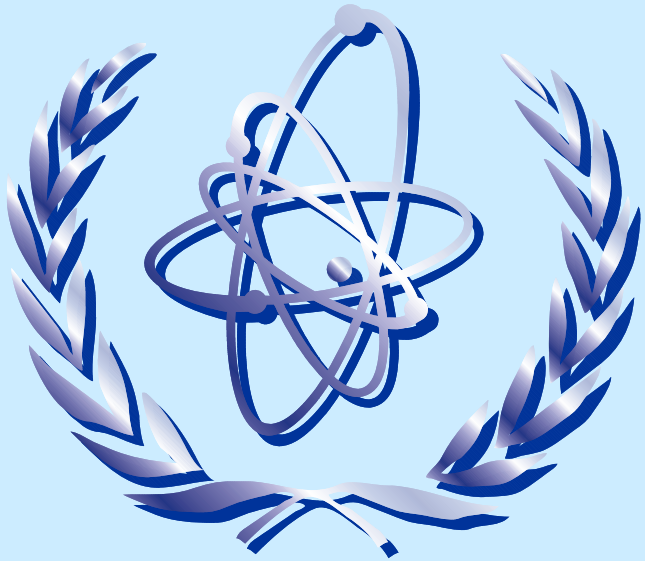
# Basic Level 1. PSA course for analysts



## System Analysis in a PSA

## part 1

# Content

- System Modelling Methods:
  - Failure Mode and Effect Analysis
  - Event Trees
  - Markov Modelling
  - Fault Trees
- Boolean Algebra
- Fault Trees
  - Fault Tree Structure: Gates and Events
  - Fault Tree Construction
  - Quantification: Cutsets

# Introduction

- System modelling is an attempt to evaluate the potential for failure or success of a system.
- The system boundary and the initial conditions of the system must be carefully defined.
- The analyst must understand how the system is designed to operate and how it can fail to meet the design requirement.
- The model of the system can be a success model, or a failure model; a graphical model, a tabular model, or a linguistic model.
- All the models study the relationship between CAUSE and EFFECT.

# FAILURE MODE AND EFFECT ANALYSIS

- **Objectives:** to determine how the failure of a component, sub-system, or system module affects the whole system.
- **History:** originated in the Space and defence industries, for the design of high reliability systems. NO SINGLE FAILURE SHOULD CAUSE SYSTEM FAILURE.
- This is a CAUSE ⇒EFFECT technique; i.e., starting from all possible causes (basic failures), the analyst assesses the effects on the system.
- A useful way to elicit the knowledge of experts in components or specific system parts.
- In practice FMEA can be very time consuming, and so FMEA may be used as a complement to fault tree analysis.

# FAILURE MODE AND EFFECT ANALYSIS (Procedure)

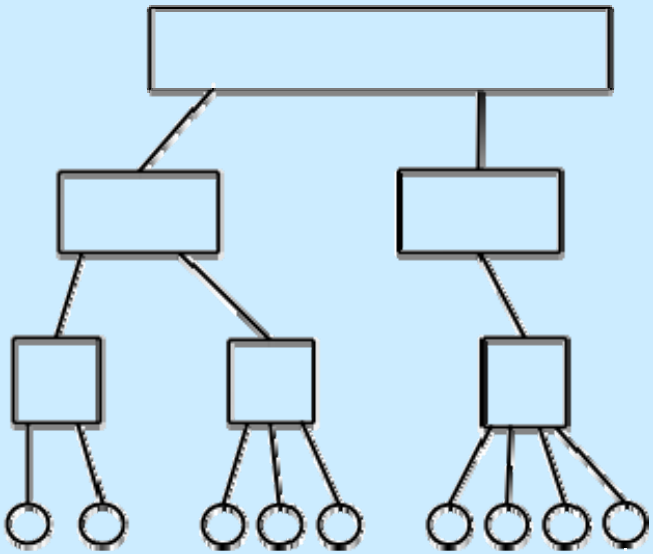
- Simplify the task by splitting the system into different levels. This may be done bearing in mind the expertise necessary to analyse each sub-system / module / component.

SYSTEM

SUB-SYSTEMS

MODULES

COMPONENTS



# FAILURE MODE AND EFFECT ANALYSIS (Procedure) (cont'd)

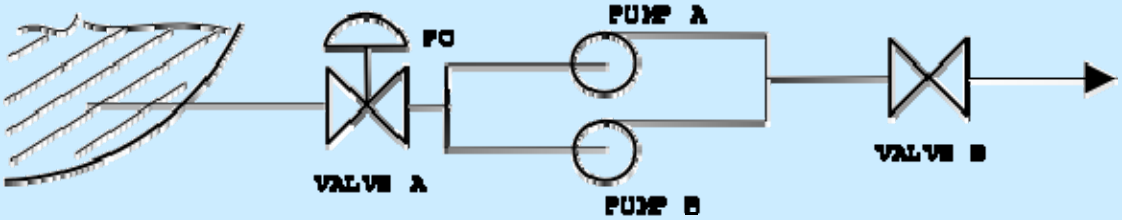
- In this way the various levels of the system may be analysed independently by experts with the necessary knowledge.
- The effects at one level become the failure modes at the next level.
- This approach can be useful in reducing the amount of time spent in the analysis.
- A TOP DOWN approach may help reduce the size of the task:

  Consider the effect of sub-system failures on the system. Subsequently, only sub-systems that seriously affect the main system need to be considered.

  This could be repeated for the effect of module failures on the important sub-systems.

# FAILURE MODE AND EFFECT ANALYSIS (Example)

## Diagram of System

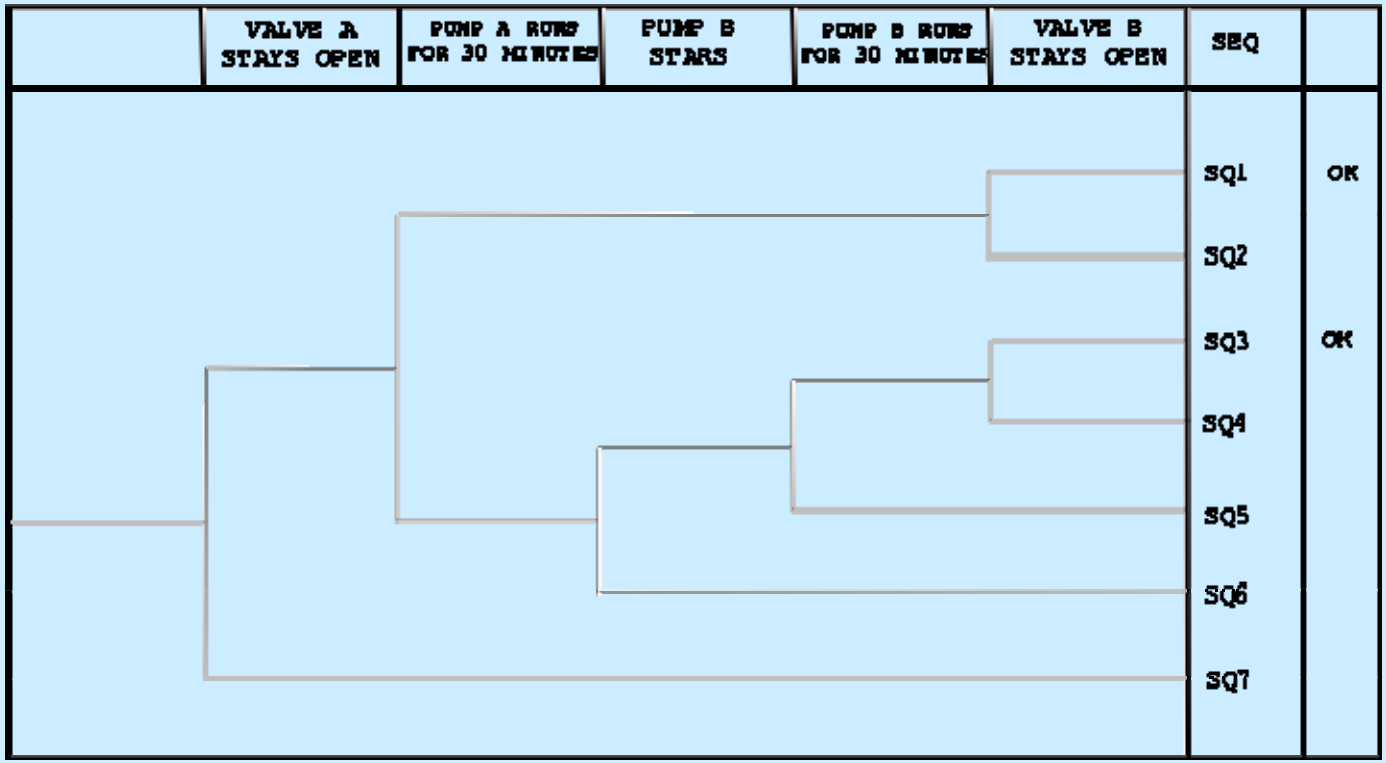



| COMPONENT | FAILURE MODE | EFFECT |
|---|---|---|
| Valve A | Spurious closure | No flow to pumps |
| Valve B | Fails to stay open | No flow out of system |
| Pump A | Fails to run | No flow out of pump A, pump B is required |
| Pump B | Fails to start | No flow out of system (if pump A has failed) |
| | Starts spuriously | System flow increases |
| | Fails to run | No system flow (if pump A has failed) |

# EVENT TREES AS A SYSTEM RELIABILITY MODEL

- **EVENT TREE MODEL OF SYSTEM:**

Notes: Pump A runs continuously,
Pump B starts if pump A stops



## EVENT TREES - ADVANTAGES AND DISADVANTAGES

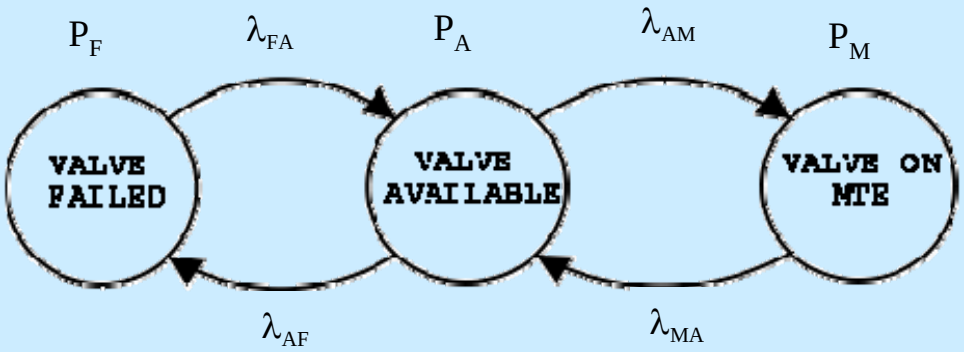
- Event trees can become very large when used as a system reliability model
- Can be useful when success criteria are complicated.
- Account for success and failures, and so the calculated answer is exact, providing that dependencies are correctly accounted for.
- More computer codes are available for quantifying fault trees.

# MARKOV MODELLING

- Objectives: to find the probability of a specific system or component state.
- Time dependency can be included.
- The model takes the form of a chain: the various states are linked by the probability of a transition from one state to another (see example - next slide).
- Can be a useful model for obtaining component data.
- Dynamic model (probabilities change with time), but it is also possible to find the long term steady state (when probabilities reach constant values).
- Pictorial representation can help understand the problem.
- Computer solution is straightforward.
- However, diagrams can become very complex.

# MARKOV MODELLING - EXAMPLE

- Consider a component, for example a valve.
- We may be interested in the probability that it is unavailable on demand. This may be due to two causes: the valve is on maintenance, or the valve has failed.
- Markov Model:




$P_i$ = Probability of state i

$\lambda_{ij}$ =Transition probability per unit time

# MARKOV MODELLING - EXAMPLE (cont'd)

- **Equations:**

$$\frac{dP_A}{dt} = -P_A \cdot \lambda_{AF} + P_F \cdot \lambda_{FA} - P_A \cdot \lambda_{AM} + P_M \cdot \lambda_{MA}$$

$$\frac{dP_F}{dt} = -P_F \cdot \lambda_{FA} + P_A \cdot \lambda_{AF}$$

$$\frac{dP_M}{dt} = P_A \cdot \lambda_{AM} - P_M \cdot \lambda_{MA}$$

$P_M + P_A + P_F = 1$ (The component must be in one of the 3 states)

A STEADY STATE solution can be found by setting:

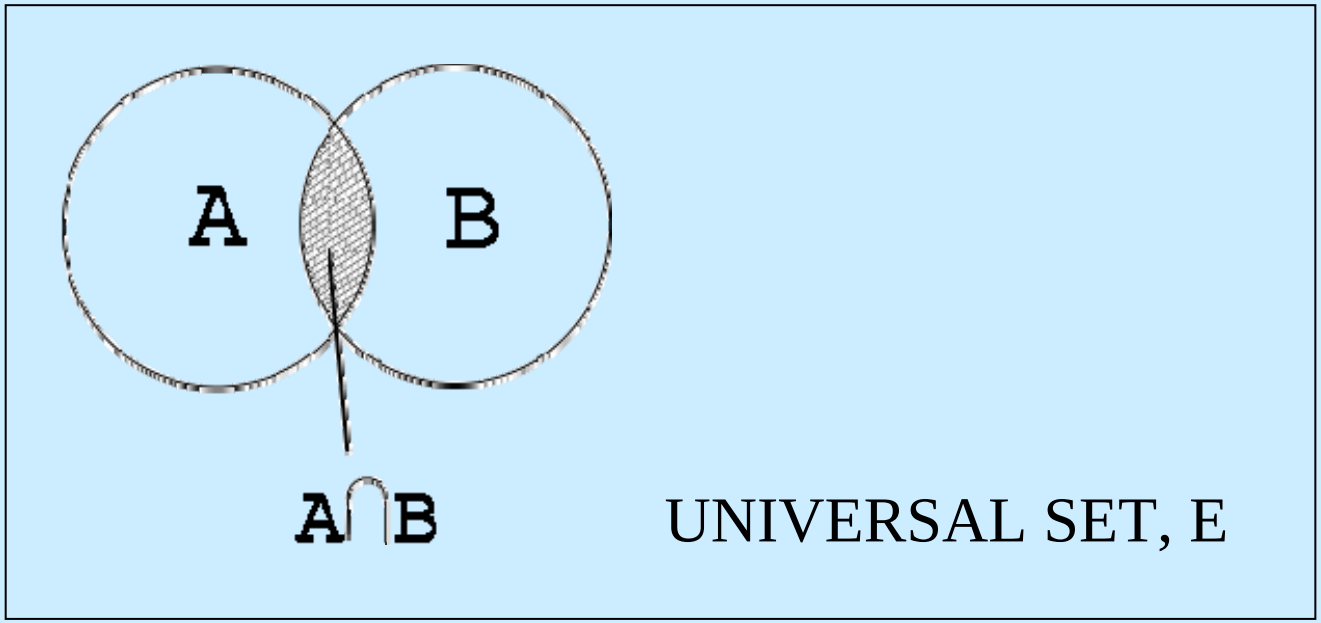
$$\frac{dP_A}{dt} = \frac{dP_F}{dt} = \frac{dP_M}{dt} = 0$$

$$\Longrightarrow \quad P_A = \lambda_{FA} \Big/ \left( \lambda_{AF} + \lambda_{FA} + \frac{\lambda_{AM} \cdot \lambda_{FA}}{\lambda_{MA}} \right)$$

$$P_{unavailable} = P_F + P_M = \left( \lambda_{AF} + \frac{\lambda_{AM} \cdot \lambda_{FA}}{\lambda_{AM}} \right) \Big/ \left( \lambda_{AF} + \lambda_{FA} + \frac{\lambda_{AM} \cdot \lambda_{FA}}{\lambda_{MA}} \right)$$

# SETS AND BOOLEAN ALGEBRA

- **A SET is a collection of outcomes that conform to a specific criterion (e.g., Set A may be the collection of outcomes that include the failure of valve X).**



# SETS AND BOOLEAN ALGEBRA (cont'd)

- The UNIVERSAL SET, E, is the set that contains all possible outcomes.
- Set B could be, for example, the set of outcomes that contain failures of pump Y.
- The INTERSECTION, , is the set of all outcomes that contain failures of valve X AND pump Y.
- The UNION, , is the set containing all the outcomes in set A + all the outcomes in set B.

## SETS AND BOOLEAN ALGEBRA - RELATION TO PROBABILITY

- **The probability of an event in set A is**

  **P(A) = (number of outcomes in A) / (number of outcomes in universal set)**

  **= N(A) / N(E)**

- **Useful results (proofs not given here):**

$$P(A \cap B) = P(A) \cdot P(B)$$ if the events are independent

$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$

- **The next slide shows how boolean expressions can be simplified, and how the probability can be calculated practically**

# SETS AND BOOLEAN ALGEBRA

- **Useful results for simplifying boolean expressions are:**

1. $A \cap A = A$
2. $A \cup A = A$
3. $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$
4. $A \cup (A \cap B) = A$

- **Using these results, any complicated boolean expression can be written in the following form:**

$$X \cap Y \cap Z ..... \cup A \cap B \cap C ..... \cup .... \cup ....$$

**This is the basis of the CUTSET concept used in fault tree analysis.**

# SETS AND BOOLEAN ALGEBRA (cont'd)

- **The probability is:**

$$P(X \cap Y \cap Z \ldots \cup A \cap B \cap C \ldots \cup \ldots \cup \ldots)$$

- **1 assumption and 1 approximation allow this to be quantified simply:**

1. **X, Y, Z, A, B, C are all independent**

   **= >** $P(X \cap Y \cap Z) = P(X) \cdot P(Y) \cdot P(Z)$

2. **The rare event approximation:**

$$P(A \cup B) = P(A) + P(B)$$

**(compare to previous slide – this is generally a good approximation since is usually small,**
**e.g., ( P(A) = 0.1 ) * ( P(B) = 0.1 ) = .01 )**

# SETS AND BOOLEAN ALGEBRA (cont'd)

- **Applying the assumption of independence and the rare event approximation:**

$$P(X \cap Y \cap Z ... \cup A \cap B \cap C ... \cup ... \cup ...)$$
$$= P(X) \cdot P(Y) \cdot P(Z) ... + P(A) \cdot P(B) \cdot P(C) ... + ... + ...$$

**Which is an expression for the probability in terms of the (known) probability of the individual events.**

# SETS AND BOOLEAN ALGEBRA (cont'd)

- **Comment on notation.**

  **The following is commonly used (known as engineering notation):**

  $\cap$ **is replaced by** $\cdot$

  $\cup$ **is replaced by** $+$

  **So,**

  $$X \cap Y \cap Z \cup A \cap B \cap C$$

  **is often written**

  $$X \cdot Y \cdot Z + A \cdot B \cdot C$$

# FAULT TREES

- Reference: The Fault Tree, Handbook (NUREG-0492 Jan 1981 )
- The undesired state of a system is specified.
- The fault tree analysis finds all the credible ways in which the undesired event can occur.
- A fault tree is a structure formed by "gates" that allow or stop the progress of the fault up the tree.
- The gates show the relationship of events needed for the occurrence of the higher event.
- The higher event is the output of the gate, the lower events are the inputs to the gate.

# SYMBOLOGY

## Primary Events

events that for one reason or another have not been further developed.

**BASIC EVENT**:

event that requires no further development - the appropriate limit of resolution has been reached

**UNDEVELOPED EVENT**:

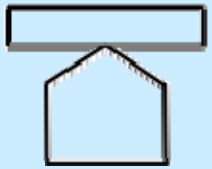
an event that is not further developed

**CONDITIONING EVENT**:

represents conditions or restrictions that apply to a gate (it is used with INHIBIT and PRIORITY AND gates)

# SYMBOLOGY (cont'd)

**EXTERNAL EVENT**:



represents an event that is expected to occur. Often called a HOUSE EVENT

**INTERMEDIATE EVENT**:

is an event, which occurs because of one or more causes acting through logic gates.

**Transfer Symbols** (only for drawing purpose)

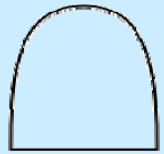
Transfer in

Transfer out

# GATES

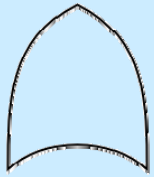
**There are two basic types of gates: OR gates and AND gates (all the rest are special cases of these two basic gates)**

**AND gate**



**Output occurs if all the input faults occur.**

**OR gate**

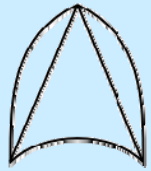


**Output occurs if at least one of the input faults occur.**

# GATES (cont'd)

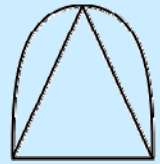
**EXCLUSIVE OR gate**

Output occurs if exactly one of the input faults occurs.



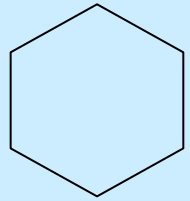
**EXCLUSIVE AND gate**

Output occurs if all of the input occurs in a specific sequence.



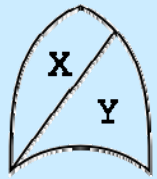
**INHIBIT gate**

Output occurs if the single input occurs in the presence of CONDITIONING EVENT.



**VOTE gate**

Output occurs if X inputs out of Y inputs occur (this is a combination of AND and OR gates)

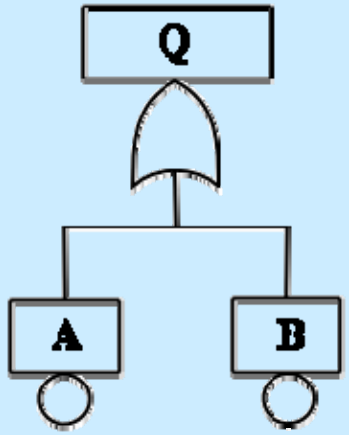


# BASIC RULES FOR FAULT TREE CONSTRUCTION

- Write the statements in the event boxes as faults.
- All inputs to a particular gate should be completely defined before further analysis of any of them is undertaken.
- Gate inputs should be properly defined fault events, and gates should not be directly connected to other gates.

# BOOLEAN ALGEBRA APPLIED TO FAULT TREE ANALYSIS

- **The OR gate represents the union of the events attached to the gate**




$Q = A \cup B \;\; (Q = A + B)$

**The minimal cutsets are:** $A \cdot B$

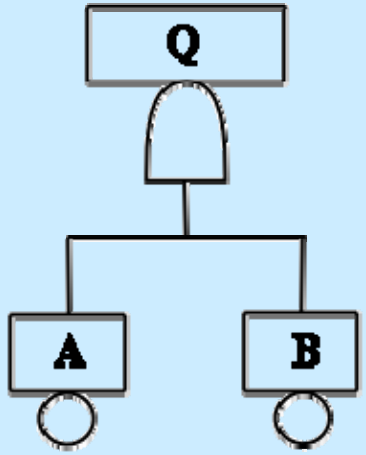
- **Using the rare event approximation:**

**The probability of the top event**

$$Q, P(Q), = P(A) + P(B)$$

# BOOLEAN ALGEBRA APPLIED TO FAULT TREE ANALYSIS (cont'd)

- **The AND gate represents the intersection of the events attached to the gate:**




$$Q = A \cap B \; (Q = A \cdot B)$$

**The minimal cutsets are:** $A \cdot B$

- **Assuming that A and B are independent:**

**The probability of event** $Q, P(Q), = P(A) \cdot P(B)$

# FAULT TREE ANALYSIS

- The qualitative analysis of the fault tree provides:
  - a) The minimal cutsets of the fault tree:

    A minimal cutset can be defined as the smallest combination of component failures which, if they all occur, will cause the top event to occur.
  - b) Component qualitative importance:

    Qualitative rankings of contribution to system failures.
  - c) Potential Common cause failures:

    Minimal cutsets potentially susceptible to a single failure cause.

# FAULT TREE ANALYSIS (cont'd)

- The quantitative analysis of the fault tree provides:

  a) Numerical probabilities:

     Probabilities of system and cutset failures.

  b) Quantitative importance:

     Quantitative rankings of contributions to system failure.

- Computer Codes:

  PSAPACK, SETS, NUPRA, etc.