7
5570      FOR G = G1 TO Y-6
5580      FOR F = F1 TO Y-5
5590      FOR E = E1 TO Y-4
```

```
5600      FOR D = D1 TO Y-3
5610      FOR C = C1 TO Y-2
5620      FOR B = B1 TO Y-1
5630      FOR A = A1 TO Y
5640          K = K + 1
5650          K10 = K10 + 1
5660 BS(K)= AS(J)+AS(I)+AS(H)+AS(G)+AS(F)+AS(E)+AS(D)+AS(C)+AS(B)+AS(A)
5670      GOSUB 820
5680      B10$(K10) = BS(K) + L$(S)
5690      NEXT A
5700      A1 = A1 + 1
5710      NEXT B
5720      A1 = B1 + 2
5730      B1 = B1 + 1
5740      NEXT C
5750      A1 = C1 + 3
5760      B1 = C1 + 2
5770      C1 = C1 + 1
5780      NEXT D
5790      A1 = D1 + 4
5800      B1 = D1 + 3
5810      C1 = D1 + 2
5820      D1 = D1 + 1
5830      NEXT E
5840      A1 = E1 + 5
5850      B1 = E1 + 4
5860      C1 = E1 + 3
5870      D1 = E1 + 2
5880      E1 = E1 + 1
5890      NEXT F
5900      A1 = F1 + 6
5910      B1 = F1 + 5
5920      C1 = F1 + 4
5930      D1 = F1 + 3
5940      E1 = F1 + 2
5950      F1 = F1 + 1
5960      NEXT G
5970      A1 = G1 + 7
5980      B1 = G1 + 6
5990      C1 = G1 + 5
6000      D1 = G1 + 4
6010      E1 = G1 + 3
6020      F1 = G1 + 2
6030      G1 = G1 + 1
6040      NEXT H
6050      A1 = H1 + 8
6060      B1 = H1 + 7
6070      C1 = H1 + 6
6080      D1 = H1 + 5
6090      E1 = H1 + 4
6100      F1 = H1 + 3
6110      G1 = H1 + 2
6120      H1 = H1 + 1
6130      NEXT I
6140      A1 = I1 + 9
```

```
6150      B1 = I1 + 8
6160      C1 = I1 + 7
6170      D1 = I1 + 6
6180      E1 = I1 + 5
6190      F1 = I1 + 4
6200      G1 = I1 + 3
6210      H1 = I1 + 2
6220      I1 = I1 + 1
6230      NEXT J
6240      GO TO 1170
6250      GO TO 7000
7000      END
```

Appendix B

Program ADDON

PROGRAM ADDON

```
100      !      THIS PROGRAM IS DESIGNED TO HANDLE SYSTEMS      !
110      !      CONTAINING UP TO 26  COMPONETS AND 10 MINIMAL   !
120      !      PATHS OR CUTS                                   !
130      !!
140      !!
150      !      THE FIRST STEP IN EVALUATION IS TO IDENTIFY     !
160      !      WHETHER THESE  ARE THESE PATHS OR CUTS.         !
170      !      IF PATHS THEN Z = 1 AND Z = 10 FOR CUTS         !
180      READ Z
190      !!
200      !!
210      !      THE SECOND STEP IS  TO INPUT THE                !
220      !      NUMBER OF PATHS OR CUTS VARIABLE Y              !
230      READ Y
240      !!
250      !!
260      !      THE THIRD STEP IS TO INPUT THE NUMBER           !
270      !      OF COMPONENTS VARIABLE X                        !
280      READ X
290 !!
300 !!
310      DIM A$(1000)
320      DIM Q$(1000)
330      DIM P$(1000)
340      DIM K$(1000)
350      DIM S$(1000)
360      DIM K1$(1000)
370      DIM K4$(1000)
380      DIM G3$(1000)
390      DIM B$(1000)
400      DIM K7$(1000)
410 !!
420 !!
430      W =INT(X/9)
440      V = W + 1
450      U$ = STR$(1)
460 !!
470 !!
480 !!
490      ! LOAD THE COMPONENTS !
500      FOR T = 1 TO X
510          READ P$(T)
520      NEXT T
530 !!
540 !!
550 !!
560      ! LOAD THE PATHS !
570      FOR T = 1 TO Y
580          READ A$(T)
590      NEXT T
600 !!
610 !!
620 !!
630 !!
640      GOSUB 840
```

```
650 !!
660     GO TO 1670
670 !!
680 !!
690 !    SEARCH THE STRING FOR DUPLICATE VARIABLES    !
700      U = U + 1
710      FOR T = 1 TO X
720           F = POS(BS(K),PS(T),1)
730           IF F = 0  THEN 760
740           TS = SEGS(BS(K),F,F)
750           SS(U) = SS(U) + TS
760      NEXT T
770 !!
780      GOSUB 920
790      RETURN
800 !!
810 !!
820 !!
830 !!
840     ! LOAD THE COMPONENT RELIABILITY'S !
850     FOR T = 1 TO X
860         READ K1S(T)
870         K4S(T) = DIFS(US,K1S(T))
880     NEXT T
890     RETURN
900 !!
910 !!
920     IF T9 = 1 THEN 960
930     SIGNS= CHRS(45)
940     L5=-1
950     GO TO 980
960     L5=1
970     SIGNS=CHRS(43)
980     L5S=STRS(L5)
990     K7S=STRS(1)
1000    FOR T = 1 TO X
1010        L = POS(SS(U),PS(T),1)
1020        IF L = 0 THEN 1080
1030        IF Z = 1 THEN 1060
1040        G3S(T) = K4S(T)
1050        GO TO 1070
1060        G3S(T) = K1S(T)
1070        K7S = PRODS(K7S,G3S(T),10)
1080    NEXT T
1090    N5 = N5 + 1
1100    K7S(N5) = PRODS(K7S,L5S,10)
1110    SPCES = SPACES(1)
1120    RELS=SUMS(RELS,K7S(N5))
1130    EXPRESSIONS = EXPRESSIONS + SPCES + SIGNS + SPCES + SS(U)
1140    RETURN
1150 !!
1160 !!
1170 !!
1180 !!
1190 !!
```

```
1200    PRINT "      ","          ","PROBABILITY","PROBABILITY"
1210    PRINT "      ","          ","OF SUCCESS ","OF FAILURE "
1220    PRINT "NUMBER","COMPONENT","     p     ","     q     "
1230    PRINT "------","----------","-----------","-----------"
1240    FOR I = 1 TO X
1250       PRINT I,"    ";P$(I),"    ";K1$(I),"    ";K4$(I)
1260    NEXT I
1270    PRINT
1280    PRINT
1290    IF Z = 10 THEN 1320
1300    PRINT " THE MINIMAL PATHS ARE "
1310    GO TO 1330
1320    PRINT " THE MINIMAL CUTS ARE "
1330    PRINT " ---------------------"
1340    PRINT
1350    FOR I = 1 TO Y
1360       PRINT " ";A$(I)
1370    NEXT I
1380    PRINT
1390    PRINT
1400    IF Z = 1 THEN 1430
1410    PRINT " NUMBER","TERM","UNRELIABILITY OF TERM"
1420    GO TO 1440
1430    PRINT " NUMBER","TERM"," RELIABILITY OF TERM"
1440    PRINT " ######","####","####################"
1450    PRINT
1460    L8 = (2**Y) - 1
1470    FOR I = 1 TO L8
1480       PRINT I,S$(I),K7$(I)
1490    NEXT I
1500    PRINT ,,"----------"
1510    PRINT " TOTAL ",,REL$
1520    PRINT
1530    PRINT
1540    IF Z = 1 THEN 1570
1550    PRINT "UNRELIABILITY EXPRESSION = ",EXPRESSION$
1560    GO TO 1580
1570    PRINT "RELIABILITY EXPRESSION = ",EXPRESSION$
1580    PRINT
1590    PRINT
1600    IF Z = 1 THEN 1640
1610    REEL$ = DIF$(U$,REL$)
1620    PRINT  "RELIABILITY OF SYSTEM = ",REEL$
1630    GO TO 1650
1640    PRINT  "RELIABILITY OF SYSTEM = ",REL$
1650    PRINT
1660    GO TO 5740
1670    !!
1680    !!
1690     A1 = 1
1700     FOR A = A1 TO Y
1710        T9 = 1
1720        K = K + 1
1730        B$(K) = A$(A)
1740        GOSUB 700
```

```
1750    NEXT A
1760    IF Y > 1 THEN 1800
1770    GO TO 1200
1780    !!
1790    !!
1800    B1 = 1
1810    A1 = 2
1820    FOR B = B1 TO Y-1
1830    FOR A = A1 TO Y
1840        T9 = -1
1850        K = K + 1
1860        B$(K) = A$(B)+A$(A)
1870        GOSUB 700
1880    NEXT A
1890    A1 = A1 + 1
1900    NEXT B
1910    !!
1920    !!
1930    C1 = 1
1940    B1 = 2
1950    A1 = 3
1960    FOR C = C1 TO Y-2
1970    FOR B = B1 TO Y-1
1980    FOR A = A1 TO Y
1990        T9 = 1
2000        K = K + 1
2010        B$(K) = A$(C)+A$(B)+A$(A)
2020        GOSUB 700
2030    NEXT A
2040    A1 = A1 + 1
2050    NEXT B
2060    A1 = B1 + 2
2070    B1 = B1 + 1
2080    NEXT C
2090    IF Y > 3 THEN 2130
2100    GO TO 1200
2110    !!
2120    !!
2130    D1 = 1
2140    C1 = 2
2150    B1 = 3
2160    A1 = 4
2170    FOR D = D1 TO Y-3
2180    FOR C = C1 TO Y-2
2190    FOR B = B1 TO Y-1
2200    FOR A = A1 TO Y
2210        T9 = -1
2220        K = K + 1
2230        B$(K) = A$(D)+A$(C)+A$(B)+A$(A)
2240        GOSUB 700
2250    NEXT A
2260    A1 = A1 + 1
2270    NEXT B
2280    A1 = B1 + 2
2290    B1 = B1 + 1
```

```
2300      NEXT C
2310      A1 = C1 + 3
2320      B1 = C1 + 2
2330      C1 = C1 + 1
2340      NEXT D
2350      IF Y > 4 THEN 2390
2360      GO TO 1200
2370      !!
2380      !!
2390      E1 = 1
2400      D1 = 2
2410      C1 = 3
2420      B1 = 4
2430      A1 = 5
2440      FOR E = E1 TO Y-4
2450      FOR D = D1 TO Y-3
2460      FOR C = C1 TO Y-2
2470      FOR B = B1 TO Y-1
2480      FOR A = A1 TO Y
2490          T9 = 1
2500          K = K + 1
2510          B$(K) = A$(E)+A$(D)+A$(C)+A$(B)+A$(A)
2520          GOSUB 700
2530      NEXT A
2540      A1 = A1 + 1
2550      NEXT B
2560      A1 = B1 + 2
2570      B1 = B1 + 1
2580      NEXT C
2590      A1 = C1 + 3
2600      B1 = C1 + 2
2610      C1 = C1 + 1
2620      NEXT D
2630      A1 = D1 + 4
2640      B1 = D1 + 3
2650      C1 = D1 + 2
2660      D1 = D1 + 1
2670      NEXT E
2680      IF Y > 5 THEN 2720
2690      GO TO 1200
2700      !!
2710      !!
2720      F1 = 1
2730      E1 = 2
2740      D1 = 3
2750      C1 = 4
2760      B1 = 5
2770      A1 = 6
2780      FOR F = F1 TO Y-5
2790      FOR E = E1 TO Y-4
2800      FOR D = D1 TO Y-3
2810      FOR C = C1 TO Y-2
2820      FOR B = B1 TO Y-1
2830      FOR A = A1 TO Y
2840          T9 = -1
```

```
2850        K = K + 1
2860     BS(K) = AS(F)+AS(E)+AS(D)+AS(C)+AS(B)+AS(A)
2870     GOSUB 700
2880     NEXT A
2890     A1 = A1 + 1
2900     NEXT B
2910     A1 = B1 + 2
2920     B1 = B1 + 1
2930     NEXT C
2940     A1 = C1 + 3
2950     B1 = C1 + 2
2960     C1 = C1 + 1
2970     NEXT D
2980     A1 = D1 + 4
2990     B1 = D1 + 3
3000     C1 = D1 + 2
3010     D1 = D1 + 1
3020     NEXT E
3030     A1 = E1 + 5
3040     B1 = E1 + 4
3050     C1 = E1 + 3
3060     D1 = E1 + 2
3070     E1 = E1 + 1
3080     NEXT F
3090     IF Y > 6 THEN 3130
3100     GO TO 1200
3110     !!
3120     !!
3130     G1 = 1
3140     F1 = 2
3150     E1 = 3
3160     D1 = 4
3170     C1 = 5
3180     B1 = 6
3190     A1 = 7
3200     FOR G = G1 TO Y-6
3210     FOR F = F1 TO Y-5
3220     FOR E = E1 TO Y-4
3230     FOR D = D1 TO Y-3
3240     FOR C = C1 TO Y-2
3250     FOR B = B1 TO Y-1
3260     FOR A = A1 TO Y
3270        T9 = 1
3280        K = K + 1
3290     BS(K) = AS(G)+AS(F)+AS(E)+AS(D)+AS(C)+AS(B)+AS(A)
3300     GOSUB 700
3310     NEXT A
3320     A1 = A1 + 1
3330     NEXT B
3340     A1 = B1 + 2
3350     B1 = B1 + 1
3360     NEXT C
3370     A1 = C1 + 3
3380     B1 = C1 + 2
3390     C1 = C1 + 1
```

```
3400      NEXT D
3410      A1 = D1 + 4
3420      B1 = D1 + 3
3430      C1 = D1 + 2
3440      D1 = D1 + 1
3450      NEXT E
3460      A1 = E1 + 5
3470      B1 = E1 + 4
3480      C1 = E1 + 3
3490      D1 = E1 + 2
3500      E1 = E1 + 1
3510      NEXT F
3520      A1 = F1 + 6
3530      B1 = F1 + 5
3540      C1 = F1 + 4
3550      D1 = F1 + 3
3560      E1 = F1 + 2
3570      F1 = F1 + 1
3580      NEXT G
3590      IF Y > 7 THEN 3630
3600      GO TO 1200
3610      !!
3620      !!
3630      H1 = 1
3640      G1 = 2
3650      F1 = 3
3660      E1 = 4
3670      D1 = 5
3680      C1 = 6
3690      B1 = 7
3700      A1 = 8
3710      FOR H = H1 TO Y-7
3720      FOR G = G1 TO Y-6
3730      FOR F = F1 TO Y-5
3740      FOR E = E1 TO Y-4
3750      FOR D = D1 TO Y-3
3760      FOR C = C1 TO Y-2
3770      FOR B = B1 TO Y-1
3780      FOR A = A1 TO Y
3790          T9 = -1
3800          K = K + 1
3810      BS(K)=AS(H)+AS(G)+AS(F)+AS(E)+AS(D)+AS(C)+AS(B)+AS(A)
3820      GOSUB 700
3830      NEXT A
3840      A1 = A1 + 1
3850      NEXT B
3860      A1 = B1 + 2
3870      B1 = B1 + 1
3880      NEXT C
3890      A1 = C1 + 3
3900      B1 = C1 + 2
3910      C1 = C1 + 1
3920      NEXT D
3930      A1 = D1 + 4
3940      B1 = D1 + 3
```

```
3950      C1 = D1 + 2
3960      D1 = D1 + 1
3970      NEXT E
3980      A1 = E1 + 5
3990      B1 = E1 + 4
4000      C1 = E1 + 3
4010      D1 = E1 + 2
4020      E1 = E1 + 1
4030      NEXT F
4040      A1 = F1 + 6
4050      B1 = F1 + 5
4060      C1 = F1 + 4
4070      D1 = F1 + 3
4080      E1 = F1 + 2
4090      F1 = F1 + 1
4100      NEXT G
4110      A1 = G1 + 7
4120      B1 = G1 + 6
4130      C1 = G1 + 5
4140      D1 = G1 + 4
4150      E1 = G1 + 3
4160      F1 = G1 + 2
4170      G1 = G1 + 1
4180      NEXT H
4190      IF Y > 8 THEN 4230
4200      GO TO 1200
4210      !!
4220      !!
4230      I1 = 1
4240      H1 = 2
4250      G1 = 3
4260      F1 = 4
4270      E1 = 5
4280      D1 = 6
4290      C1 = 7
4300      B1 = 8
4310      A1 = 9
4320      FOR I = I1 TO Y-8
4330      FOR H = H1 TO Y-7
4340      FOR G = G1 TO Y-6
4350      FOR F = F1 TO Y-5
4360      FOR E = E1 TO Y-4
4370      FOR D = D1 TO Y-3
4380      FOR C = C1 TO Y-2
4390      FOR B = B1 TO Y-1
4400      FOR A = A1 TO Y
4410          T9 = 1
4420          K = K + 1
4430 B$(K)=A$(I)+A$(H)+A$(G)+A$(F)+A$(E)+A$(D)+A$(C)+A$(B)+A$(A)
4440      GOSUB 700
4450      NEXT A
4460      A1 = A1 + 1
4470      NEXT B
4480      A1 = B1 + 2
4490      B1 = B1 + 1
```

```
4500     NEXT C
4510     A1 = C1 + 3
4520     B1 = C1 + 2
4530     C1 = C1 + 1
4540     NEXT D
4550     A1 = D1 + 4
4560     B1 = D1 + 3
4570     C1 = D1 + 2
4580     D1 = D1 + 1
4590     NEXT E
4600     A1 = E1 + 5
4610     B1 = E1 + 4
4620     C1 = E1 + 3
4630     D1 = E1 + 2
4640     E1 = E1 + 1
4650     NEXT F
4660     A1 = F1 + 6
4670     B1 = F1 + 5
4680     C1 = F1 + 4
4690     D1 = F1 + 3
4700     E1 = F1 + 2
4710     F1 = F1 + 1
4720     NEXT G
4730     A1 = G1 + 7
4740     B1 = G1 + 6
4750     C1 = G1 + 5
4760     D1 = G1 + 4
4770     E1 = G1 + 3
4780     F1 = G1 + 2
4790     G1 = G1 + 1
4800     NEXT H
4810     A1 = H1 + 8
4820     B1 = H1 + 7
4830     C1 = H1 + 6
4840     D1 = H1 + 5
4850     E1 = H1 + 4
4860     F1 = H1 + 3
4870     G1 = H1 + 2
4880     H1 = H1 + 1
4890     NEXT I
4900     IF Y > 9 THEN 4940
4910     GO TO 1200
4920     !!
4930     !!
4940     J1 = 1
4950     I1 = 2
4960     H1 = 3
4970     G1 = 4
4980     F1 = 5
4990     E1 = 6
5000     D1 = 7
5010     C1 = 8
5020     B1 = 9
5030     A1 = 10
5040     FOR J = J1 TO Y-9
```

```
5050      FOR I = I1 TO Y-8
5060      FOR H = H1 TO Y-7
5070      FOR G = G1 TO Y-6
5080      FOR F = F1 TO Y-5
5090      FOR E = E1 TO Y-4
5100      FOR D = D1 TO Y-3
5110      FOR C = C1 TO Y-2
5120      FOR B = B1 TO Y-1
5130      FOR A = A1 TO Y
5140          T9 = -1
5150          K = K + 1
5160 BS(K) = AS(J)+AS(I)+AS(H)+AS(G)+AS(F)+AS(E)+AS(D)+AS(C)+AS(B)+AS(A)
5170          GOSUB 700
5180      NEXT A
5190      A1 = A1 + 1
5200      NEXT B
5210      A1 = B1 + 2
5220      B1 = B1 + 1
5230      NEXT C
5240      A1 = C1 + 3
5250      B1 = C1 + 2
5260      C1 = C1 + 1
5270      NEXT D
5280      A1 = D1 + 4
5290      B1 = D1 + 3
5300      C1 = D1 + 2
5310      D1 = D1 + 1
5320      NEXT E
5330      A1 = E1 + 5
5340      B1 = E1 + 4
5350      C1 = E1 + 3
5360      D1 = E1 + 2
5370      E1 = E1 + 1
5380      NEXT F
5390      A1 = F1 + 6
5400      B1 = F1 + 5
5410      C1 = F1 + 4
5420      D1 = F1 + 3
5430      E1 = F1 + 2
5440      F1 = F1 + 1
5450      NEXT G
5460      A1 = G1 + 7
5470      B1 = G1 + 6
5480      C1 = G1 + 5
5490      D1 = G1 + 4
5500      E1 = G1 + 3
5510      F1 = G1 + 2
5520      G1 = G1 + 1
5530      NEXT H
5540      A1 = H1 + 8
5550      B1 = H1 + 7
5560      C1 = H1 + 6
5570      D1 = H1 + 5
5580      E1 = H1 + 4
5590      F1 = H1 + 3
```

```
5600      G1 = H1 + 2
5610      H1 = H1 + 1
5620      NEXT I
5630      A1 = I1 + 9
5640      B1 = I1 + 8
5650      C1 = I1 + 7
5660      D1 = I1 + 6
5670      E1 = I1 + 5
5680      F1 = I1 + 4
5690      G1 = I1 + 3
5700      H1 = I1 + 2
5710      I1 = I1 + 1
5720      NEXT J
5730      GO TO 1200
5740      GO TO 7000
7000      END
```

VITA

Mary Karen Lyle was born in Johnson City, Tennessee on June 8, 1964. She attended high school at University High in Johnson City and after graduating in June of 1981, she entered college at East Tennessee State University. In January of 1983 she transfered to the University of Tennessee at Knoxville and received her Bachelors degree of Industrial Engineering in June of 1985. Karen started the master's program immediately after graduation and will receive her masters degree in December of 1987.