SRC TR 86-55

APPLICATION OF GOAL TREE-SUCCESS TREE MODEL AS THE KNOWLEDGE-BASE OF OPERATOR ADVISORY SYSTEMS

by

I. S. Kim & Modarres

APPLICATION OF GOAL TREE -SUCCESS TREE MODEL AS THE KNOWLEDGE-BASE OF OPERATOR ADVISORY SYSTEMS

I. S. KIM
M. MODARRES*

Submitted for Publication to Nuclear Engineering
and Design Journal, October 1986

Department of Chemical and Nuclear Engineering
University of Maryland, College Park, MD 20742

*To whom correspondence should be sent

ABSTRACT

The most important portion of an expert system development is the articulation of knowledge by the expert and its satisfactory formulation in a suitable knowledge representation scheme for mechanization by a computer. A "deep knowledge" approach called Goal Tree-Success Tree model is devised to represent complex dynamic domain knowledge. This approach can hierarchically model the underlying principles of a given process domain (for example nuclear power plant operations domain). The Goal Tree-Success Tree can then be used to represent the knowledge-base and provide means of selecting an efficient search routine in the inference engine of an expert system.

A prototype expert system has been developed to demonstrate the method. This expert system models the operation of a typical feedwater system used in the pressurized water reactors. The expert system is modeled for real-time operations if an interface between plant parameters and the expert system is established. The real-time operation provides an ability to quickly remedy minor-disturbances that can quickly lead to system malfunction or trip.

A description of both the Goal Tree-Success Tree Model and the prototype expert system is presented.

SUMMARY

Complex processes and their control are cumbersome tasks. Although plant operators are well trained in standard operating procedures, they may have difficulty in handling complex or rare operating conditions. In addition, there are often wide differences between operators' ability to achieve a high or optimum level of process performance. That is, better operators operate the process closer to the optimum level of performance, and thus increase productivity and performance. Handling these operating conditions during the plant operation requires a fundamental understanding of the physical principles based on which plants operate. Recent expert systems [1-5] have shown the feasibility of including structural models of the problem domain in the knowledge base. Application of this type of knowledge representation allows real-time direct access to plant measurements of the plant control data base for improved and more consistent plant operation in spite of personnel turnover, lapses in human attention, and the like.

Two general approaches have been applied to develop expert systems to aid operators. One is model-based and the other is model-free. The model-free approach acquires the knowledge from experts directly, usually in the form of (if-then) production rules. This type of approach utilizes empirical associations between evidence and conclusions, but does not depend on a 'deep' (functional) understanding

of the domain itself. The knowledge captured in these expert systems is a shortcut through, or an experiential compilation of the underlying principles in the problem domain [6]. This has been referred to as the 'shallow' knowledge approach. This approach is typical in medical diagnosis programs, where underlying disease mechanisms are unknown or difficult to describe (for example, MYCIN [7] is based on this approach). The shallow approach has been regarded as the standard approach in building expert systems.

For many reasons, the shallow approach is not ideal for complex plant modeling in the form of expert systems. For example, the knowledge needed to solve a plant upset condition is broad and ill-defined. The required knowledge may include the plant layout, physical and chemical properties of the fluids, design specifications, knowledge of past and current operating conditions, and interpretation of process measurements. However, there is a structured theoretical and conceptual basis which relates all aspects of plant operation together. During plant operation a large fraction of the knowledge is structured, and physical models based on principles of mechanics, dynamics, thermodynamics, heat, mass and momentum transfer, kinetics and process chemistry, along with operator interactions, can be hierarchically modeled.

Since the 'deep knowledge' approach utilizes a model of the problem domain, it would be more appropriate for process control, plant operation and diagnosis representation. Examples of well known expert systems that utilize deep knowledge include ABEL [8] and IDM [9]. By acquiring the underlying principles of a domain explicitly, the model based approach need not anticipate every possible situation in the domain, which enables the expert system to handle a wider range of problem types and larger problem domains such as nuclear power plant operation.

In this paper we have introduced a new type of deep knowledge representation model called a Goal Tree-Success Tree model (herein called GTST model) in developing expert systems for plant monitoring, controlling and alarm handling activities. This model can completely and rigorously describe a process plant and its operations. It incorporates a structured approach that shows how a specific objective in a plant is achieved. This is done by defining the objective and partitioning it into a series of related sub-objectives or goals; these goals are then broken down into subgoals. The partitioning of goals and subgoals continues until their description can not be made without referring to plant hardware. At this point logical model of the hardware should be represented in the form of success trees.

In this paper we have shown how a GTST model is used to represent an expert system's knowledge base. Also improved search process of the inference engine by using this model is discussed. An expert system for a typical feedwater system operation of a pressurized water reactor is developed on an IBM-AT, by using the micro-PROLOG language. [10] The system showed that the GTST model can be effectively used in developing the expert system knowledge base and improves search strategies.

The expert system is developed such that it can be used to operate in a real-time environment, in order to provide quick response to operator's needs.

2. Development of Goal Tree-Success Tree Model

The concept of GTST model is not new. This approach has been successfully used as a tool for a variety of engineering applications [11-15].

In complex process plants human operators tend to think in terms of achieving plant goals by successfully operating relevant plant hardware that support the plant goals. For instance, the operators in a nuclear power plant have indications of current reactor power level and combination of hardware that are currently operating to achieve such a power level. In this case maintaining power at a given power level is a goal and combination of supporting hardware represent a success path. As a process upset condition occurs in the plant, a goal can be challenged; for example by losing success path hardware. In this case, the operator attempts to locate and maintain the challenged goal and its corresponding success path(s).

A close view of the above facts recognizes the complexity and dynamic nature of the challenges and operator intervention.

The GTST can be efficiently utilized as a deep-knowledge representation scheme in an expert system to provide timely access to plant knowledge and quick advice. In the remainder of this section the method of constructing GTST models are discussed.

Fig. 1 shows the structure of a GTST model. Since the GTST model is a top-down structure, first a description of the top part (i.e., the goal tree) is provided and then the description of the lower part (i.e., the success tree) will follow:

a) Goal Tree Development - The first step in developing a goal tree model involves definition of the top plant goal or objective. This top goal must be explicitly defined in terms which make it a single unambiguous statement. It is from this definition that the analyst will identify and relate all the different plant goals and subgoals which must be achieved to attain the overall objective.

The goal tree is constructed vertically downward from the objective in levels, wherein the analyst subsequently decomposes each identified goal into a necessary and sufficient set of dependent subgoals. As the vertical detailed development of the tree increases, it is necessary that tests be applied to ensure its accuracy and completeness, and the proper hierarchy between goals and subgoals be rigorously maintained.

The tests which are to be applied at every level of the tree to ensure adequacy are:

1) Upon looking upward from any subgoals towards

the top objective, it is possible to define explicitly <u>why</u> the specific goal or subgoal must be satisfied.

2) Upon looking downward from any goal towards the bottom of the tree, it is possible to define explicitly <u>how</u> the specific goal or subgoal is satisfied.

Failure of the tree to pass either of these tests at any level implies that:

1) there is a lack of completeness and that intermediate subgoals have been omitted, or
2) the hierarchy of goals or their interdependencies have been neither rigorously nor completely defined.

The process of downward development of the tree continues until the analyst can not further compare the tree without referring to a hardware. As soon as hardware or a process condition is explicitly mentioned, the tree can be viewed as one which describes success paths, not goals.

While a plant is operating, it passes through various plant states. Each plant state requires certain functions or goals to be achieved. Therefore, the goal tree can be viewed as a dynamic model (i.e., the element of time can also be considered in the model) in that as the plant enters a new state, only the goals applicable to this state can be retained for expert system considerations. Moreover, the level of the goals or functions in the tree does not indicate their importance. It is the degree of the effect on the objective that determines their importance. In fact, it is possible to visualize the entire tree as a 'mobile' which is held in space with each of the elements connected by elastic whose length is determined by the inverse of its importance. As the plant moves through its spectrum of possible states, applicable goals would move relative to each other, so that those which have the higher importance at any point in time will be closer to the objective, conceptually.

b) Success Tree - The success tree part of the GTST model is a logical model of a hardware or plant system from which success paths can be determined. A success path shows various components whose proper operation guarantee the successful operation of the system. A system may have more than one success path, in which case these are alternate ways of achieving system success (alternate success paths are designed for achieving high redundancy).

It is important that the boundary between goals and success tree (or success paths) be recognized in the composite GTST model. Within the goal tree, all goals are connected by logical AND gates, whereas the success paths are connected to the goals they serve by logical OR gates. The fact that all connectors in the goal tree are logical ANDs can be used to establish a convention which provides the analyst with a means for easy distinction between the goal tree and the hardware success paths. Therefore, by convention, the goal tree is drawn with no AND gates explicitly displayed. If a gate is shown, success tree or success paths are being described. Since all components of a success path are required to function for the success of that path, these components must be connected by an AND gate. However, AND gates in the success tree should be explicitly shown. Fig. 1 illustrates the relationship of objective, goals, subgoals, and success path, and hardware.

3. Use of The GTST Model As A Dynamic Knowledge Base

The GTST model can be effectively used as a suitable basis for developing a dynamic knowledge base of a real-time expert system for operator assistance. In this case, a model of the plant in form of the GTST model can be used to provide the ability for reasoning by abstraction to the plant operators. The success tree part of the model provides all the success paths available for operation for a given plant state to the operator, and the goal tree part shows how and why these success paths support the plant objective.

A logic or symbolic programming language such as PROLOG or LISP can be easily used to encode the logic GTST model. The dynamic knowledge base constructed by use of such a language can be utilized along with an efficient inference engine to infer the status of plant goals and their corresponding optimal success path(s). For demonstration purposes the language micro-PROLOG was selected, since its built-in inference engine (i.e., backward chaining and depth-first search [16-17]) can be effectively used for the reasoning purpose. The GTST structure, however, provides an ability to improve the built-in search strategy if necessary. The method of improvement may be different for various GTST models.

The success paths and the goals of a GTST model can be conveniently represented by relational sentences in micro-PROLOG. The dynamic knowledge base also consists of rules and facts. The rules are used for example to verify a lost hardware and to verify availability of standby or non-operating hardware. There are two kinds of facts in the knowledge base, one being time-independent facts and the other being time-varying facts. The time-independent facts represent relations of the physical structure of the plant or the inherent characteristic of the process that does not vary with time, while the time-varying facts represent the current plant hardware states .

In developing a real-time expert system an interface between plant data and the expert system should be provided in order to directly feed time-varying process data from the process control or monitoring system into the dynamic knowledge base. From time to time additional time-varying data may also be supplemented by user/operator, if they are not available through the direct communication links.

4. CFWAVA Expert System for Operator Assistance

A pilot expert system called CFWAVA (Expert System for Condensate and Feedwater System Availability Operation) has been developed based on the GTST model. The CFWAVA expert system is an expert-advisor that can assist plant operators in coping with process upsets in the Condensate and Feedwater System (CFWS) of a pressurized water reactor. The partial loss of the CFWS is the most significant contributor to the plant outages [18]. However, the partial loss of this system does not completely disable the heat removal capability of the CFWS provided that quick remedial actions be provided. For example, a survey of operating nuclear power plants [19] has shown that out of 1161 outages there were 274 which appeared preventable by more rapid actions. The system responsible for the largest number of outages (235 out of 1161) was CFWS. It was also indicated that 131 of these outages were preventable by a more rapid operator action. Reduced number of CFWS outages will also improve plant safety by reducing the frequency of loss of feedwater system transients.

Fig. 2 represents a simplified schematic of a typical CFWS which is used in this pilot expert system. The system shows six condensers, three condensate pumps, three condensate booster pumps and two main feed pumps (for simplification all other auxiliary hardware such as valves, preheaters are eliminated). The CFWS has several discrete success levels such that each level corresponds to one of the reactor power levels. By the timely and appropriate response of the operator, the reactor power can be reduced to a level corresponding to the heat removal capability of the CFWS at a given time. The success criteria defining the individual CFWS component capabilities to the overall plant states (here discrete power levels) are shown in Table 1.

Table 1

	Success Criteria For CFWS Plant Power Levels			
	100%	75%	50%	30%
Condenser Success	6/6	5/6	4/6	3/6
Cond. Pump Success	3/3	2/3	1/3	1/3
Booster Pump Success	2/3	2/3	1/3	1/3
Main Feed Pump Success	2/2	2/2	1/2	1/2
	State 1	State 2	State 3	State 4

If the plant power cannot be maintained at any of the four discrete success levels of the CFWS, then the operator should be quickly advised to shutdown the plant.

The CFWAVA code is capable of detecting an upset condition for a given power level, success path, and the time-varying process data, and provide the operator with the system state diagnosis result, the optimal achievable power level, corresponding optimal success path(s), and the required operator actions to achieve the new power level. The CFWAVA's real-time data interface is currently being developed. However, the expert system itself is modeled such that monitored plant parameters can be handled in a real-time basis.

In the rest of this section a discussion of the corresponding GTST model, and a representation of this model as the knowledge base of the CFWAVA code is presented.

4.1 Development of Dynamic Goal Tree for CFWAVA

The GTST model (Fig. 3) is developed with the top objective of "Cycle Equivalent Availability Maximized". This objective is then broken down into its relevant goals. The rules for constructing GTST models as discussed earlier in this paper are strictly applied to maintain hierarchy and completeness of the tree.

Each of the goals is then decomposed into its lower level goals. For example, the goal "Availability of Energy Sinks Maximized" is broken down into "Equivalent Availability of Electric Switchyard Maximized" and "Availability of Thermal Condensing is Maximized". The former goal is achieved by the operation of hardware such as transformers, electric lines, switchyard,..., etc. and the latter is achieved by condensers, ..., etc.

Similarly the goal "Equivalent Availability of the Energy Transport Maximized" is broken down into three subgoals. One of these subgoals is "Equivalent Availability of Main Feed Water System Maximized". This goal can be achieved by maintaining a high level of reliability (e.g., by minimizing the failure rate of hardware supporting the goal) or by maximizing the goal's operability (e.g., by maximizing availability or readiness of supporting hardware).

In this very broad GTST model we will only detail the success paths of the goal "Operability of Main Feedwater System Maximized". For a comprehensive representation of the GTST model's top objective, all systems supporting the lowest level goals of this GTST model should be developed. However, since this expert system is developed for demonstration purpose, we will only detail the important goal of maximizing the feedwater operability.

In the success tree portion of the GTST model, the success paths of the goal "Operability of Main Feedwater System Maximized", the set of hardware required for operation in each defined power level (Table 1) is shown. Because of the complexity of the success tree, only success paths of one of the six condensers (CD 11A), and one of the three condesate pumps (CDP11) and one of the three condensate booster pumps (CDBP11) and one of the two steam generator feed pumps (CDFP11) are shown. The rest of these hardware have identical success path configurations which are not explicitely shown on the GTST model in Fig. 3 but are modeled in the expert system.

Let us assume that the plant has been operating with the success path of ((CD11A CD11B CD12A CD12B CD13A) (CPP11 CDP13) (CDBP11 CDBP12) (SGFP11 SGFP12)). This indicates that according to the GTST model the current power level is 75%. Thus the operator goal would be to maintain the power at this level or the next highest possible power level.

Suppose that at a given time, a process upset causes the discharge pressure of the condensate pump 11 to become out of the allowed set range which in turn causes the trip of this pump according to the control logic. Then the current success path would no longer be viable (according to the logic demonstrated in the GTST model).

If the condensate pump 12 which has not been operating is available for operation and no further operational restrictions (such as low condensate inventory in the main feed system exists), then the condensate pump 12 will be able to compensate for pump 11. Therefore, the right branch of the subtree for condensate pump 12 will be examined to find out if the condensate pump is available for operation at this time. If pump 12 is not available for operation, the next possible power level will be examined for potential usage. The result will quickly be provided to the operator.

4.2 Knowledge Base of CFWAVA

The knowledge base of CFWAVA is constructed based on the structured knowledge formulated in the GTST model. Since the feedwater system can successfully operate in different power levels, a state concept is devised for representing power levels. A state is defined as a list such as (A B C D), in which A, B, C, and D indicate the number of condensers, condensate pumps, booster pumps, and feed pumps respectively. There are four possible states as follows:

1) previous state: A state list showing the system configuration prior to a disturbance

2) reduced state: Similar to "previous state" with the disturbed item(s) eliminated from the list

3) feasible state: Similar to "reduced state" with the available item(s) added to the list to replace the disturbed item(s)

4) optimal state: Similar to "feasible state" with unnecessary item(s) removed so as to match the corresponding power level

The success paths corresponding to various power levels are also represented by a list consisting of four sublists; each sublist indicating condensers, condensate pumps, booster pumps, and feed pumps. For example, the following list shows the discussed sublists:

$((CD_1,CD_2,\ldots)\ (CDP_1,CDP_2,\ldots)\ (CDBP_1,CDBP_2\ldots)\ (SGFP_1,SGFP_2,\ldots))$.

The number of item(s) in each sublist of success path corresponds to the respective item in the state list. For example, for the state list [previous state (4 1 1 1)], one possible success path would be:

previous-success-path ((CD11A CD12A CD12B CD13A) (CDP11) (CDBP12) (SGFP11)).

If the expert system detects a disturbance or failure in say CD12A, then the reduced success path would be determined as follows:

reduced-success-path ((CD11A CD12B CD13A) (CDP11)
(CDBP12) (SGFP11)).

Further, assume that a check on the readiness and availability of other condensers indicates that CD11B is available for operation. Therefore, the feasible success path can be determined. In this case:

feasible-success-path ((CD11A CD11B CD12B CD13A)
(CDP11) (CDBP12) (SGFP11)).

If a comparison with the success criteria for operation configuration (See Table 1) indicates that the current power level can be maintained with the feasible success path then this path will be selected as an optimal success path. Otherwise, the expert system would determine the next possible optimal success path (perhaps relevant to a lower power level). Suppose that the expert system indicates that CD11B can be replaced by CD12A within the constraints set forth by the defined success criteria, then the optimal success path is:

optimal-success-path ((CD11A CD11B CD12B CD13A)
(CDP11) (CDBP12) (SGFP11)).

Relationships between goals can also be represented by micro-PROLOG sentences such as:

```
((how (maximize-energy-source-availability) (maximize-
    cycle-equivalent-availability)))
((how (maximize-equivalent-availability-of-energy-conversion)
    (maximize-cycle-equivalent-availability)))
```

In order to determine the operating conditions of items (either working within acceptable limits or not), rules such as the one shown below can be used:

```
((diag _CD11A) (P "Enter TS1-sensor-value in degree-Fahrenheit:")
(R _TS1) (P "Enter TS2-sensor-value in degree-Fahrenheit:")
(R _TS2) (EQ _TD1 #(_TS2 - _TS1)) (OR ((lesseq _TD1 12)
(/)) ((delete ((lost _oL1))) (APPEND (_oL1 (CD11A) _nL1))
(add (lost _nL1))))).
```

This rule diagnoses the operating condition of CD11A. If the temperature difference between the temperature sensor TS1 and TS2 is less than or equal to 12 degrees Fahrenheit, then CD11A is considered to be successfully operating. Otherwise, the unit is considered to be in failed condition.

The availability of an item can be checked by rules such as the one below:

```
((check-availability-CDP _P) (P "Check tagging-status of
_P [TAG/NO]:")(R _tag)(P "Check motor-ammeter for _p (4KV-Bus12)
[OK/NO]:")(R _ma)(P "Check cooling-oil-pressure > 23psig
[OK/NO]:") (R _cop) (OR ((EQ _tag NO) (EQ _ma OK)
(EQ _cop OK) (branch-OP-CDP _p) (/))((branch-NOP-CDP _p))))).
```

This rule acquires directly the tagging status, the readiness of the motor ammeter, and the appropriateness of the cooling oil pressure of the condensate pump. If the pump is not tagged out and the support systems (e.g., electric motor power and the cooling oil here) are ready for operation, then the hardware is regarded as being available for operation.

The backward chaining (goal-driven) inference built in PROLOG was used to find the plant state, but improvement in the search process was achieved by heuristic procedures. For instance, conditions such as 1) insufficient condensate inventory, or 2) complete loss of any single condenser shell is checked just after the completion of system state diagnosis. This very quick check reduces the search time significantly since in these cases the plant should be shutdown. This is because time-consuming processes such as state and success state determination, operability and availability checks can be avoided in such a case.

Because of dynamic effects of feedwater systems, when a failure occurs, a quick response will minimize progression of the occurred transient to other hardware. A real-time interface with this expert system is currently being developed that automatically provides the operating and availability conditions. Also, other auxiliary components such as valves and heaters are also being added. With such a real-time interface completed, the response actions can be performed quickly before the transient effect can progress to other hardware.

The user can ask CFWAVA "why" a recommendation is made. Presently, explanations are provided to the user only in cases where plant shutdown is recommended. This kind of explanation capability is important in this context since the explanation facility helps to resolve the conceptual differences between the user and the expert system when such a conflict exists.

Fig. 4 shows a typical CFWAVA session.

5. Conclusions

The GTST model is used to develop an expert system for operator assistance during a process upset condition. It is observed that the GTST provides an ability to represent deep knowledge structure in the knowledge base of an expert system. Furthermore, the well-organized knowledge structure such as the GTST model can help the knowledge engineer find better search routine and thereby assist in developing a more efficient real-time expert system.

Based on this approach the expert system CFWAVA is developed. This system has demonstrated the efficiency of the GTST model. The quick advice provided from this expert system can significantly improve operational ability and safety of plants when quick actions are necessary to maintain important plant goals.

ACKNOWLEDGEMENT

The authors wish to acknowledge the support of the University of Maryland's Systems Research Center provided through NSF Grant.

REFERENCES

1. N.S. Sridharan, "AIMDS user manual," Technical Report CBM-TR-89, New Brunswick, N.Y. (June 1978), Computer Science Dept., Rutgers University.

2. T. Bylander, S. Mittal & B. Chandrasekaran, "CSRL:A Language for Expert Systems for Diagnosis," Proceedings IJCAI-83(1983).

3. J. Batali & A. Hartheimer, "The Design Procedure Language Manual", Report AIM-598, AI Laboratory, MIT, Cambridge, Mass. (September 1980).

4. I.P. Goldstein & R.B. Roberts, "Using Frames in Scheduling", Artificial Intelligence: An MIT Perspective, MIT Press (1979).

5. J.M. Wright & M.S. Fox, SRL/1.5 user manual, Robotics Institute, Carnegie-Mellon University, Pittsburgh, Pa., December 1983.

6. P.K. Fink, "Control and Integration of Diverse Knowledge in a Diagnostic Expert System", Proc., 9th IJCAI (1985).

7. B. Buchanan, "New Research on Expert Systems", Machine Intelligence 10 (1982).

8. R.S. Patil, P. Szolovits & W.B. Schwartz, "Casual Understanding of Patient Illness in Medical Diagnosis", Proc. 7th IJCAI (1981).

9. M.A. Kramer & B.L. Palowitch, Jr., "Expert System and Knowledge-Based Approaches to Process Malfunctions Diagnosis", AIChE National Meeting (November 1985).

10. K.L. Clark and F.G. McCabe, micro-PROLOG: Programming in logic (Prentice Hall International, London, 1984).

11. R.N. Hunt and M. Modarres, Integrated economic risk management in a nuclear power plant, Annual Meeting of the Society for Risk Analysis, Knoxville, TN, October 1984.

12. M.L. Roush, M. Modarres and R.N. Hunt, "Application of goal trees to evaluation of the impact of information upon plant availability, Proceedings of the ANS/ENS topical meeting on probabilistic safety methods and applications, San Francisco, CA, February 1985.

13. M. Modarres, M.L. Roush and R.N. Hunt, Application of goal trees in reliability allocation for systems and components of nuclear power plants, Proccedings of the twelfth international reliability availability maintainability conference for the electric power industry, Baltimore, MD, April 1985.

14. M. Modarres, M. L. Roush and R.N. Hunt, Application of goal trees for nuclear power plant hardware protection, Proceedings of the eight international conference on structural mechanics in reactor technology, Brussels, Belgium, August 1985.

15. M. Modarres and T. Cadman, A method of alarm system analysis in process plants with the aid of an expert computer system, Computers and Chemical Engineering, 1986, to appear.

16. N.J. Nilsson, Principles of artificial intelligence, Tioga Pub. Co., California, 1980.

17. A. Barr and E. A. Feigenbaum, The handbook of artificial intelligence, vol. 1,Heuris Tech Press, Stanford, California, 1981.

18. F.P. Lees, Process computer alarm and disturbance analysis: Review of the state of the art, Computers and Chemical Engineering, Vol. 7, No.6, pp.669-694, 1983.

19. C.H. Meijer and B. Frogner, On-line power plant alarm and disturbance analysis system, Final report, Electric Power Res. Inst., Palo Alto, California, Rep. EPRI-1379 (1980).

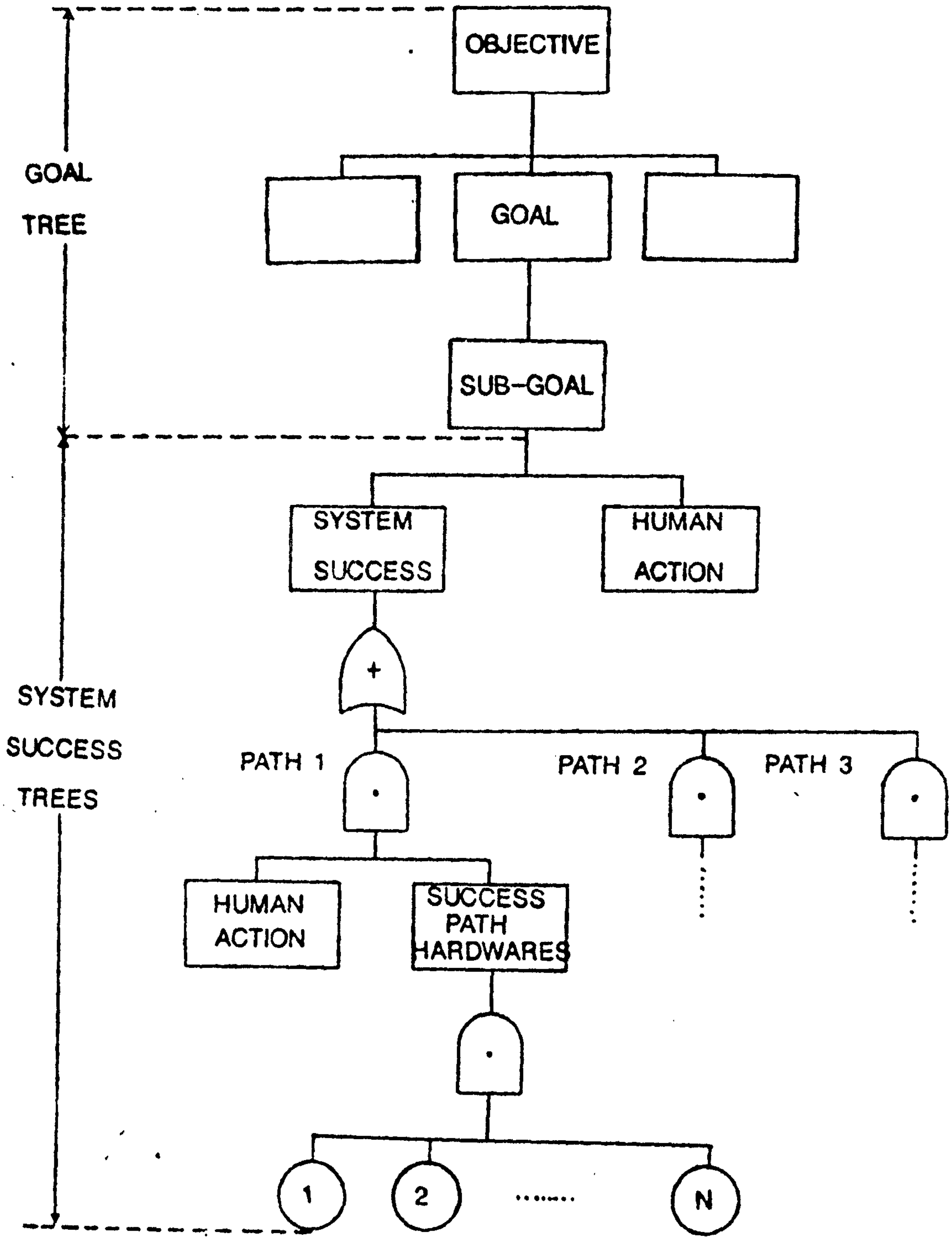


Fig. 1 A Typical Structure of Goal Tree - success Tree

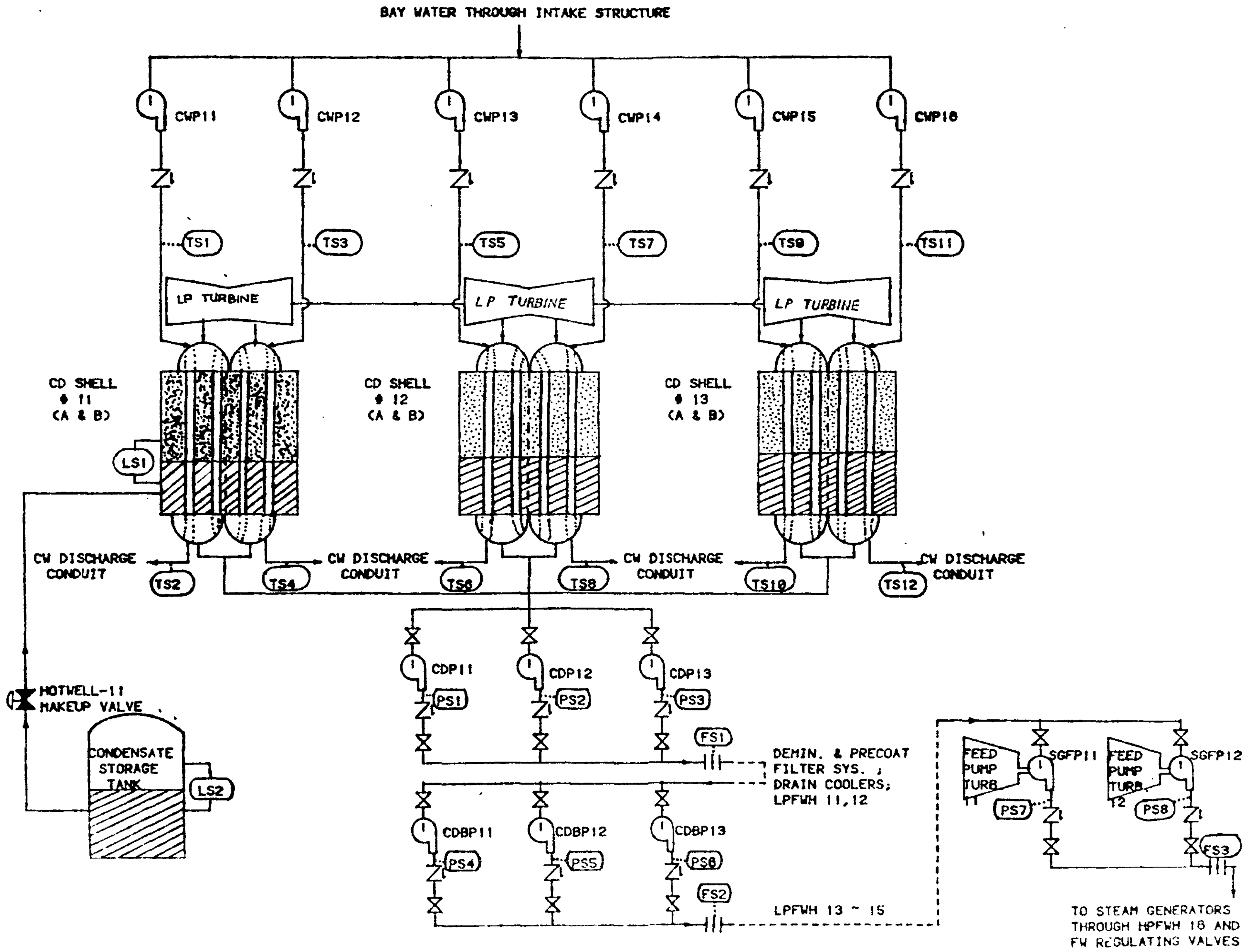


Fig. 2 Symplified Schematic of Condensate and Feed Water System

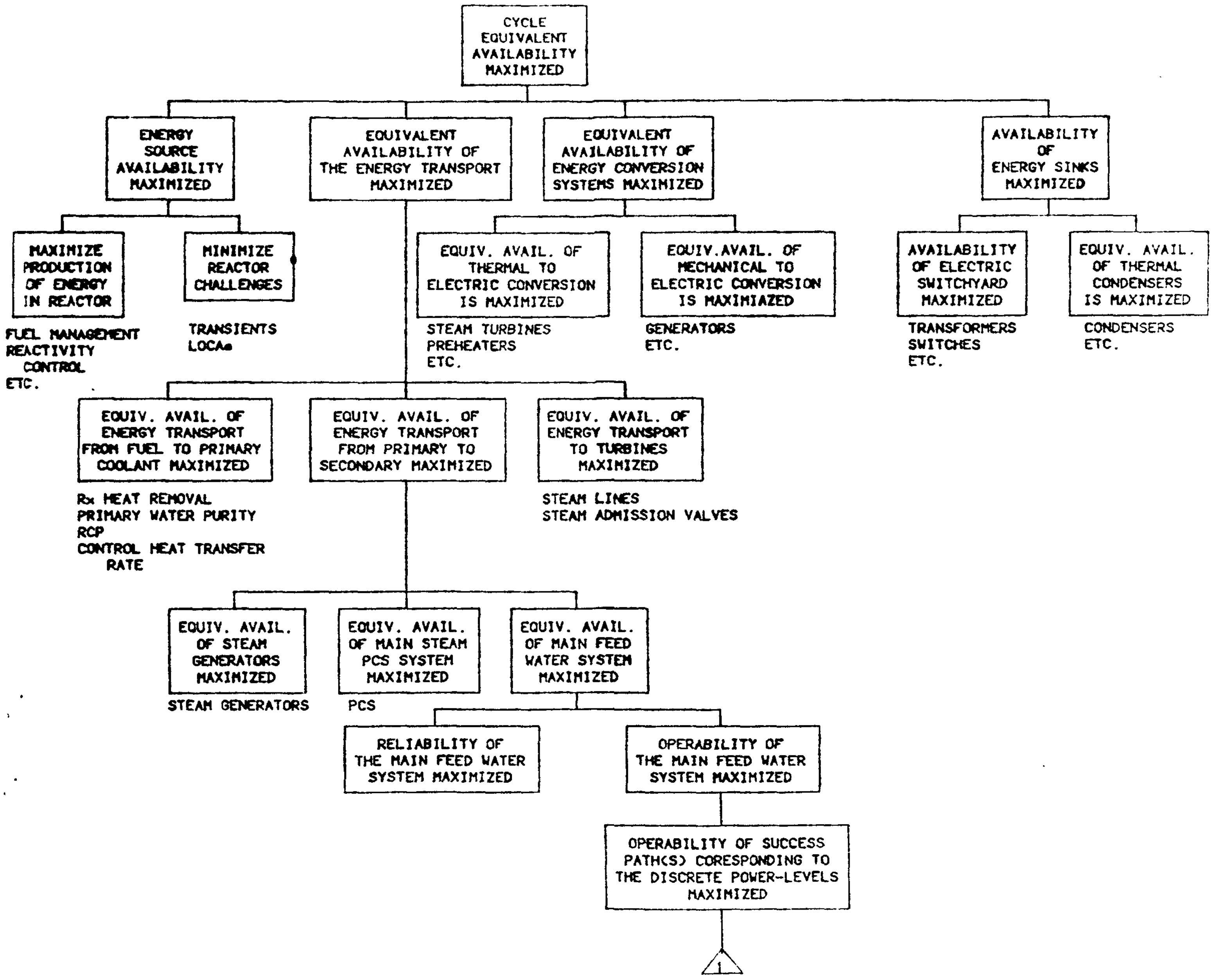


Fig. 3 Goal Tree - Success Tree Developed for Condensate and Feed Water System Availability

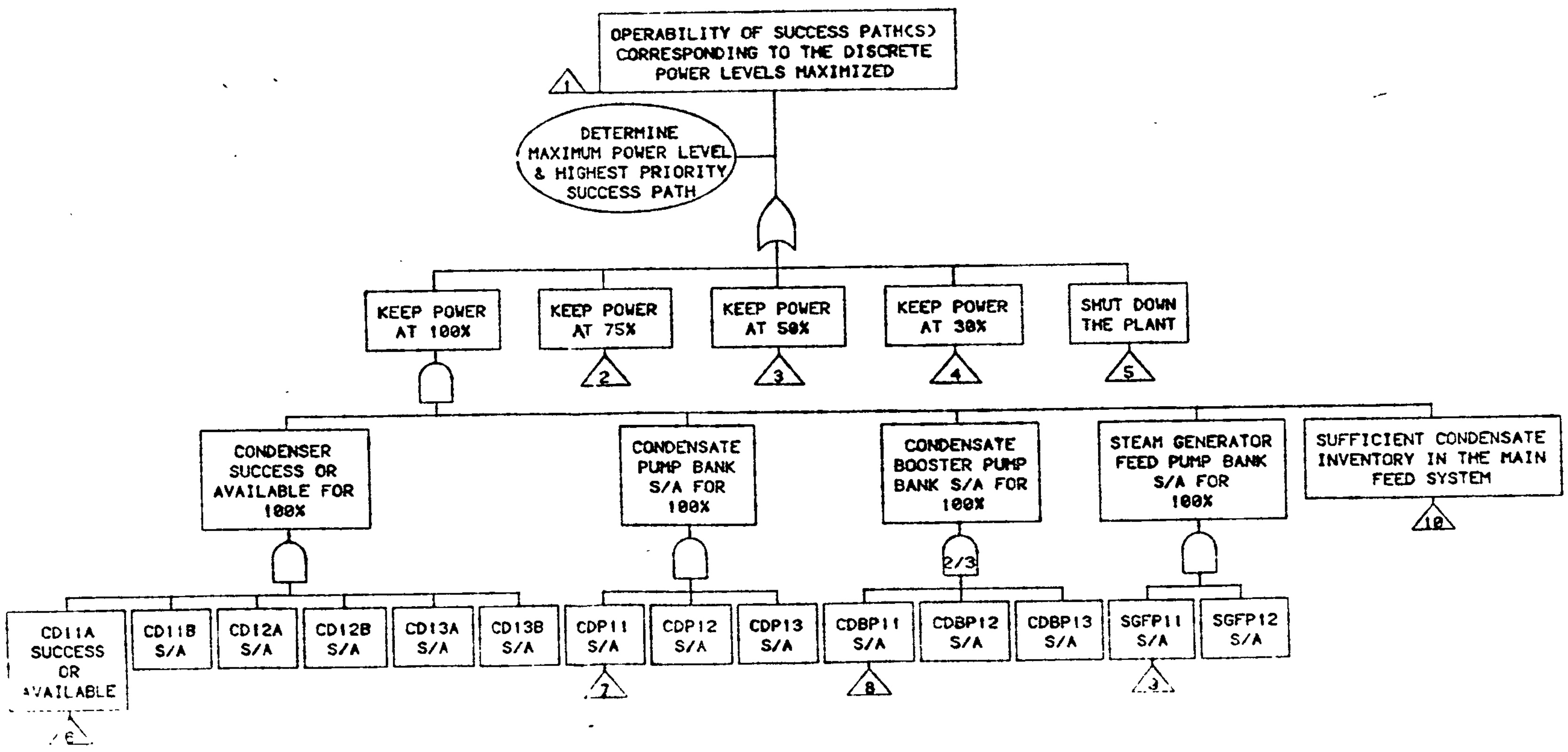
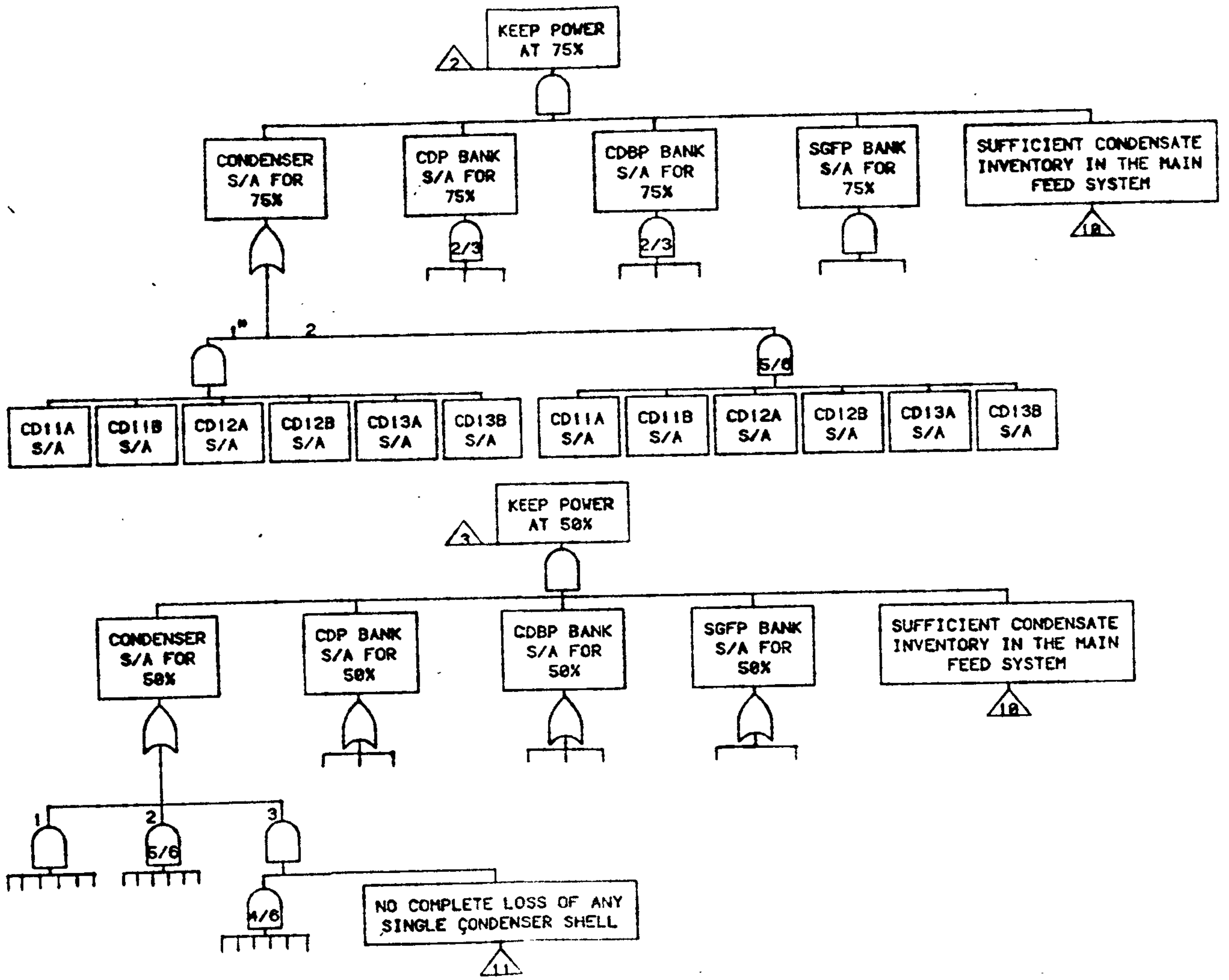


Fig. 3 (continued)



* LOWER NUMBER INDICATES HIGHER PRIORITY PATH

Fig.3 (continued)

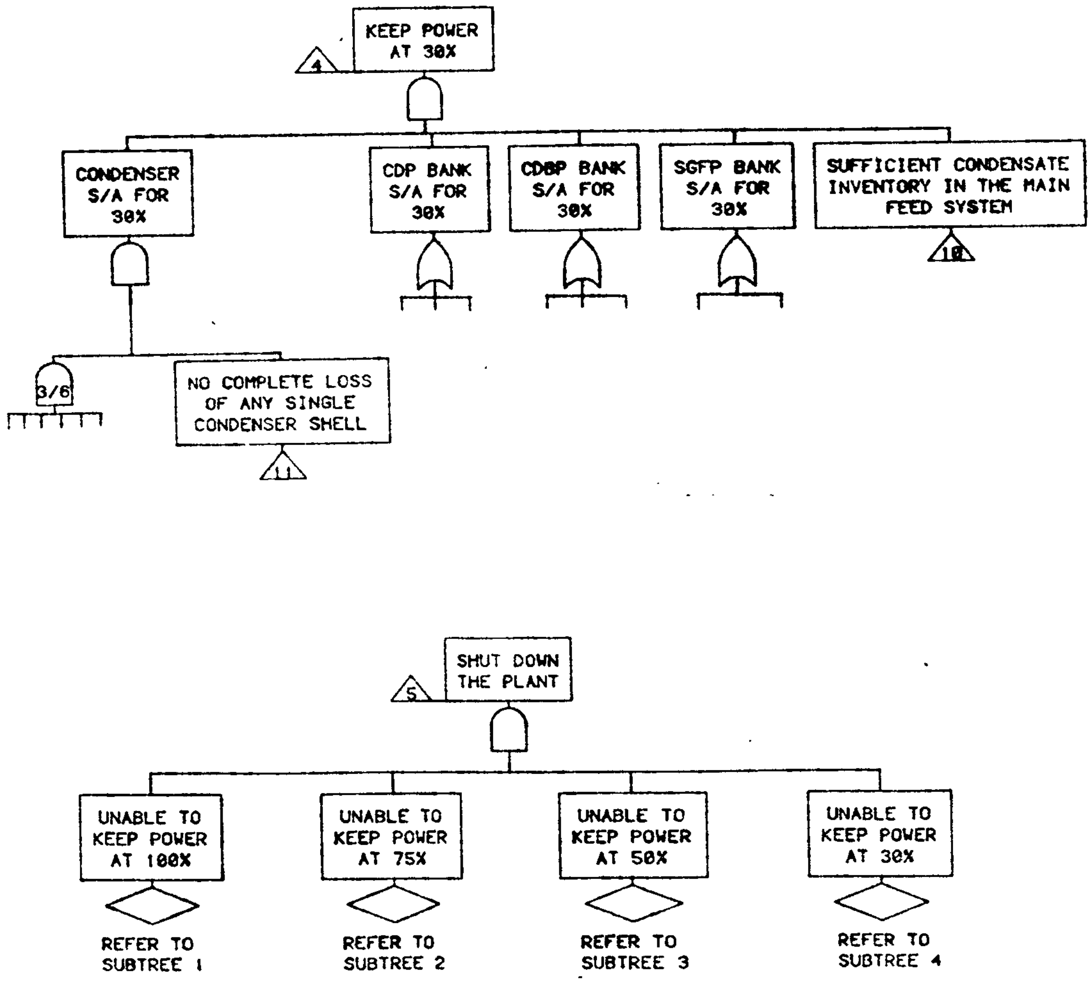


Fig.3 (continued)

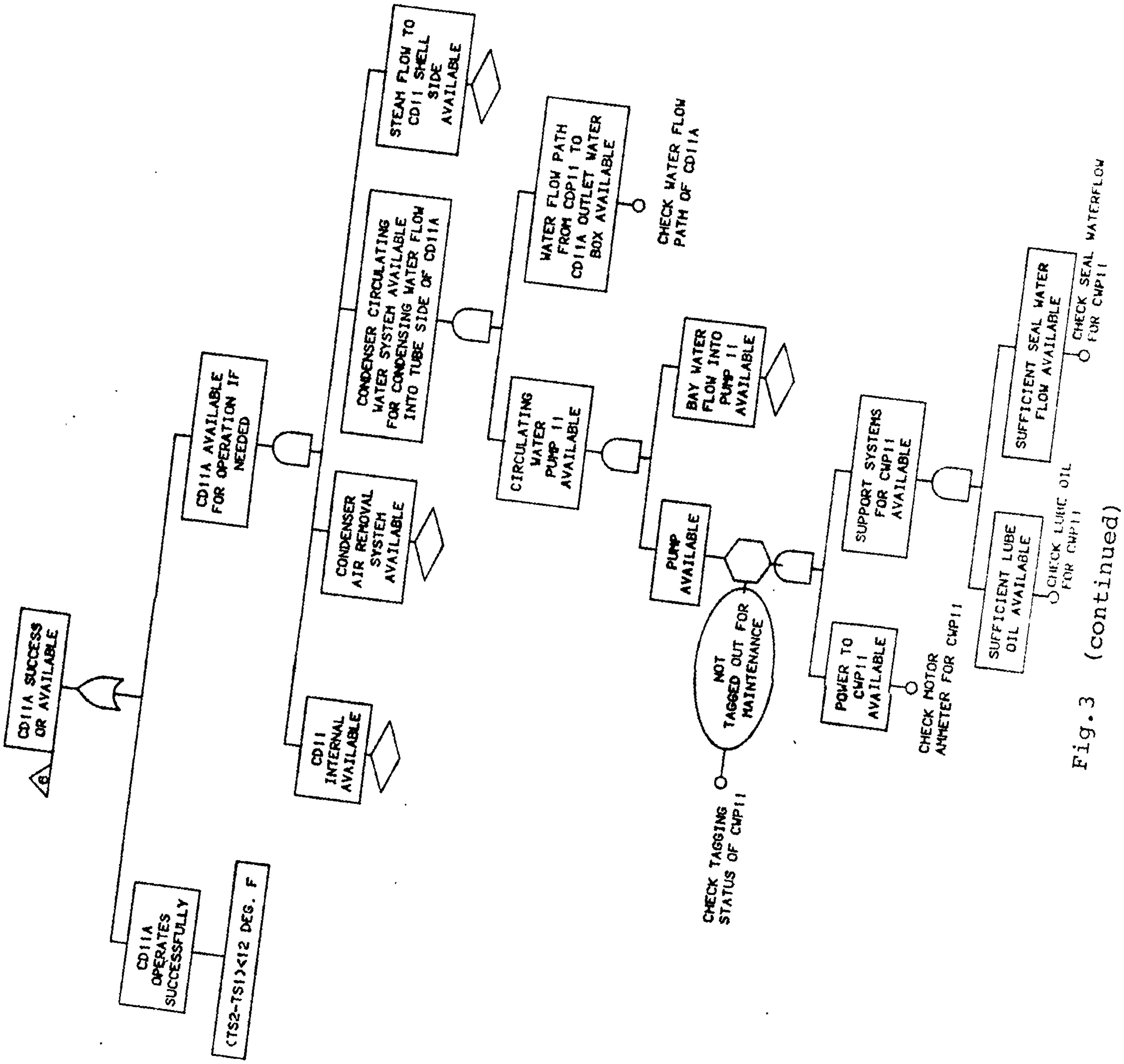


Fig.3 (continued)

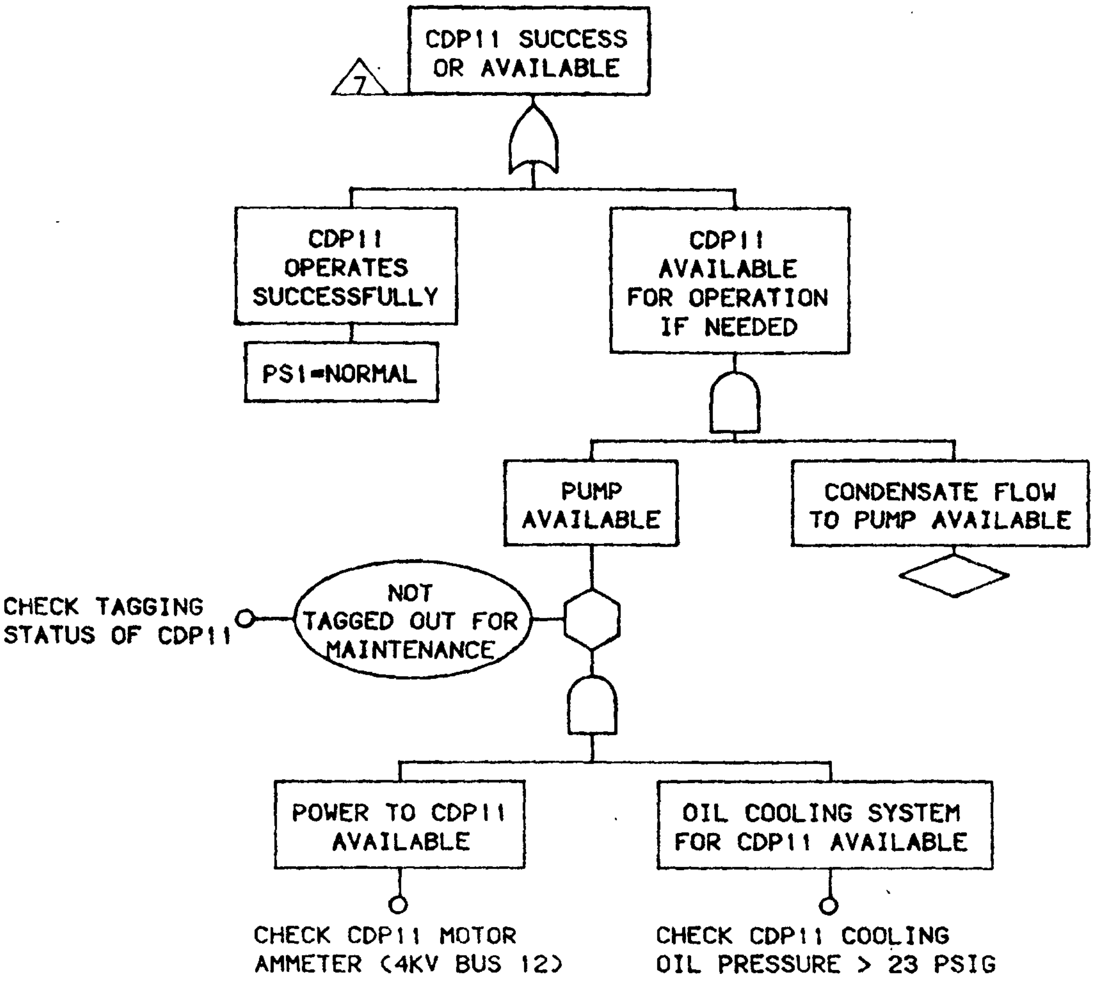


Fig.3 (continued)

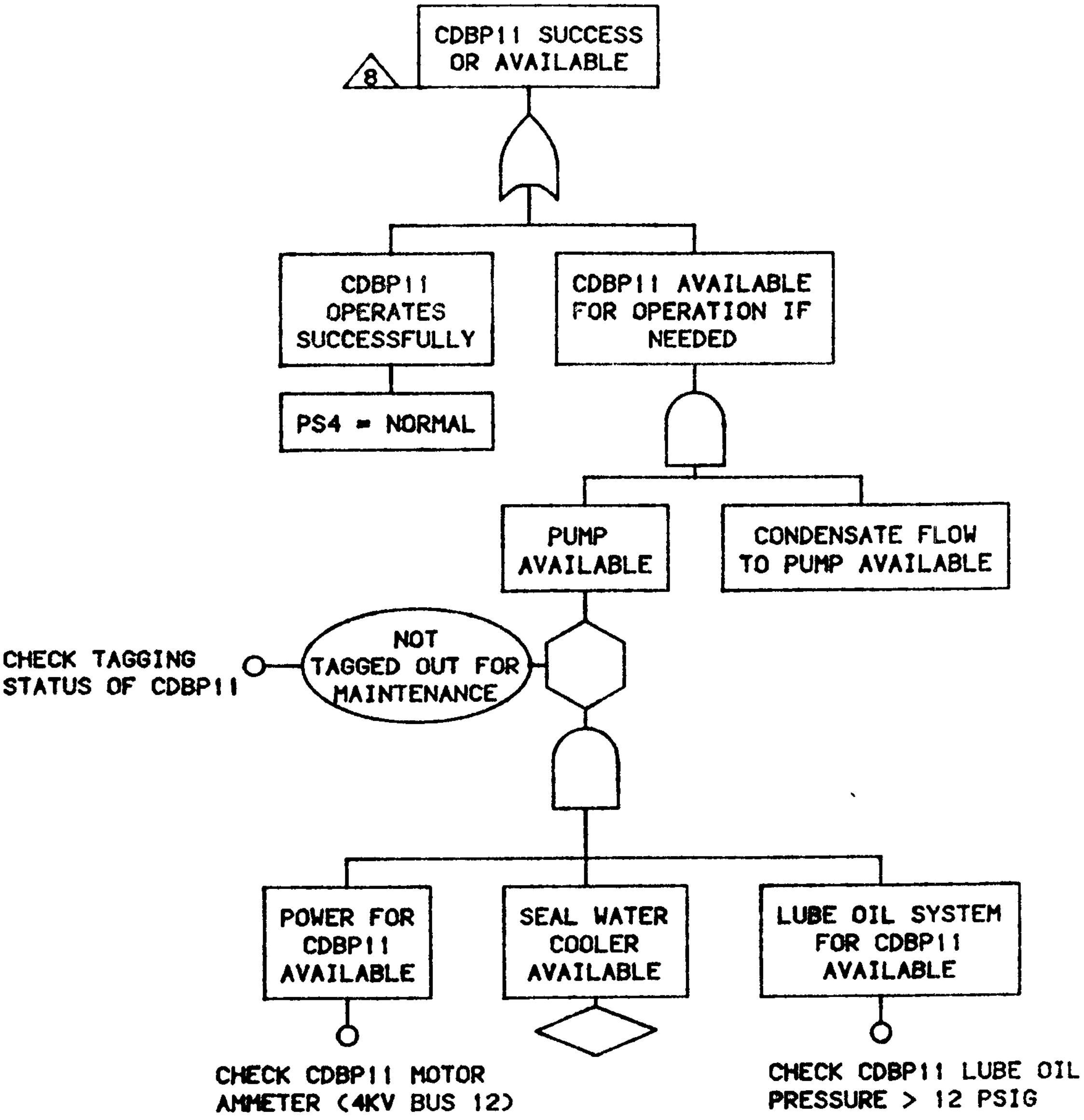


Fig. 3 (continued)

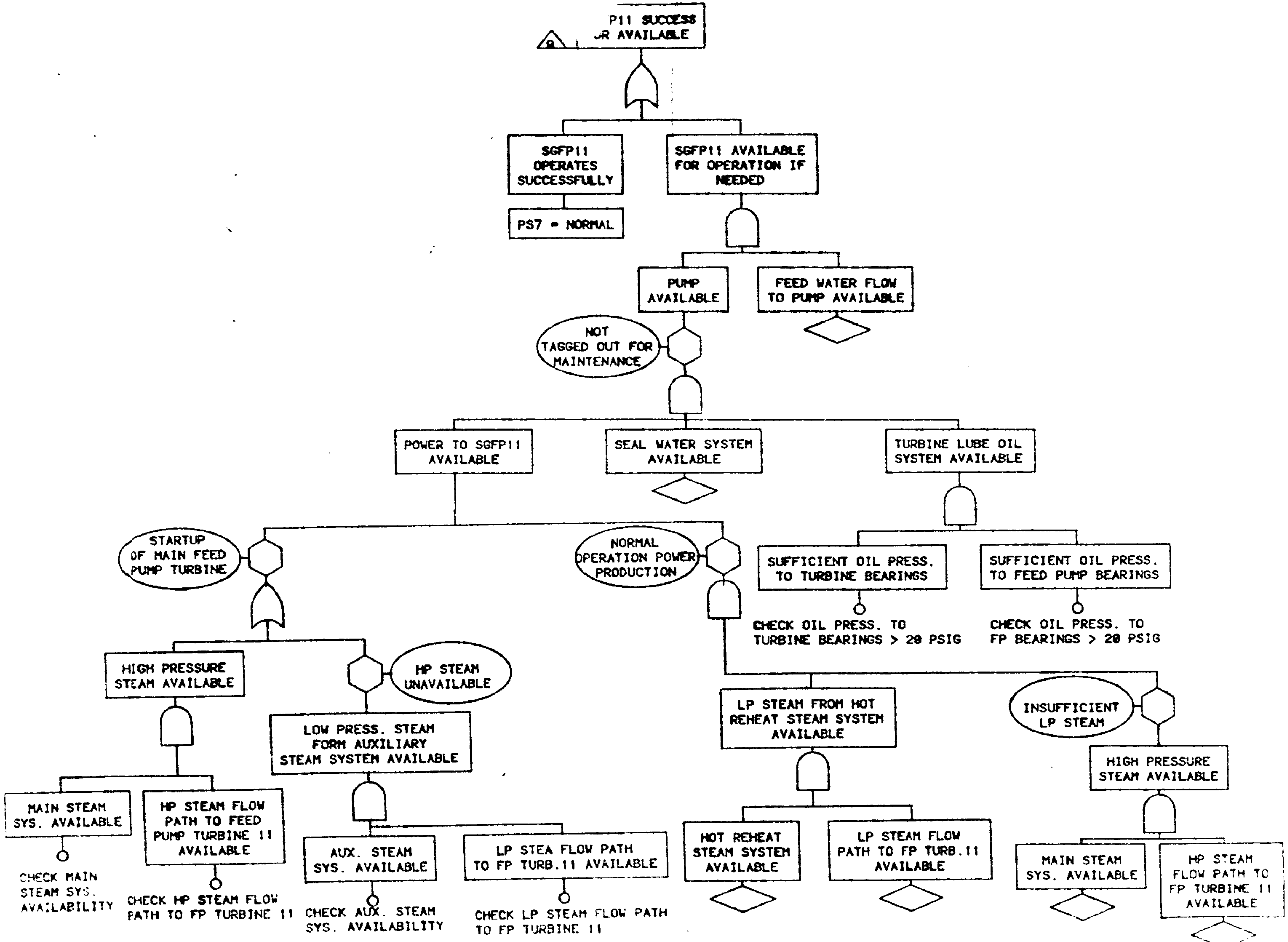


Fig. 3 (continued)

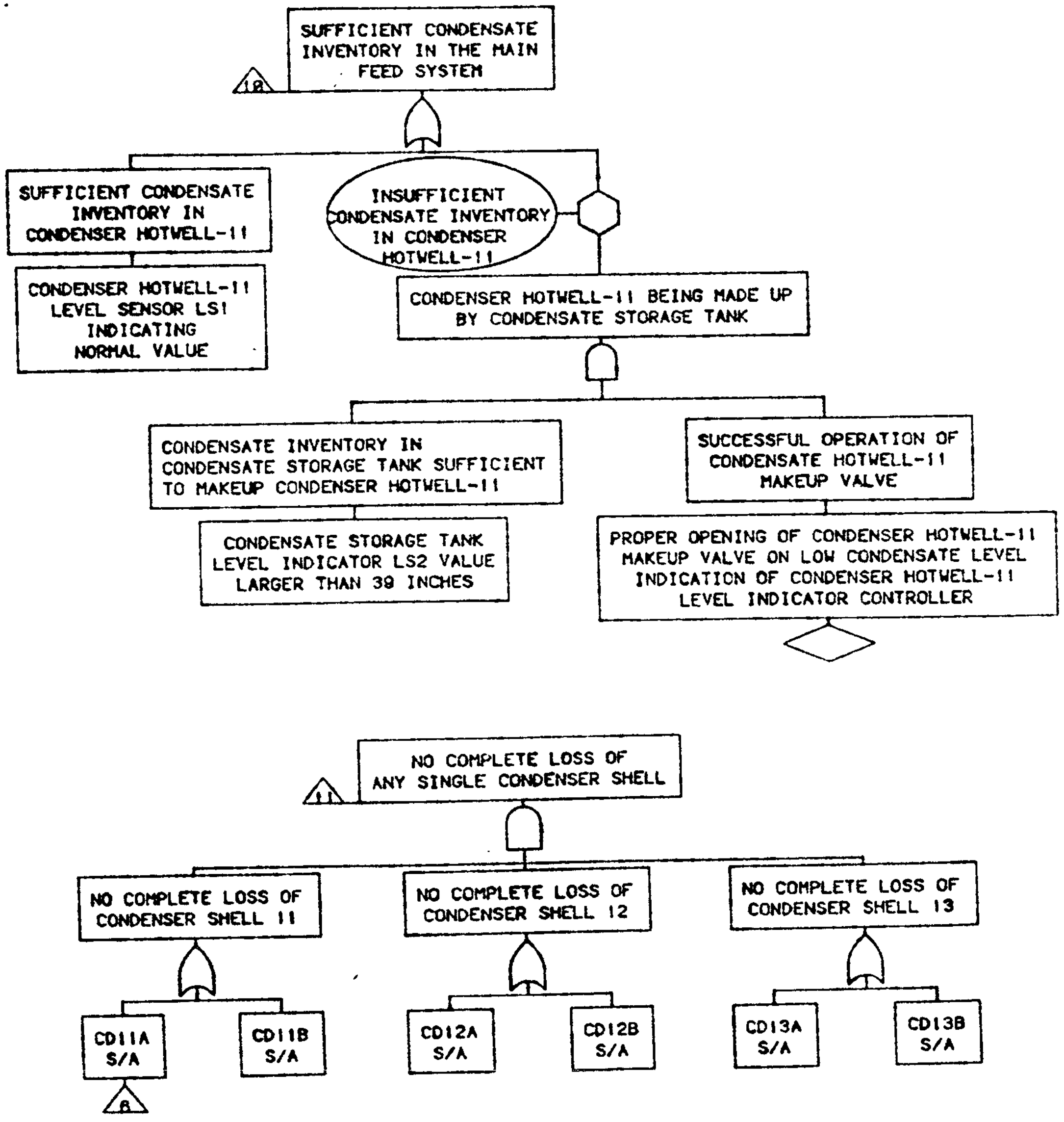


Fig.3 (continued)

```
                          <<CFWAVA>>
Was the previous-power-level 75%?
   If yes, type y. Otherwise, enter the previous-power-level, e.g. 75:  y
Was the previous-success-path ((CD11A CD11B CD12A CD13A CD13B) (CDP12 CDP13) (CD
BP11 CDBP13) (SGFP11 SGFP12))?
   If yes, type y. Otherwise, enter the previous-success-path:  y
GET-SENSOR-DATA:
   Enter TS1-sensor-value in degree-Fahrenheit, e.g. 70:  67
   Enter TS2-sensor-value in degree-Fahrenheit:  78
   Enter TS3-sensor-value in degree-Fahrenheit, e.g. 70:  69
   Enter TS4-sensor-value in degree-Fahrenheit:  83
   Enter TS5-sensor-value in degree-F.:  66
   Enter TS6-sensor-value in degree-F.:  77
   Enter TS9-sensor-value in degree-F.:  69
   Enter TS10-sensor-value in degree-F.:  80
   Enter TS11-sensor-value in degree-F.:  68
   Enter TS12-sensor-value in degree-F.:  78
   Enter FS1-sensor-value [L/N/H]:  N
   Enter PS2-sensor-value [L/N/H]:  N
   Enter PS3-sensor-value [L/N/H]:  N
   Enter FS2-sensor-value [L/N/H]:  L
   Enter PS4-sensor-value [L/N/H]:  N
   Enter PS6-sensor-value [L/N/H]:  L
   Enter FS3-sensor-value [L/N/H]:  N
   Enter PS7-sensor-value [L/N/H]:  N
   Enter PS8-sensor-value [L/N/H]:  N
   Enter LS1-sensor-value [L/N/H]:  N
   Enter LS2-sensor-value in inches, e.g. 45:  66

CFWS-STATE-DIAGNOSIS:
   Lost-components:  (CD11B CDBP13)

AVAILABILITY CHECK:
   Check tagging-status of circulating-water-pump-14 (tagged-out-for-maintenance
or not-tagged-out [TAG/NO]):  NO
   Check motor-ammeter for circulating-water-pump-14 [OK/NO]:  OK
   Check lube-oil for circulating-water-pump-14 [OK/NO]:  OK
   Check seal-water-flow for circulating-water-pump-14 [OK/NO]:  OK
   .... CD12B Available!
```

Fig. 4 A Sample Session of CFWAVA Run

```
   Check tagging-status of condensate-booster-pump-12 (tagged-out-for-maintenance
or not-tagged-out [TAG/NO]):  TAG
   Check motor-ammeter for condensate-booster-pump-12 (4KV-Bus13) [OK/NO]:  OK
   Check condensate-booster-pump-12 lube-oil-pressure > 12 psig  [OK/NO]:  NO
   .... CDBP12 Unavailable!


<<CFWAVA-SUMMARY-AND-RESULT>>
   PREVIOUS-POWER-LEVEL:  75%
   PREVIOUS-SUCCESS-PATH:  ((CD11A CD11B CD12A CD13A CD13B) (CDP12 CDP13) (CDBP11
CDBP13) (SGFP11 SGFP12))
   SYSTEM-STATE-DIAGNOSIS:
     Lost-components:  (CD11B CDBP13)
     Available-components:  (CD12B)
     Unavailable-components:  (CD11B CDBP13 CDBP12)
   NEW-OPTIMAL-GOAL:  Maintain power at 50% level!
   NEW-OPTIMAL-SUCCESS-PATH:  ((CD11A CD12A CD13A CD13B CD12B) (any-one-of (CDP12
CDP13)) (CDBP11) (any-one-of (SGFP11 SGFP12)))
   REQUIRED-ACTIONS:
     Turn-on-switch:
          circulating-water-pump-14
     Turn-off-switch:
          any-one-of (condensate-pump-12 condensate-pump-13)
          any-one-of (steam-generator-feed-pump-11 steam-generator-feed-pump-12)

(END OF CFWAVA RUN)
```

Fig. 4 (continued)