ROUGH DRAF

UCID-

Initial Guidance on Digraph-Matrix Analysis for Systems Interaction Studies at Selected LWR's

H. P. Alesso
Lawrence Livermore National Laboratory

I. J. Sacks
Analytic Information Processing, Inc.

C. F. Smith
Science Applications, Inc.

June 14, 1982

Lawrence Livermore Laboratory

Work performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore Laboratory under Contract W-7405-Eng-48.

Table of Contents

	page
1.0 Introduction	1
2.0 Initial Guidance on Digraph-Matrix Analysis for Systems Interactions	6
2.1 Step 1: Selection of Combinations of Systems for Detailed Evaluation	6
2.1.1 Substep 1a: Study Plant Design and Operating History	6
2.1.2 Substep 1b: Development of a List of Accident Initiators	7
2.1.3 Substep 1c: Development of Function Event Trees	7
2.1.4 Substep 1d: Assignment of Front-Line Systems to Functional Event Trees	8
2.1.5 Substep 1e: Results of Event Tree Analysis	9
2.1.6 Substep 1f: Assignment of Support Systems to Front-Line Systems	9
2.2 Step 2: Constructing a Global Digraph Model for each Accident Sequence	10
2.2.1 Substep 2a: Constructing a Success-Oriented Digraph Model for each Accident Sequence	16
2.2.2 Substep 2b: Forming the Dual Digraph Model	18
2.2.3 Substep 2c: Forming Adjacency Matrix Input	21
2.2.4 Substep 2d: Processing Adjacency Matrix	24
2.2.5 Substep 2e: Consistency and Completeness Check	28
2.2.6 Substep 2f: Finding Singleton/Doubleton Minimum Cut-Sets	29
2.2.7 Substep 2g: Constructing Unit Models	29
2.2.8 Substep 2h: Expand Adjacency Matrices	31
2.2.9 Substep 2i: Repeat Reachability Processing	31
2.2.10 Substep 2j: Expand Model	37

Table of Contents (continued) page

2.3 Step 3: Partitioning the Global Digraph Model into Independent Subdigraphs 37

2.3.1 Substep 3a: Partitioning the Global Digraph Model 38

2.3.2 Substep 3b: Reachability Processing of Subdigraphs 41

2.3.3 Substep 3c: Interior Component Elimination 42

2.3.4 Substep 3d: Construction of the Reduced Adjacency Matrix 42

2.3.5 Substep 3e: Reachability Processing on Reduced Adjacency Matrix 42

2.3.6 Substep 3f: Addition of Eliminated Components 42

2.4 Step 4: Evaluate Singletons and Doubletons using Probability and Display Answers 44

3.0 Conclusion 45

References 46

Appendix 1 51

List of Illustrations

	page
1. Overview of Construction of Global Digraph	11
2. Overview of Step 2	14
3. Simplified Cooling System Example	17
4. Logic Symbol Conventions for Success-Oriented Digraph	18
5. Success-Oriented Digraph of Simplified Cooling System Example	19
6. Dual Digraph of Simplified Cooling System Example	20
7. Adjacency Matrix Conventions	21
8. Adjacency Input for Dual Digraph	22
9. Variable List for Dual Digraph	23
10. Example Network	24
11. Example Network with AND Gate	24
12. Example of Network with Multiple Paths	25
13. Example of a System with Singleton	26
14. Adjacency Matrix for Dual Digraph	27
15. Singleton Matrix for Dual Digraph	28
16. Example of a System with a Doubleton	29
17. Singletons and Doubletons for Simplified Cooling System Example	30
18. Pump Unit Model	30
19. Unit Model with Redundant Supplies for Pump	30
20. Example Unit Models (Duals)	32
21. Expanded Dual System Digraph	33
22. Input System for Expanded System	34
23. Singletons and Doubletons of Example	36
24. Partitioning the Global Digraph	39
25. Partitioning Process	40

List of Illustrations

		page
1.	Overview of Construction of Global Digraph	11
2.	Overview of Step 2	14
3.	Simplified Cooling System Example	17
4.	Logic Symbol Conventions for Success-Oriented Digraph	18
5.	Success-Oriented Digraph of Simplified Cooling System Example	19
6.	Dual Digraph of Simplified Cooling System Example	20
7.	Adjacency Matrix Conventions	21
8.	Adjacency Input for Dual Digraph	22
9.	Variable List for Dual Digraph	23
10.	Example Network	24
11.	Example Network with AND Gate	24
12.	Example of Network with Multiple Paths	25
13.	Example of a System with Singleton	26
14.	Adjacency Matrix for Dual Digraph	27
15.	Singleton Matrix for Dual Digraph	28
16.	Example of a System with a Doubleton	29
17.	Singletons and Doubletons for Simplified Cooling System Example	30
18.	Pump Unit Model	30
19.	Unit Model with Redundant Supplies for Pump	30
20.	Example Unit Models (Duals)	32
21.	Expanded Dual System Digraph	33
22.	Input System for Expanded System	34
23.	Singletons and Doubletons of Example	36
24.	Partitioning the Global Digraph	39
25.	Partitioning Process	40

1.0 INTRODUCTION

A crucial assumption in performing a risk analysis of nuclear power plants is that basic events, the last events in a fault tree, are independent. Recent complex accidents, such as Three Mile Island-2 [1], Brown's Ferry-3 [2-6] and Crystal River-3 [7], however, have demonstrated that this assumption can, in some cases, be seriously in error. These events occurred primarily as a result of dependent (common cause/mode) failures.

Dependencies, such as, shared environmental conditions, shared support systems and dynamic human error are now being called systems interactions. The NRC is presently developing guidance for the identification and evaluation of systems interactions. Several national laboratories and utilities have contributed preliminary procedures toward this effort. As a result two points of view of the systems interaction problem have been identified.

One point of view is that systems interactions can be adequately analyzed by enhancing existing Probabilistic Risk Assessement (PRA). This could be accomplished by expanding the scope and boundary conditions of fault tree analysis and putting additional emphasis on dependency analysis through such techniques as generic analysis [8], minimum cut-set common cause/mode analysis [8-13] or digraph-fault tree analysis [9, 14]. NRC's initial guidance for this point of view has already begun [15]. The advantage of pursuing this point of view is based on the fact that there are numerous experienced analysts and proven computer codes available to industry. However, this point of view has been reproached primarily on the basis of completeness. Because of computer limitations in finding the complete set of minimum cut-sets there are two areas where traditional PRA may be considered incomplete. First, the individual front-line systems composing an accident sequence have their fault trees constructed separately. As a result, some involved interrelationships and feedback conditions may not be completely represented when the individual systems are ANDed together. Secondly, the fault trees limit themselves by including neither the details of the front-line system's support systems, nor the interrelationship between support systems.

Fault tree analysis requires an analyst to flawlessly use deductive reasoning as he works from the top event down to the basic events. There is no unique mathematical algorithm that will result in a uniquely correct fault tree. Indeed, there is often significant variation between analysts over the same system. In addition, in order to meet computer limitations a specific failure mode in the form of the top event is chosen by the fault tree analyst. This choice precludes consideration of other types of failures or partial system degradation which may be of interest in studying systems interactions.

In attempting to overcome the question of completeness, an enhanced PRA produces a second serious problem; that of saturating the analyst. As the analyst is faced with modeling ever larger and more complex systems and groups of systems, he faces a severe burden on his ability to correctly model the combinatorical interrelationships.

A second point of view of the systems interaction problem is that a new methodology called Digraph-Matrix Analysis utilizing matrix representation of logic diagrams may offer a more complete and possibly more efficient analysis in certain areas. A review of the fundamental mathematical aspects of fault-oriented and success-oriented risk analyses (including Digraph-Matrix Analysis) was presented in [9], which offered insight into the trade-off advantages and disadvantages of each. In particular, the disadvantages of the Digraph Matrix Analysis include: (1) few trained analysts and few available computer codes that can be used to develop and subsequently apply the analysis, and (2) for certain types of logic diagrams the analyst's attempt to be more complete may lead to more severe computer limitations than for fault tree analysis. On the other hand, Digraph Matrix Analysis has the advantages of being more complete than fault-oriented analysis because: (1) it analyzes each accident sequence as a single model including its support systems and, therefore, any partitioning done on the basis of independent subgraphs is more rigorous than the usual fault tree analysis which partitions accident sequences in terms of arbitrarily defined systems, and (2) it utilizes success-oriented modeling which for large complex systems can be more easily accomplished by the analyst than fault-oriented modeling.

This report will present proposed initial guidance on Digraph Matrix Analysis for application to systems interactions. It is expected that peer review and subsequent example applications will provide feedback toward a final guidance in the future.

The proposed initial Digraph-Matrix Analysis guidance is presented in Section 2. It consists of a four step process (see Table 1), and exludes shared environmental conditions and most dynamic human error from consideration. The shared environmental conditions are to be evaluated by an independent walk-through evaluation by a multidisciplinary team of experts [16].

The first step of the Digraph-Matrix Analysis (DMA) guidance is similar to the start of a traditional PRA [8, 15, 17]. The first step consists of studying plant design and continues with the development of functional (relating to the four vital safety functions) and systemic event trees to find the accident sequences. Then instead of conducting an independent fault tree analysis on each front line system in the accident sequences, the subsequent Steps 2 and 3 broaden the scope of the analysis, by considering the entire accident sequence along with all the non-safety grade support systems as one model. In order to perform computer analysis of such an expanded problem only singleton and doubleton cut-sets (common modes/causes) are found. In an effort to solve for singletons and doubletons of very large models, Digraph-Matrix Analysis [19-23, 28] and such computer codes as CLAMOR or SQUEAK can be utilized which find the path-sets of the model in prime implicant form. Step 2 consists of constructing a global logic diagram model of the successful operation of each accident sequence which consists of a group of systems along with their non-safety grade support systems. The dual model is then formed by changing the original AND gates to OR gates and vice versa. (See Appendix A and [26, 27] for a further explanation of the use of the dual model.)

Table 1. Overview of Digraph Matrix Analysis

Step 1:	Select combinations of systems for detailed evaluation. (Similar to PRA event tree analysis to find accident sequences).
Step 2:	Construct a global digraph model for each accident sequence.
Step 3:	Partition digraph models into independent subdigraphs and find singleton and doubleton minimum cut-sets of plant function.
Step 4:	Evaluate singletons and doubletons on basis of probability and display answers.

Step 3 is a computer efficiency effort. It tries to attack the large computer problem by partitioning each accident sequence model into independent subdigraphs. A rigorous partitioning of the accident sequence model will improve the completeness of the results while staying within computer limitations.

The fourth and final step is the evaluation of the singleton and doubleton cut-sets on the basis of probability, in order to assess their relative importance. In addition, the doubletons found in the DMA effort will provide a starting point for the walk-through evaluation by a multidisciplinary team of experts who will evaluate shared environmental conditions.

The final results of a Digraph-Matrix Analysis, it should be noted, will not be an entire risk assessment of a plant. It will only find singleton and doubleton minimum cut-sets which are essentially the dependent (common cause/mode) failures of systems interactions. DMA, therefore, must be considered either as a "preprocessor" to a traditional risk assessment or as a stepping stone to a new analysis that is as complete as DMA, but can overcome the greater computer limitations necessary for a complete plant risk assessment.

(Note: Initial utility efforts on systems interactions have been based on a success-oriented operational diagram [24-25].)

2.0 INITIAL GUIDANCE FOR DIGRAPH-MATRIX ANALYSIS OF SYSTEMS INTERACTIONS

In this section of the report, the proposed initial guidance for Digraph-Matrix Analysis (DMA) for Systems Interactions is detailed, with additional explanation and examples to enhance clarity. The steps are broken down into substeps as necessary. The four steps are listed in Table 1.

2.1 Step 1: Selection of the Combination of Systems for Detailed Evaluation

This step focuses on the four vital plant safety functions and using event tree methodology results in the identification of combinations of front-line systems among which a system interaction might exist. This includes an understanding of their operating modes as well as the most general types of system interactions.

This step is accomplished in a manner similar to a traditional PRA by event tree analysis. Event tree analysis is an inductive logic technique that sequentially models the progress of events, either success or failure, leading from some initiating event to a series of logical outcomes. An event tree begins with an initiating failure, and then maps out a sequence of events on the system level that forms a set of branches, each of which represents a specific accident sequence whose consequence relates directly to the events in the sequence. Complete event tree analysis requires the identification of all possible initiating events and the development of an event tree for each.

The first step of finding accident sequences (combinations of systems for detailed evaluation) can be accomplished by the following six substeps.

2.1.1 Substep 1a: Study Plant Design and Operating History

The analyst first gathers all pertinent existing information about the plant. A large amount of information is collected, synthesized, and documented to form the basis for subsequent analytical activities. A list of plant systems is developed and reviewed for potential impact on risk. Appropriate sources of information include: design documents, safety evaluation reports, and previous safety studies.

2.1.2 Substep 1b: Development of a List of Accident Initiators

The Reactor Safety Study (WASH-1400) generic list of accident initiators is reviewed to see which apply to the particular plant in question. This list should reflect applicable operating experience. The accident initiators are then grouped in terms of common mitigation requirements.

Accident-initiating events are identified and grouped according to similarity of plant response. Generic lists, operating histories, and plant-specific data can be factored into a generalized engineering process wherein an exhaustive listing of initiating events, including their occurence frequency, are eventually compiled and grouped. Efforts must be made to ensure that the set of initiating events considered is as complete and comprehensive as practical.

There are two major classes of accident initiators, namely LOCAs and power transients.

2.1.3 Substep 1c: Development of Funct [illegible] vent Trees

To avoid unacceptable reactor core damage and a release of unacceptable levels of radioactivity to the site environs, the following basic safety functions have been specified [15]:

- o To maintain the primary coolant inventory.
- o To transfer the heat from the reactor to the ultimate heat sink.
- o To render and keep the entire core subcritical.
- o To maintain the integrity of the containment and control radioactivity releases.

Systems interactions that fail a single (or multiple) vital safety function are of concern.

For each category of accident initiators identified in Substep 1b, the four basic safety functions [15] are further analyzed into subfunctions and the corresponding functional event trees are generated.

To summarize, when the group of initiating everts has been selected, the attendant response of the plant must be determined. This may be accomplished through a functional analysis where safety functions required for each response are defined and ordered in a function event tree. Success criteria for each function are stated in terms of the required collection of systems that perform each individual function. Success criteria are then developed for individual systems and form the basis for characterizing the logic description of the success-oriented top desired event of the system event tree. The value of this approach is the stepwise ordered approach of identifying broad functional considerations with specific systems. It provides a framework for the complex task of sorting system responses.

2.1.4 Substep 1d: Assignment of Front-Line Systems to Functional Event Trees

The safety functions utilized in preparing the functional event trees (Substep 1c) are performed by engineered systems designed specifically for this purpose. In other words, further analyses of the safety functions into simpler functions yields the specific engineered systems. The operability of these systems defines whether the safety functions are performed or not and therefore, completely defines the course of an accident. These systems are called front-line systems (FLS). The success criteria for the FLS should also be defined in this step.

The event-tree headings are defined to be logic statements describing composite events representing the minimal operability states of front-line systems and their required supporting systems. This approach leads to event trees with a minimum number of event tree branch headings, and thus facilitates the understanding of the overall accident-progression path. However, it requires that support systems be included in the system models.

Each event-tree heading must have a definite logic statement of the minimum acceptable complement of equipment of system performance required to successfully accomplish the event described by the event-tree. These success criteria should be stated in discrete hardware terms, such as number of pumps or required flow. The basis for such criteria may be derived from licensing information, which should be recognized as inherently conservative. Alternatively more realistic information can be used, such as results of particular thermal-hydraulics calculations that are supportable and documented.

2.1.5 Substep 1e: Results of Event Tree Analysis

The event tree analysis of Step 1 produces the set of accident sequences. Each accident sequence is a combination of several front-line systems whose success or failure (as specified) results in serious consequences to the plant involving the loss of one or more vital safety functions.

In a simplifying assumption, as is done in PRA, it is necessary to assume that systems required to work in an accident sequence, work with probability equal to one. We, then, are left with accident sequences which are combinations of front-line systems that must fail in order to result in a serious consequence to the plant.

2.1.6 Substep 1f: Assignment of Support Systems to Front-Line Systems

To successfully perform their function, the front-line systems depend on the operability of other support systems. Support systems affect the accident response of a plant only through their effect on the FLS.

To identify the support systems for each front-line system, the following procedures can be followed:

1. The operation of the front-line system is examined in detail, identifying all the necessary inputs as well as all of its outputs. If, for example, the FLS is a fluid system, all potential sources of the fluid should be identified. Next, all the systems with which the FLS interact directly (e.g. as discharge points) or indirectly (e.g. as secondary sides of heat exchangers) should be identified.

2. The power sources necessary for the operation of the active components, e.g., electric power and steam, should be identified.

3. The modes of actuating and/or controlling the system must be identified. In particular, one should examine whether the system is actuated and controlled automatically or by operator action. In both cases the signals necessary to initiate control system or operator action must be identified. The possiblity of manual overriding of automatic control should also be established. In the case of automatic control the type of the controlling system should be identified (e.g., electrical pneumatic) along with the systems associated with each type (e.g., power supply, instrument air).

4. The cooling systems of the various components of the FLS should be identified.

5. The lubrication systems (if any) of the various components of the FLS should be identified.

6. The general location of the FLS should be established. More detailed location identification will support the spatial coupling portion of the study to be carried out by an idependent walk-through evaluation.

The identification of the support systems that contribute to the initiating event are then compiled and used in Step 2.

2.2 Step 2: Constructing a Global Digraph Model For Each Accident Sequence

The resulting accident sequences found in Step 1 are combinations of front-line systems that must fail in order to produce a severe consequence to the plant involving the loss of one or more vital safety functions. In Step 2, a single global digraph model is to be constructed for each of these accident sequences. The construction of this Global Digraph follows the iterative procedure illustrated in Figure 1.

As shown in Figure 1, the analyst will construct the global digraph by a series of expansion steps. These expansions are centered on each of the components identified in the digraph and follow an algorithm. The expansion of each of these components will identify new components which must then be expanded. This procedure will be repeated until the analyst is satisfied that the global digraph contains sufficient detail to allow discovery of all significant failure sources.

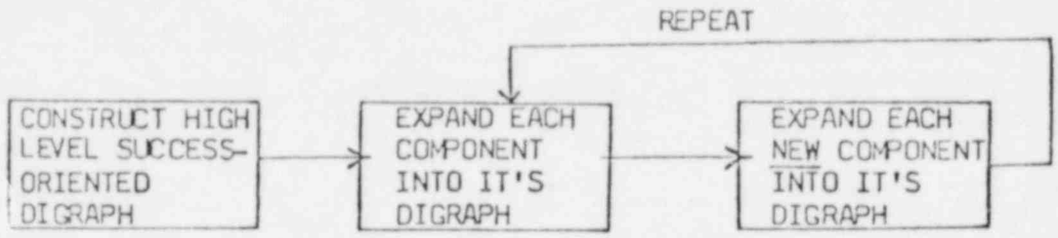


Figure 1. Overview of Construction of Global Digraph.

Using this iterative procedure the analyst can formally assign the expansions of each component to that person most familiar with that component's operation and function*. The expansion of each of the assigned components is performed using a specified algorithm. This algorithm specifies a description of the direct relationships of the component to the other components in the system.

Figure 2 illustrates the procedure for the Global Digraph construction. The procedure follows the broad outline described above and includes substeps which allow the verification of the modeling procedure and the generation of intermediate results prior to the completion of the Global Digraph. These intermediate results are useful for both verifying the input data and for identifying components which can cause system failure before the complete digraph is constructed.

The construction of the Global Digraph in Substep 2a, starts with a single success-oriented digraph built for each accident sequence and includes a combination of front line systems and their support systems. This digraph serves as the foundation on which the Global Digraph will be constructed by

* The construction of the global digraph may involve many specialists each concerned only with his area of expertise.

successive expansions. The next substep is the conversion of the success-oriented digraph into the dual digraph. The subsequent expansions will be performed on this dual. The dual is created by converting all original AND gates to OR gates and all original OR gates to AND gates (see Appendix A for a discussion of dual relationships.)

In Substep 2c, the analyst will convert the dual digraph into the adjacency input required by a Reachability code [19-23]. The Reachability code* used in Substep 2d will use a path finding procedure capable of Boolean manipulation to process adjacency input to find all common mode failures (Singletons). The code should provide a printed list of the input data, the component mnemonics, and the singletons. In Substep 2e, the analyst will use this output to identify any errors in the input data. After correcting any errors in the input data, the analyst will process the adjacency information through the Reachability code for all Singletons and Doubletons.

The model is now expanded through the use of a detailed model (called a Unit Model) for each component based on the Unit Models to be contained in a Unit Model Library. Then Substep 2g begins the expansion process which will eventually result in the detailed Global Digraph. The expanded digraph is now processed in the same manner as the original digraph (Substeps 2h-2j). After analyzing the results from this stage of expansion the analyst will again expand the digraph around the new components introduced in the preceeding step. This iterative expansion and analysis procedure leads ultimately to the Global Digraph and all of the Singleton and Doubleton cut-sets of the accident sequence. As the digraph grows in size it may eventually exceed the size limitation of the computer code. At this point it will become necessary to partition it into subdigraphs and then to combine the results of processing each subdigraph. This process will be described in Step 3.

---

* There are at least three versions of Reachability codes capable of processing conditioned digraphs, two at LLNL -- CLAMOR and SQUEAK, and one at AIP -- NEWARS. The first two codes were designed for the LLNL CDC7600 and the third code is written in ANSI Fortran 4 and now operates on a Digital Equipment Corporation PDP 11/34.

As the substeps are described in greater detail in the following section, a simple example will be carried through all of Step 2 as an illustration. The simplified version of a corewater injection system shown in Figure 3 will serve as the example problem. Valves (V), pumps (P), normally open (NO), normally closed (NC) reserve water storage tank (RWST), notations will be carried through the example.

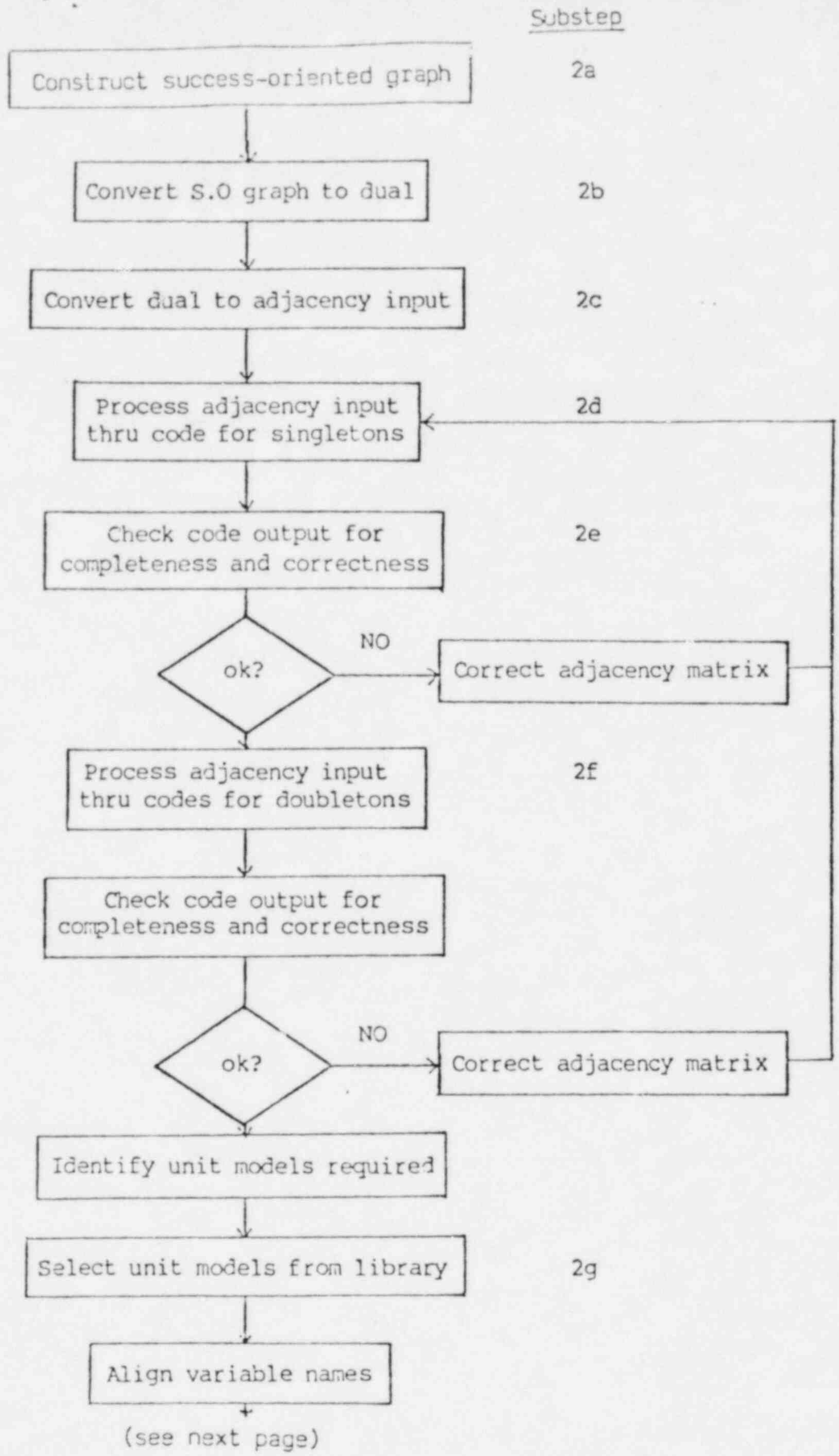


Figure 2. Overview of Step 2

(see prior page)

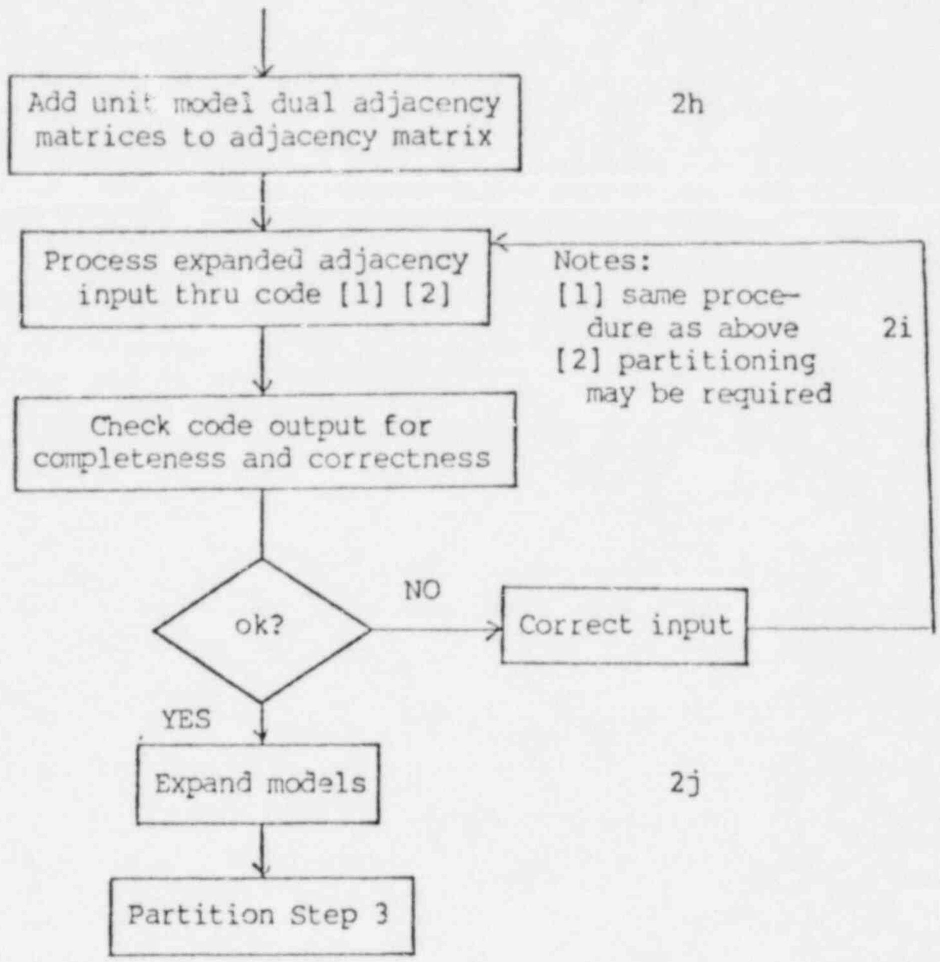


Figure 2 (continued). Overview of Step 2

2.2.1 Substep 2a: Construct a Success-Oriented Digraph Model for each Accident Sequence

In order to construct a success-oriented (S.O.) digraph model for each accident sequence which will serve as a beginning foundation for the Global Digraph, it is necessary to first identify each front-line system within a given accident sequence and each support system required by the front-line systems. Then all pertinent system information is gathered including: schematic diagrams, piping and instrumentation diagrams, specifications, and operational and emergency procedures.

Once this comprehensive information is available, the analyst begins to piece it together into a coherent success-oriented operational diagram of the systems within a specified accident sequence. This diagram must include: (1) the success-oriented schematic representation of each front-line system, (2) the interconnectivity between front-line systems, (3) the success-oriented schematic representation of required support systems, (4) the interconnectivity between support systems and (5) operational, emergency, and human actions must be integrated into the diagram.

The S.O. digraph will at this stage not capture all of the interconnections between subsystems nor will it identify all of the components which are necessary for system function. It will, however, identify all of the subsystems, components, and procedures which are directly associated with the front-line systems. This digraph will provide the framework around which a detailed global model of the system will be built through expansion.

The analyst will construct the success-oriented digraph using AND and OR gates to explicitly model the relationships between components necessary for successful functioning of the collective group of subsystems. The algorithm for choosing the appropriate gate is:

If a component requires the successful operation of two or more components which supply it, these supply components are connected to the component by an AND gate. For example, a pump may require both electric power and lubrication. The use of the AND gate is shown in Figure 4a.

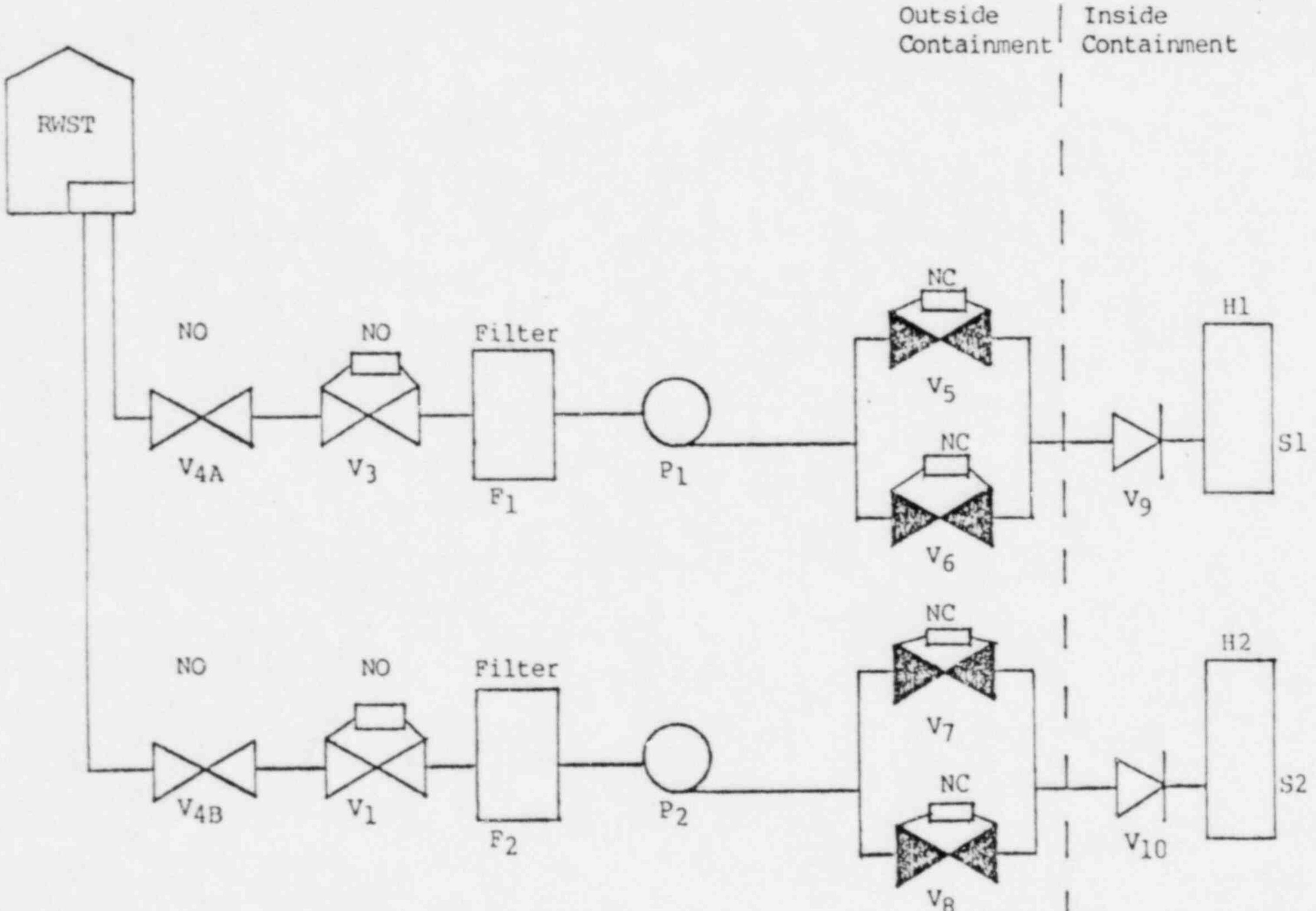
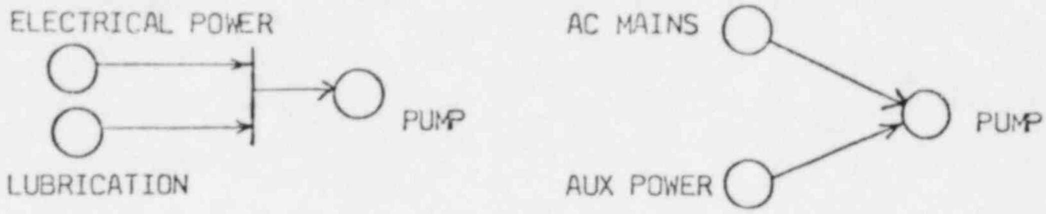


Figure 3. Simplified Cooling System Example.

If a component requires the successful operation of only one of a group of components which supply it, these supply components are connected to the component by an OR gate. For example, a pump may be supplied with electrical power from the ac mains or from an auxiliary generator. The use of the OR gate is shown in Figure 4b.



(a) Use of the AND gate. (b) Use of the OR gate.

Figure 4. Logic Symbol Conventions for Success-Oriented Digraph.

The direction of flow is indicated by the arrows on the edges between the components of the system. This integration of connectivity, operational information, and logic is called a conditional digraph for the cooling system example of Figure 3 and is shown in Figure 5. The pumps (P1 and P2) require both a supply of water and a control signal to operate, thus there is an AND gate which joins the Filter and Controller to the Pump. The spray into Containment will occur if either a spray occurs through Spray Nozzle 1 or Spray Nozzle 2, thus the spray nozzles are joined to the containment by an OR gate.

This integration of connectivity, operational information, and logic produces a success-oriented logic diagram commonly referred to as a digraph.

2.2.2 Substep 2b: Forming the Dual Digraph Model

The success-oriented digraph model produced in Substep 2a is now converted into a dual-digraph model. (See Appendix A and [25, 26] for an explanation of dual and complement relationships.) The dual digraph model is formed simply by changing all AND gates in the original model into OR gates and vice versa. The dual digraph for the cooling system example is shown in Figure 6.

Figure 5. Success Oriented System Digraph.

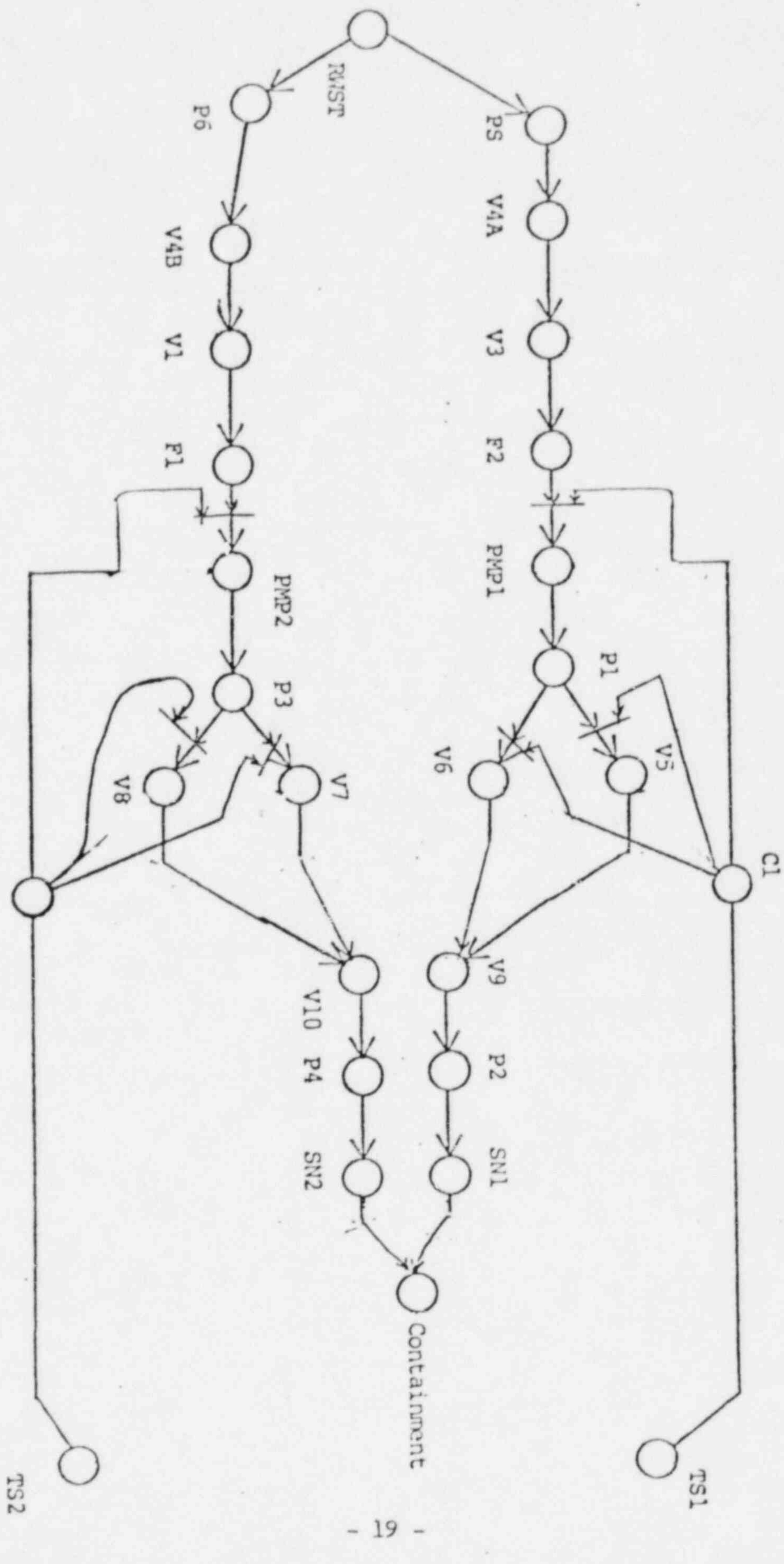
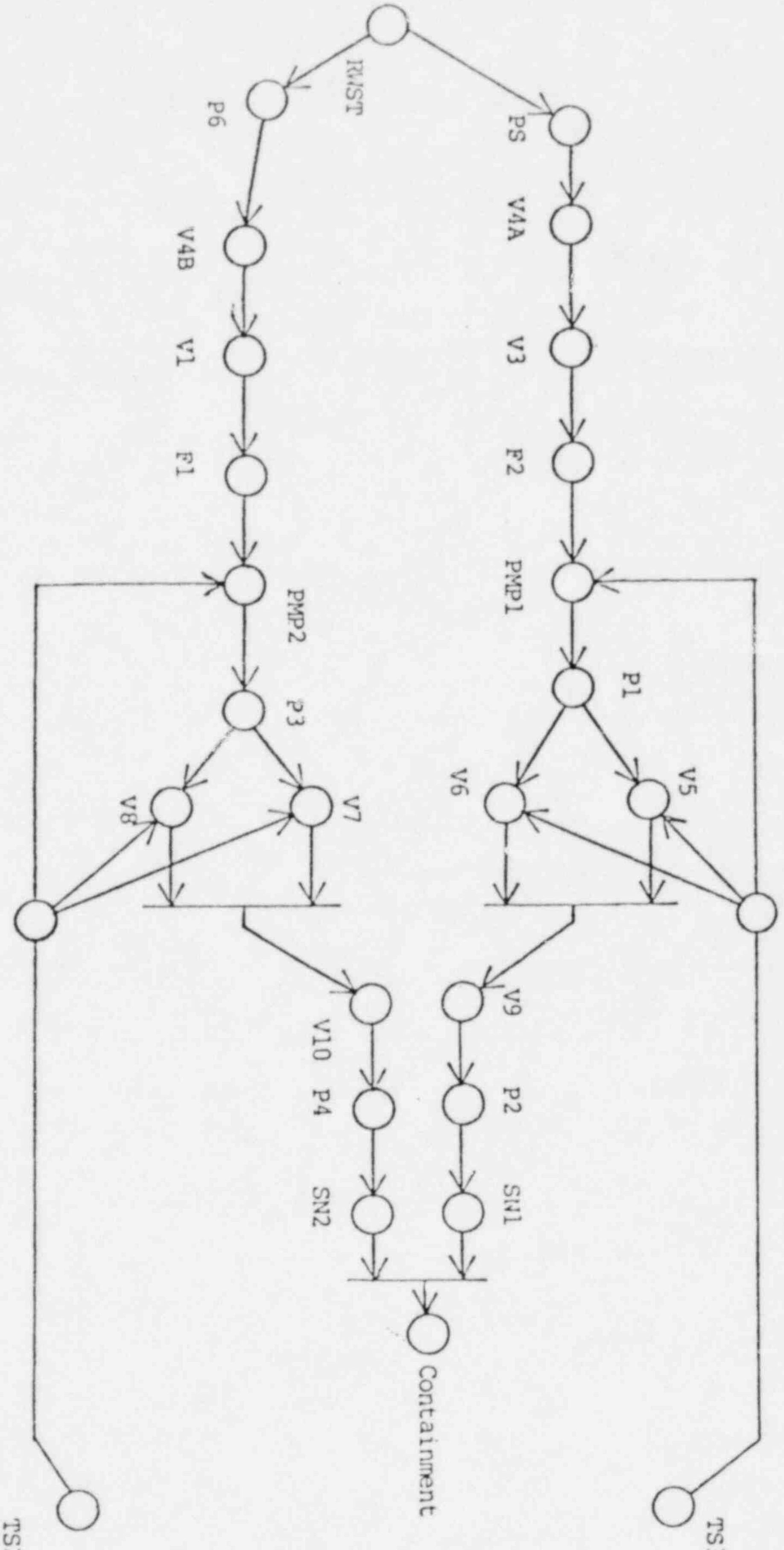
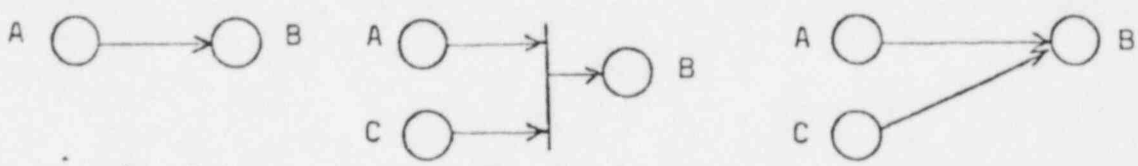


Figure 6. Dual Digraph of Simlified Cooling System Example.



2.2.3 Substep 2c: Forming Adjacency Matrix Input

An adjacency matrix [20,21,27,28] can be used as a representation of a graph. The matrix, which is always square, indicates whether there can be flow from node i to node j. An entry of 1 in element A(i,j) indicates that component i is unconditionally and directly connected to component j and that flow goes from i to j. An entry of 0 in element A(i,j) indicates that there is no direct connection between component i and j. Any other entry for a pair of nodes (i,j) indicates that there is flow from i to j only with the condition indicated by the entry. The use of the conditional adjacency is a way of representing an AND gate in the matrix. Figure 7 presents a summary of the conventions used in creating the conditional adjacency matrix.



	A	B
A	0	1
B	0	0

Direct Connection

	A	B	C
A	0	C	0
B	0		0
C	0	A	0

AND Gate

	A	B	C
A	0	1	0
B	0	0	0
C	0	1	0

OR Gate

Figure 7. Adjacency Matrix Conventions.

To minimize the effort in inputting data into the Reachability code, only the nonzero matrix elements are entered. The input for the available present versions of such codes (CLAMOR, SQUEAK, NEWARS) follow the format:

from,to,condition

Therefore the input for the adjacency matrices of Figure 7 would be:

Direct Connection
A,B,1

AND Gate
A,B,C
C,B,A

OR Gate
A,B,1
C,B,1

The adjacency input for the dual digraph of Figure 6 is shown in Figure 8.

```
RWST,P5,1
P5,V4A,1
V4A,V3,1
V3,F2,1
F2,PMP1,1
PMP1,P1,1
P1,V5,1
P1,V6,1
V5,V9,V6
V6,V9,V5
V9,P2,1
P2,SN1,1
SN1,CONT,SN2
SN2,CONT,SN1
TS1,C1,1
C1,PMP1,1
C1,V5,1
C1,V6,1
RWST,P6,1
P6,V4B,1
V4B,V1,1
V1,F1,1
1F,PMP2,1
PMP2,P3,1
P3,V7,1
P3,V8,1
V7,V10,V8
V8,V10,V7
V10,P4,1
P4,SN2,1
TS2,C2,1
C2,PMP2,1
C2,V7,1
C2,V8,1
0,0,0
```

Figure 8. Adjacency Input for Dual Digraph.

Number of Variables = 29

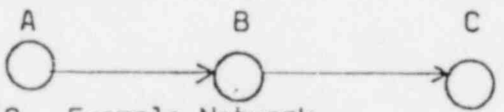
1	1
2	RWST
3	P5
4	V4A
5	V3
6	F2
7	PMP1
8	P1
9	V5
10	V6
11	V9
12	P2
13	SN1
14	CONT
15	SN2
16	TS1
17	C1
18	P6
19	V4B
20	V1
21	F1
22	PMP2
23	P3
24	V7
25	V8
26	V10
27	P4
28	TS2
29	C2

Figure 9. Variable List for Dual Digraph.

(The 0,0,0 entry at the end of the list is used to indicate the end of the data.) This adjacency information is then processed by a code which reads the alphanumeric input and assigns a node number to each of the variables used. The variable list found by this code for the example is shown in Figure 9. This list should be checked by the analyst for missing or extraneous variables (this is a common source of error in all PRA types of analysis). The code also converts the alphanumeric adjacency input into a numeric adjacency input which is used by the Reachability code in the next substep.

2.2.4 Substep 2d: Processing Adjacency Matrix

The adjacency input created by the previous substep is now input for the Reachability code. This code links all of the connected components of the adjacency matrix to determine all unique paths through the dual digraph. This operation can be illustrated by a few simple examples. First consider the network and corresponding adjacency elements shown in Figure 10.



Input

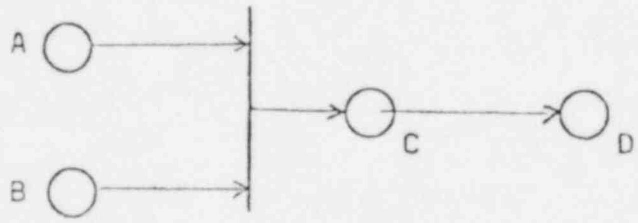
A,B,1
B,C,1

Figure 10. Example Network.

The reachability calculation will determine the path between all pairs of nodes finding that A is connected to C. The full set of reachability elements is

A,B,1
B,C,1
A,C,1

As an example of a reachability calculation in the presence of conditional information (AND gates), consider the network and adjacency information shown in Figure 11.



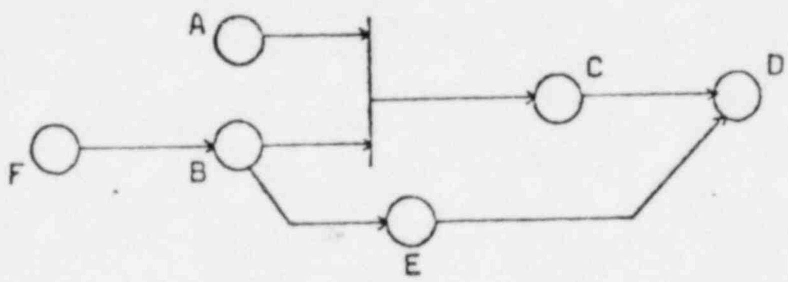
A,C,B
B,C,A
C,D,1

Figure 11. Example Network with AND Gate.

The reachability calculation will determine that node A is connected to node D if B and that node B is connected to node D if A. In the input notation the full set of reachability elements is:

A,C,B
B,C,A
C,D,1
A,D,B
B,D,A

The reachability calculation determines all unique paths between all pairs of nodes. That is, the technique will find all alternate paths in the network. As an example, consider the network in Figure 12.



A,C,B
B,C,A
B,E,1
C,D,1
E,D,1
F,B,1

Figure 12. Example of Network with Multiple Paths.

The reachability calculation would find the following set of reachability elements for this network.

Adjacency Information

A,C,B
B,C,A
B,E,1
C,D,1
E,D,1
F,B,1

New Paths

A,D,B
A,D,F
A,C,F
F,D,1
F,E,1
F,C,A
F,D,A
B,D,1

It should be noted that there is no reachability element B,D,A. This path is dominated by the path B,D,1. This effect can be seen by consideration of the equivalent Boolean equation representation

$$D = B * A + B$$

which by the absorption rule becomes

$$D = B.$$

The analyst will input the adjacency elements from the previous substep into a Reachability code which uses the above concepts to determine all single point failure modes (singletons). A singleton occurs when an unconditional path exists from any node to the end node. Figure 13 illustrates a system with a singleton failure.

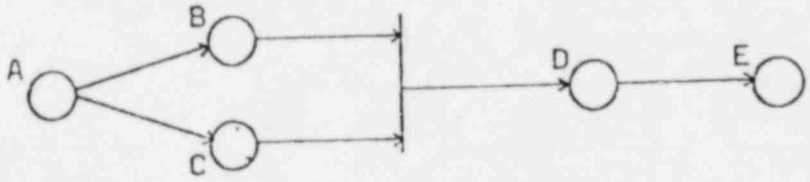


Figure 13. Example of a System with a Singleton.

It is important to note that using a reachability code to find the singletons of a path set is computationally more efficient than finding the singletons of a fault tree using codes such as SETS or FTAP. The ouptut of the Reachability code for the example is shown in Figures 14 and 15. The unconditional adjacency matrix is shown in the first of these figures. This matrix shows all, direct connections between node pairs. For example, in Row 2 there are numerous ones, indicating 2 is directly connected to many components. Any connection which requires a condition is not displayed here. The second figure shows the singleton reachability matrix. This matrix contains all single failure modes. For example, it can be seen that node 2 is a singleton to node 14. Node 14 is the Spray into containment and node 2 is the reservoir (RWST).

1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0

Figure 14. Adjacency Matrix for Dual Digraph.

														14															
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	0	0
3	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1	1	1	1	1	1	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	1	1	1	1	1	1	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	1	1	1	1	1	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	1	1	1	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	1	1	1	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1	1	1	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	1	1	1	0	1
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	1	1	1	0	0

Figure 15. Singleton Matrix for Dual Digraph.

2.2.5 Substep 2e: Consistency and Completeness Check

The analyst will now review the outputs of the Reachability code for consistency with the paths that he can determine by eye from the dual digraph. If any errors are detected, the analyst will correct the adjacency input and repeat the preceding steps. Once the analyst is satisfied with the results, he will instruct the code to find all doubletons.

2.2.6 Substep 2f: Finding Singleton/Doubleton Minimum Cut-Sets

The Reachability code will find all singletons and doubletons from the input information contained in the adjacency element list. A doubleton is a pair of component failures which will cause system failure. Figure 16 illustrates a system with a doubleton. In this example, both A and C or B must fail to cause E.

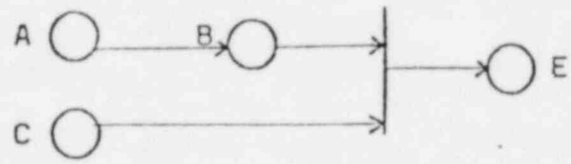


Figure 16. Example of a System with a Doubleton.

The matrix shown in Figure 17 gives all of the singletons and doubletons for the example of the simplified cooling system of Figure 3. An asterisk in row i , column 1 indicates that component 1 is a singleton.

The row and columns indicate the number of the components which are failure modes of the accident sequence. For example, component 3 and component 15 are a doubleton for 29 (the spray into containment). In this way, the several hundred doubletons can be completely listed.

2.2.7 Substep 2g: Constructing Unit Models

The analyst will now expands the basic digraph by using unit models for each of its components. These unit models describe the direct dependence of a component on other components and allow the analyst to expand the information in the digraph to contain these dependencies. The unit model for a pump is shown in Figure 18a. This pump requires lubrication and power and a control signal to operate. Thus the supporting components are connected to the pump via an AND gate. Unit models will be attached to the dual digraph, hence the dual form (as shown in Figure 18b) is used.

Many components will be supplied from a redundant system. For example, the pump might receive its electrical power from either the electrical mains or an auxiliary power system. In this case the dual digraph will show the supply components connected to the pump by an AND gate as shown in Figure 19.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
1		*												*															
2	*		*	*	*	*	*	*	*	*	*	*	*		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
3			*												*			*	*	*	*	*	*			*	*	*	*
4			*												*			*	*	*	*	*	*			*	*	*	*
5			*												*			*	*	*	*	*	*			*	*	*	*
6			*												*			*	*	*	*	*	*			*	*	*	*
7			*												*			*	*	*	*	*	*			*	*	*	*
8			*												*			*	*	*	*	*	*			*	*	*	*
9			*																										
10			*																										
11			*												*			*	*	*	*	*	*			*	*	*	*
12			*												*			*	*	*	*	*	*			*	*	*	*
13			*												*			*	*	*	*	*	*			*	*	*	*
14	*																												
15			*	*	*	*	*	*			*	*	*			*	*												
16			*												*			*	*	*	*	*	*			*	*	*	*
17			*												*			*	*	*	*	*	*			*	*	*	*
18			*	*	*	*	*	*			*	*	*		*	*	*												
19			*	*	*	*	*	*			*	*	*		*	*	*												
20			*	*	*	*	*	*			*	*	*		*	*	*												
21			*	*	*	*	*	*			*	*	*		*	*	*												
22			*	*	*	*	*	*			*	*	*		*	*	*												
23			*	*	*	*	*	*			*	*	*		*	*	*												
24			*																										
25			*																										
26			*	*	*	*	*	*			*	*	*		*	*	*												
27			*	*	*	*	*	*			*	*	*		*	*	*												
28			*	*	*	*	*	*			*	*	*		*	*	*												
29			*	*	*	*	*	*			*	*	*		*	*	*												

Figure 17. Singletons and Doubletons for Example.

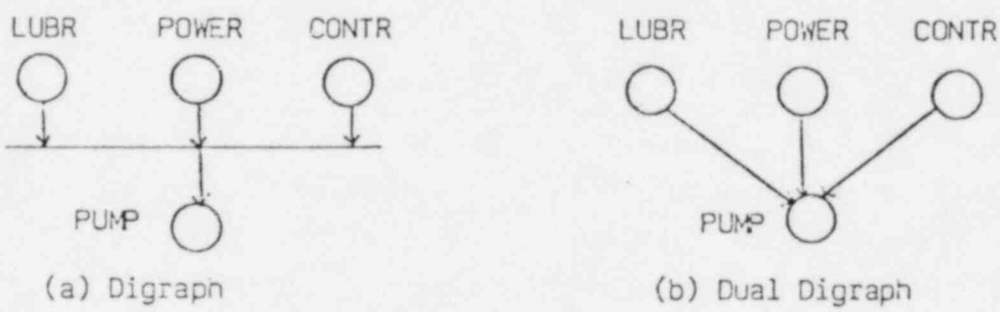


Figure 18. Pump Unit Model.

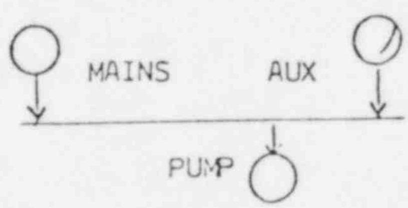


Figure 19. Unit Model with Redundant Supplies for Pump.

A library of these unit models will be developed into a Digraph-Matrix Handbook. A typical set of unit models is shown in Figure 20. The analyst will select the appropriate unit model from this handbook and make the variable names consistent with the variables used in the basic dual digraph in preparation for the expansion of the dual digraph. The addition of unit models for the controlled valves of our example is shown in Figure 21.

2.2.8 Substep 2h: Expand Adjacency Matrix

The analyst will now attach the adjacency information from each unit model to the end of the original adjacency list. The attachment of a set of unit models for the valves of our simplified cooling system example to the adjacency input is shown in Figure 22 with the variables as shown.

2.2.9 Substep 2i: Repeat Reachability Processing

The expanded adjacency information is now processed through the same Reachability procedure as above. After the input data has been corrected, the data is processed for all singletons and doubletons. Figure 23 shows the results of this processing for the expanded adjacency input of Figure 22.

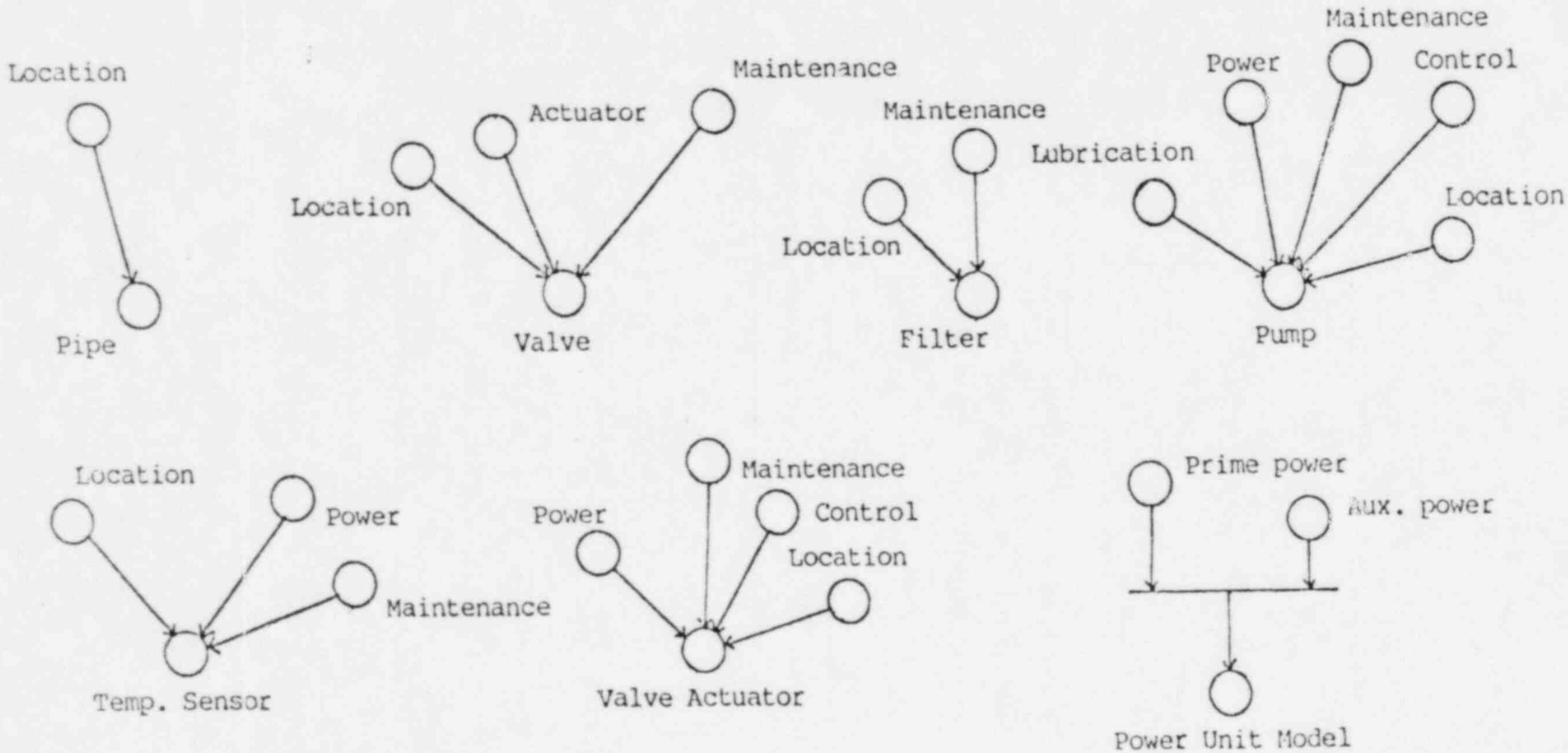
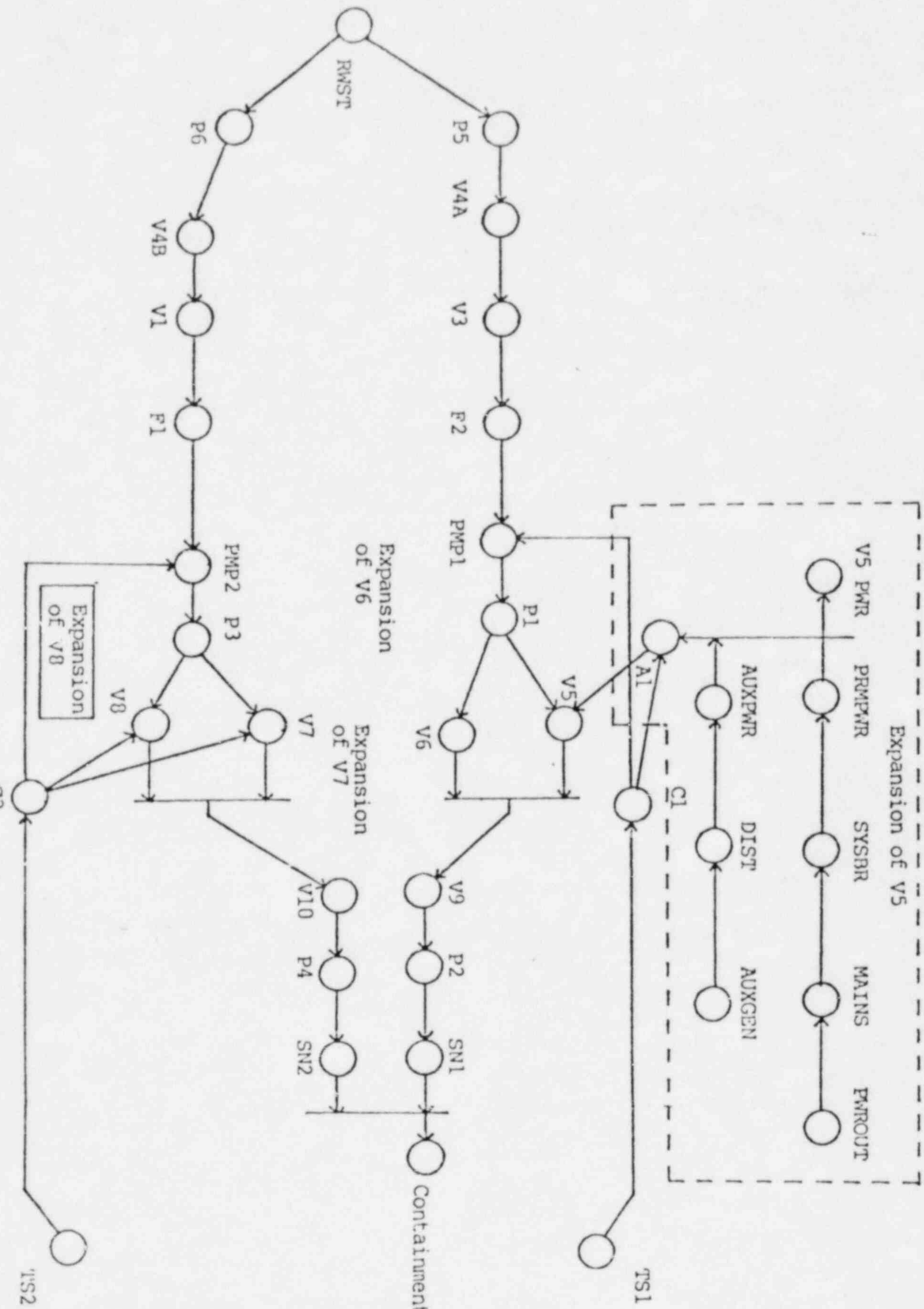


Figure 20. Unit Models (Duals).

Figure 21. Expanded Dual System Digraph.



RWST	V6A1,V6,1	
P5,V4A,1	C1,V6A1,1	
V4A,V3,1	V5PWR,V5A1,1	
V3,F2,1	PRMPWRV5,V5PWR,AUXPWRV5	Unit
F2,PMP1,1	AUXPWRV5,V5PWR,PRMPWRV5	Model
PMP1,P1,1	SYSBRV5,PRMPWRV5,1	for
P1,V5,1	DISTV5,AUXPWRV5,1	Valve
P1,V6,1	MAINS,SYSBRV5,1	V5
V5,V9,V6	PWROUT,MAINS,1	
V6,V9,V5	AUXGEN,DISTV5,1	
V9,P2,1	V6A1,V6,1	
P2,SN1,1	C1,V6A1,1	
SN1,CONT,SN2	V6PWR,VA1,1	Unit
SN2,CONT,SN1	PRMPWRV6,V6PWR,AUXPWRV6	Model
TS1,C1,1	AUXPWRV6,V6PWR,PRMPWRV6	for
C1,PMP1,1	SYSBRV6,PRMPWRV6,1	Valve
C1,V5,1	DISTV6,AUXPWRV6,1	V6
C1,V6,1	MAINS,SYSBRV6,1	
RWST,P6,1	PWROUT,MAINS,1	
P6,V4B,1	AUXGEN,DISTV6,1	
V4B,V1,1	PWROUT,MAINS,V1	
V1,F1,1	AUXGEN,DISTV6,1	
F1,PMP2,1	V7A1,V7,1	
PMP2,P3,1	C2,V7A1,1	
P3,V7,1	V7PWR,V7A1,1	Unit
P3,V8,1	PRMPWRV7,V7PWR,AUXPWRV7	Model
V7,V10,V8	AUXPWRV7,V7PWR,PRMPWRV7	for
V8,V10,V7	SYSBRV7,PRMPWRV7,1	Valve
V10,P4,1	DISTV7,AUXPWRV7,1	V7
P4,SN2,1	MAINSSYSBRV7,1	
TS2,C2,1	PWROUT,MAINS,1	
C2,PMP2,1	AUXGEN,DISTV7,1	
C2,V7,1		
C2,V8,1		

Figure 22. Input Data for Expanded System.

V8A1,V8,1	
C2,V8A,1	Unit
V8PWR,V8A1,1	Model
PRMPPWRV8,V8PWR,AUXPWRV8	for
AUXPWRV8,V8PWR,PRMPWRV8	Valve
SYSBRV8,PRMPWRV8,1	V8
DISTV8,AUXPWRV8,1	
MAINS,SYSBRV8,1	
PWROUT,MAINS,1	

Figure 22 (continued). Input Data for Expanded System.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	36	37	38	39	40	45	46	51	52
1	*												*																											
2	*		*	*	*	*	*	*	*	*	*	*	*		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*				*	*	*	*	*	*
3			*												*			*	*	*	*	*	*			*	*	*	*											
4			*												*			*	*	*	*	*	*			*	*	*	*											
5			*												*			*	*	*	*	*	*			*	*	*	*											
6			*												*			*	*	*	*	*	*			*	*	*	*											
7			*												*			*	*	*	*	*	*			*	*	*	*											
8			*												*			*	*	*	*	*	*			*	*	*	*											
9			*																																					
10			*																																					
11			*												*			*	*	*	*	*	*			*	*	*	*											
12			*												*			*	*	*	*	*	*			*	*	*	*											
13			*												*			*	*	*	*	*	*			*	*	*	*											
14	*																																							
15			*	*	*	*	*	*			*	*	*			*	*																							
16			*												*			*	*	*	*	*	*			*	*	*	*											
17			*												*			*	*	*	*	*	*			*	*	*	*											
18			*	*	*	*	*	*			*	*	*		*	*	*																							
19			*	*	*	*	*	*			*	*	*		*	*	*																							
20			*	*	*	*	*	*			*	*	*		*	*	*																							
21			*	*	*	*	*	*			*	*	*		*	*	*																							
22			*	*	*	*	*	*			*	*	*		*	*	*																							
23			*	*	*	*	*	*			*	*	*		*	*	*																							
24			*																																					
25			*																																					
26			*	*	*	*	*	*			*	*	*		*	*	*																							
27			*	*	*	*	*	*			*	*	*		*	*	*																							
28			*	*	*	*	*	*			*	*	*		*	*	*																							
29			*	*	*	*	*	*			*	*	*		*	*	*																							
30			*																																					
31			*																																					
36																																	*							

Figure 24. Result of the singleton and doubleton processing on expanded graph.

?.2.10 Substep 2j: Expand Model

New components will have been introduced by the expansion step. A unit model should be added for each of these new components and the resulting expanded adjacency matrix processed again. This iterative expansion process will generate a constantly more detailed (and larger) Global Digraph which will utlimately represent as much detail as the analyst feels is necessary to analyze the accident sequence. As the adjacency matrix expands, it may exceed the size which can be processed at one time by the Reachability code. To overcome this problem, the analyst must rigorously partition the digraph into smaller subdigraphs which can be Boolean processed separately. If the partitioning is performed rigorously, the singletons and doubletons from the analysis of these subgraphs can then be directly combined into the singletons and doubletons of the Global Digraph without introducing error of Boolean independence. Partitioning can and should be performed by the computer since it is easy for a human to incorrectly partition the network. Partitioning will be discussed in more detail in the following section as part of Step 3.

2.3 Step 3: Partitioning the Global Digraph Model into Independent Subdigraphs

The global digraph produced from Step 2 is the dual-digraph of the expanded success-oriented operational model for the specified accident sequence. The digraph and its corresponding adjacency matrix will grow larger with each expansion step. At some size, the matrix will exceed the memory space limits of the Reachability code or will take excessive amounts of computer processing time. There are two ways of overcoming these computer limitations. First, we can separate the global digraph into independent subdigraphs which may be solved through Boolean minimization as independepent modules.

Partitioning into independent subdigraphs can be accomplished according to the following definition:

A connected graph G is separable (capable of being partitioned) if there exists a subgraph g in G such that the complement of g (g') and g have only one vertex in common.

Wherever possible, the global digraph can be partitioned into independent subdigraphs directly by the analyst. In addition, a second way to overcome computer limitations deals with matrix partitioning and Boolean reduction that takes advantage of Boolean absorption within the Reachability code used for the analysis (available on CLAMOR). The adjacency matrix can be partitioned into submatrices which can later be recombined into the Global Digraph. The processing of the submatrices will lead to smaller submatrices. That is, nodes which are not on the boundary of the graph represented by the submatrix will be eliminated through Boolean absorption before the recombination.

The partitioning/recombination procedure must not eliminate any singletons or doubletons from the Global Digraph. The steps in this partitioning procedure are shown in Figure 24, and will be briefly discussed in the following subsections. Figure 25 schematically illustrates this partitioning, reduction, recombination, expansion process. It must be emphasized that the partitioning should be performed by the computer and that the operation should be transparent to the analyst. It will appear that the Reachability code is processing the full global adjacency matrix and therefore the following explanation will not be as detailed as that provided for Step 2.

2.3.1 Substep 3a: Partitioning the Global Digraph Model

The global digraph is partitioned into independent subdigraphs by the analyst according to the above definition. Each subdigraph is labeled (i=1,...n). If no independent subdigraphs are found a partitioner subroutine, as follows, can be used that takes advantage of Boolean absorption with the processing matrices.

The Reachability code is based on the basic structure success-oriented dual digraph and the unit model expansions around this graph. The unit model expansions from "natural" partitions due to the operational and geographic considerations. That is, one would expect most of the components connected to a given valve to be different than components attached to a different valve. As an illustration of this natural partitioning, consider the global dual digraph shown in Figure 25a.

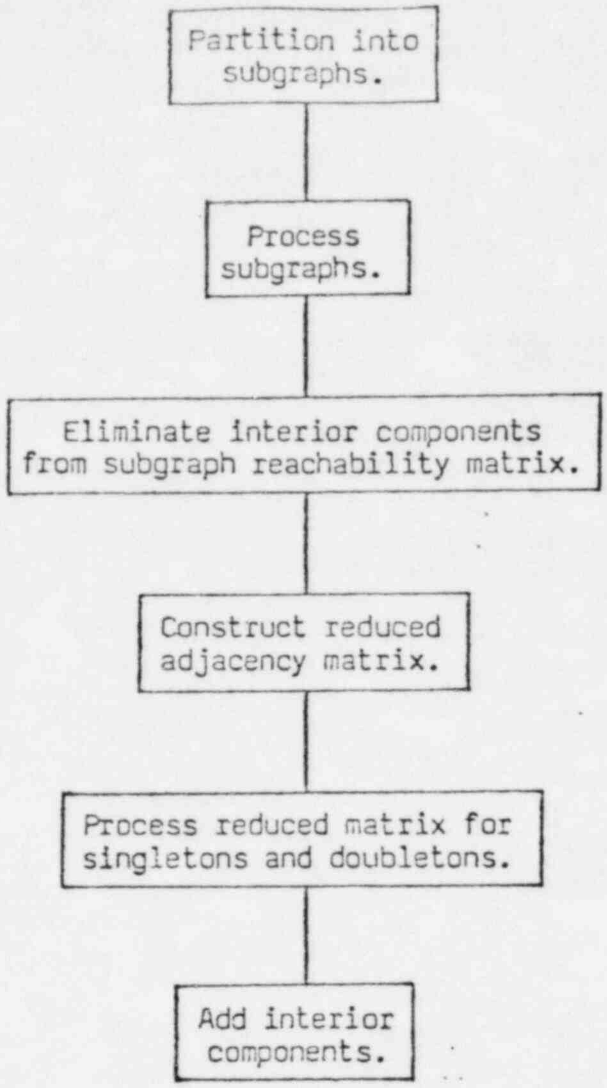
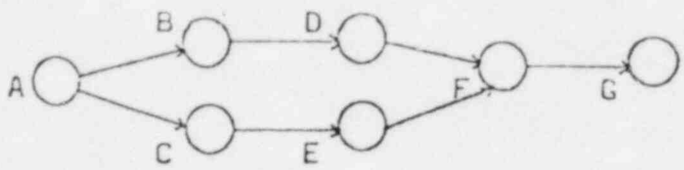


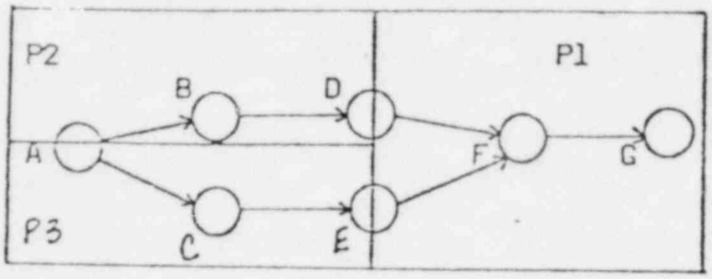
Figure 24. Partitioning the Global Digraph.



(a) Global digraph.

	A	B	C	D	E	F	G
A	0	1	1	0	0	0	0
B	0	0	0	1	0	0	0
C	0	0	0	0	1	0	0
D	0	0	0	0	0	1	0
E	0	0	0	0	0	1	0
F	0	0	0	0	0	0	1
G	0	0	0	0	0	0	0

(b) Global Adjacency Matrix.



(c) Digraph Partitions.

	A	B	C
A	0	1	0
B	0	0	1
D	0	0	0

P2

	D	E	F	G
D	0	0	1	0
E	0	0	1	0
F	0	0	0	1
G	0	0	0	0

P1

	A	C	E
A	0	1	0
C	0	0	1
E	0	0	0

P3

(d) Partition Adjacency Matrices

	A	B	D
A	0	1	1
B	0	0	1
D	0	0	0

P2

	D	E	F	G
D	0	0	1	1
E	0	0	1	1
F	0	0	0	1
G	0	0	0	0

P1

	A	C	E
A	0	1	1
C	0	0	1
E	0	0	0

P3

(e) Reachability Matrices for Partitions.

A,B = D D,E,F = G A,C = E

(f) Singletons for partitions.

	A	D	E	G
A	0	1	1	0
D	0	0	0	1
E	0	0	0	1
G	0	0	0	0

(g) Reduced Adjacency Matrix.

	A	D	E	G
A	0	1	1	1
D	0	0	0	1
E	0	0	0	1
F	0	0	0	0

(h) Reduced Reachability Matrix.

A becomes A,C,E
D becomes D,F
E becomes E,F

therefore the full set is A,B,C,D,E,F

(i) Full Set of Singletons.

Figure 25. Partitioning Process.

As can be seen, each of the two "unit models" added to the structure at F share only one component with each other A and one component with the basic structure F. In addition, each of the three partitions shown in Figure 25c contains components which do not link outside of the partitions and each of the subgraphs could be reduced (by Boolean Absorption) to include only those components which link outside of the partition. Components which do not come from any other component are defined as lying on the boundary of a partition. By removing the components which are fully contained within a partition, the size of the adjacency matrix can be reduced. The code should identify these interior components and record their identity. The actual selection of subgraph partitions is performed by an algorithm which traces through the graph backwards from the components of the basic structure dual digraph. The tracing for each component continues until all subgraph with the following conditions exists:

(a) Number of components less than or equal to the maximum size the Reachability code can process.
(b) A set of "interior nodes" which can be eliminated exists in each subgraph.

It is not necessary to make the subgraphs disjoint when conducting the Boolean absorption process. That is, subgraphs may share common components. These shared components will not be interior nodes and hence will not be eliminated in the subsequent processing. The next step is to process each subgraph through the Reachability code.

2.3.2 Substep 3b: Reachability Processing of Subdigraphs

The adjacency matrices of the subgraph partitions are now individually processed through the Reachability code. The Reachability matrices for each of the subgraphs are shown in Figure 25e.

2.3.3 Substep 3c: Interior Component Elimination

Components which are totally interior to the partitions are now identified and eliminated (on the basis of Boolean absorption) from the subgraph reachability matrices. The unreduced subgraph reachability matrices are retained for use later in determining if these "eliminated" components are singletons or doubletons of the Global Digraph.

2.3.4 Substep 3d: Construction of the Reduced Adjacency Matrix

The reduced reachability matrices of the subgraphs are now combined into a "reduced global digraph adjacency matrix." This is shown in Figure 25g. This matrix is not the same as the original adjacency matrix with interior components eliminated. This matrix contains all of the connectivity information between non-eliminated components that is contained in the original adjacency matrix, whereas a component eliminated adjacency matrix would not.

2.3.5 Substep 3e: Reachability Processing on Reduced Adjacency Matrix

The reduced matrix is now processed by the Reachability code. This step links up the partitions. The result of this processing is shown in Figure 25h. In this processing sequence, the size of the global adjacency matrix has been reduced, thus overcoming the size constraint of the code. The reduced reachability matrix contains all singletons and doubletons for the components which have not been eliminated. These eliminated components will now be considered.

2.3.6 Substep 3f: Addition of Eliminated Components

The adjacency matrices for the partitions contain all essential information about the eliminated components. Each of these components which is connected to a component which is a singleton or part of a doubleton in the reduced reachability matrix must be considered as a potential singleton or doubleton. The type of connection is important. If there is a direct (unconditional) element between the component in the reduced matrix and the interior component, then the interior component has the same impact as the component in the reduced matrix. If the component has a conditional relationship, the effect of this relationship must be considered.

Each accident sequence would result in the violation of one of the four basic safety functions. In this step, the accident sequences of Substep 3d are organized according to the four basic safety functions. Within each function, the reachability matrices are combined to provide a single such matrix for each major function.

Once all singleton and doubleton minimum cut-sets of the global digraph model of the accident sequence have been found, the following Step 4 will evaluate and rank order them.

2.4 Step 4: Evaluate Singletons on the Basis of Probability and Display Answers

Once the singleton and doubleton minimum cut-sets for all accident sequences have been found, it is necessary to evaluate the systems interactions by a ranking criteria. These can be evaluated through normal PRA techniques such as risk which combines the accident sequence consequence with probability of failure.

The ranking evaluation will be displayed by listing minimum cut-sets of the plant functions in order of risk value.

Singleton minimum cut-sets are likely to be considered unacceptable in all cases. Doubletons, however, can form the starting point for the multidisciplined team of experts that are to physically review the plant for shared environmental conditions.

3.0 CONCLUSIONS

Recent events such as Three Mile Island-2, Brown's Ferry-3, and Crystal River-3 have demonstrated that complex accidents can occur as a result of systems interactions (common cause/mode failure). The NRC is preparing an initial guidance that will aid industry in analyzing systems interactions. Part of that guidance covers enhanced PRA approaches for solving the systems interaction problem and has already been well documented. However, additional guidance for an alternative approach utilizing Digraph-Matrix techniques is now being developed. This report contains a four step procedure that will aid the development of the initial guidance for digraph-matrix analysis for systems interactions.

References

1. G. E. Cummings, "Operator/Instrument Interactions During the Three Mile Island Incident," IEEE Symp. Nucl. Power Sys., October 19, 1979.

2. C. Michelson, OAEOD, memorandum to H. R. Denton, NRR, U.S. Nuclear Regulatory Commission, "Potential for Unacceptable Interaction Between the Control Rod Drive System and Non-essential Control Air System at the Brown's Ferry Nuclear Plant," August 18, 1980.

3. S. Rubin and G. Lanik, U.S. Nuclear Regulatory Commission, "Report on Brown's Ferry-3, Partial Failure to Scram Event on June 28, 1980," July 30, 1980 (with Executive Summary).

4. U.S. Nuclear Regulatory Commission, "Transient Response of Babcock & Wilcox-Designed Reactors," U.S. Nuclear Regulatory Commission Report NUREG-0667, May 1980.

5. U.S. Nuclear Regulatory Commission, "The Approach to Systems Interactions in LWRs," Reliability and Risk Assessment Branch, June 1981 (Draft).

6. U.S. Nuclear Regulatory Commission, "Verbatim Transcript of Advisory Committee on Reactor Safeguards, Fluid Dynamics Subcommittee Meeting," Tuesday, August 19, 1980, Inglewood, California.

7. Nuclear Safety Analysis Center and Institute of Nuclear Power Operations, "Analysis and Evaluation of Crystal River Unit 3 Incident," Joint NSAC/INPO Report NSAC-3/INPO-1, March 1980.

8. D. M. Rasmussen, G. R. Burdick and J. R. Wilson, Common Cause Failure Analysis Techniques: A Revision and Comparative Evaluation, EG&G Idaho, Inc., TREE 1349, September 1979.

9. H. P. Alesso, "Some Fundamental Aspects of Fault Tree and Digraph-Matrix Relationships for a Systems Interaction Evaluation Procedure," UCID-19131, May 1982.

10. A. Buslik, I. Papazoglou, and R. Bari, Brookhaven National Laboratory, "Review and Evaluation of Systems Interactions Methods," U.S. Nuclear Regulatory Commission Report NUREG/CR-1901, January 1981.

11. P. Cybulskis, et. a., Battelle Memorial Institute, "Review of Systems Interaction Methodologies," U.S. Nuclear Regulatory Commission Report NUREG/CR-1896, January 1981.

12. J. J. Lim, H. P. Alesso, T. R. Rice, R. K. McCord, J. E. Kelly, Lawrence Livermore National Laboratory, "Systems Interaction Evaluation Procedure for Application to Indian Point-3," U.S. Nuclear Regulatory Commission Report NUREG/CR-2050, May 1981.

13. J. J. Lim, R. K. McCord, T. R. Rice, J. E. Kelly, Lawrence Livermore National Laboratory, "Systems Interaction: State-of-the-Art Review and Methods Evaluation," U.S. Nuclear Regulatory Commission Report NUREG/CR-1859, January 1981.

14. H. E. Lambert, J. J. Lim, and F. M. Gilman, A Digraph Fault Tree Methodology for the Assessement of Material Control Systems, NUREG/CR-07777 UCRL 52710, May 21, 1979.

15. F. D. Coffman, "Initial Guidance for the Performance of Systems Interaction Reviews at Selected LWR's," U.S. Nuclear Regulatory Commission, October 1, 1981 (Draft).

16. Pacific Gas & Electric Co., "Description of Systems Interaction Program for Seismically-Induced Events, Diablo Canyon Units 1 and 2," Revision 2, Dockets 50-275/323, July 9, 1980.

17. U.S. Nuclear Regulatory Commission, Fault Tree Handbook, NUREG-0492, January 1981.

18. U.S. Nuclear Regulatory Commission, "Reactor Safety Study," WASH-1400 (NUREG-75/014), October 1975.

19. R. G. Busacker and T. L. Saaty, Finite Graphs and Networks, McGraw-Hill Book Co., N.Y., 1965.

20. I. Sacks, "Techniques For The Determination of Potential Adversary Success With Temporary," Lawrence Livermore National Laboratory, Report, MC 78-928D, October 17, 1978.

21. A. A. Parziale, et. al., "Structured Assessment Approach Version I," NUREG/CR 2301, UCID 18146, Lawrence Livermore National Laboratory, October 1979.

22. A. A. Parziale, I. Sacks, T. Rice, and S. Derby, "A Structural Assessment Analysis of Facility X," Vol. I, Lawrence Livermore National Laboratory, Internal Report, January 3, 1979.

23. P. A. Renard, "CLAMOR," Lawrence Livermore National Laboratory, MC-79-96, September 1979.

24. Power Authority of the State of New York, Systems Interaction Study, December 1981, Vol. I and II.

25. H. P. Alesso, "Review of PASNY Systems Interaction Study," Lawrence Livermore National Laboratory, UCID 19130, April 1982.

26. H. P. Alesso and H. J. Benson, "Fault Tree and Reliability Relationships For Analyzing Noncoherent Two-State Systems," Nuclear Engineering and Design, Vol, 56, pp. 309-320, 1980.

27. H. P. Alesso, "Some Algebraic Aspects of Decomposed Noncoherent Structure Functions," Lawrence Livermore National Laboratory, UCRL-85100, April 1979.

28. Tremblay, Sorenson, An Introduction to Data Structures with Applications, p. 389, McGraw-Hill, 1976.

Acknowledgement

It is a pleasure to acknowledge the helpful suggestions, critical comments, and tireless efforts of Tracy Massey in the preparation of this manuscript.