Idaho National Laboratory

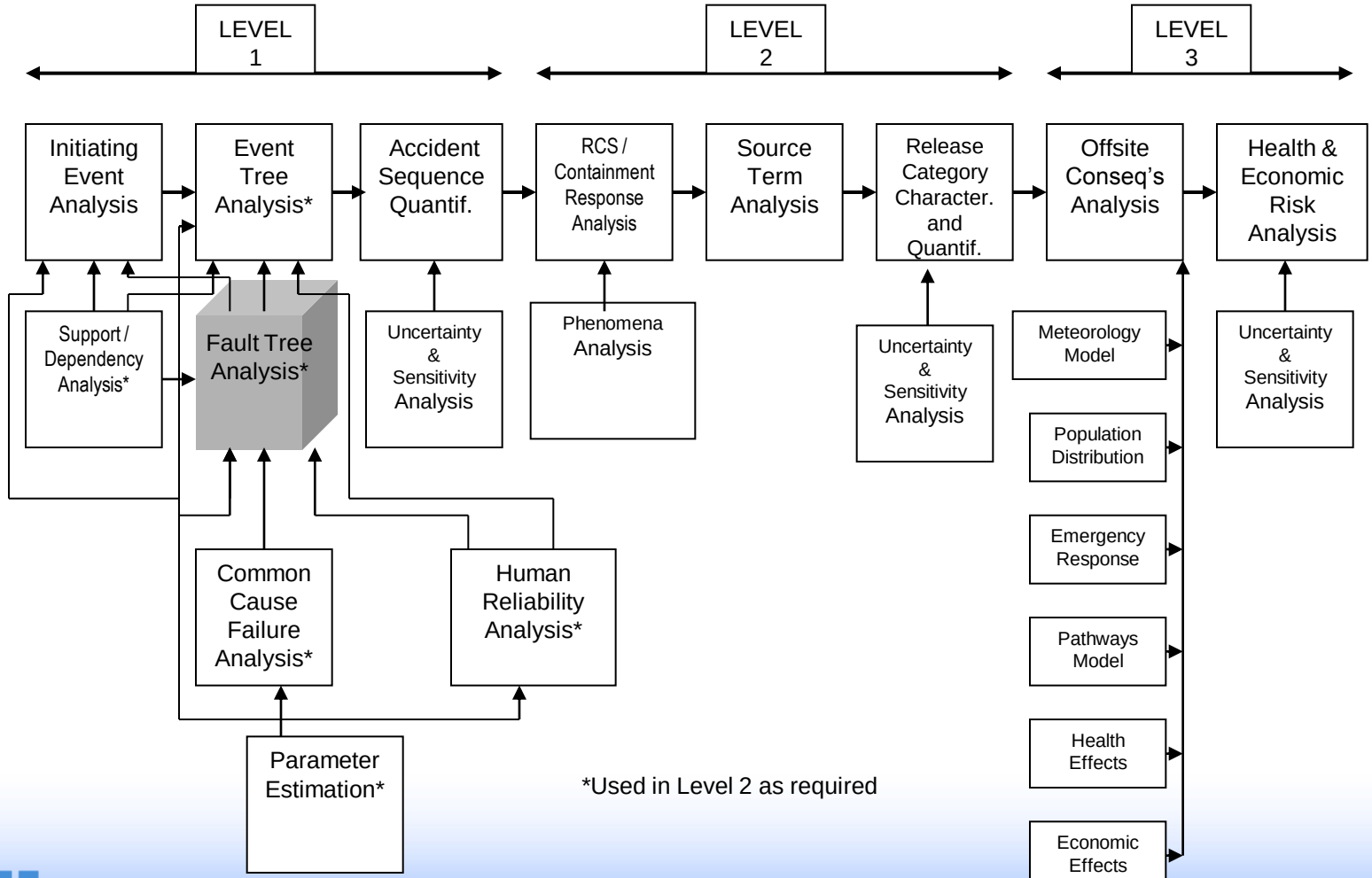
MODULE F

SYSTEMS ANALYSIS USING FAULT TREES

Fault Tree Concepts

- **Purpose: Students will learn the purposes of fault tree analysis. Students will learn how the appropriate level of detail for a fault tree analysis is established. Students will become familiar with the terminology, notation, and symbols employed in fault tree analysis.**
- **Objectives:**
  - **List the purposes of fault tree analysis**
  - **Define the terminology, notation, and symbols used in fault tree analysis**
  - **Interpret the results of fault tree reduction**
  - **Define and correctly apply the definition of "minimal cut sets"**
- **References: NUREG-0492, NUREG/CR-2300**

Principal Steps in PRA

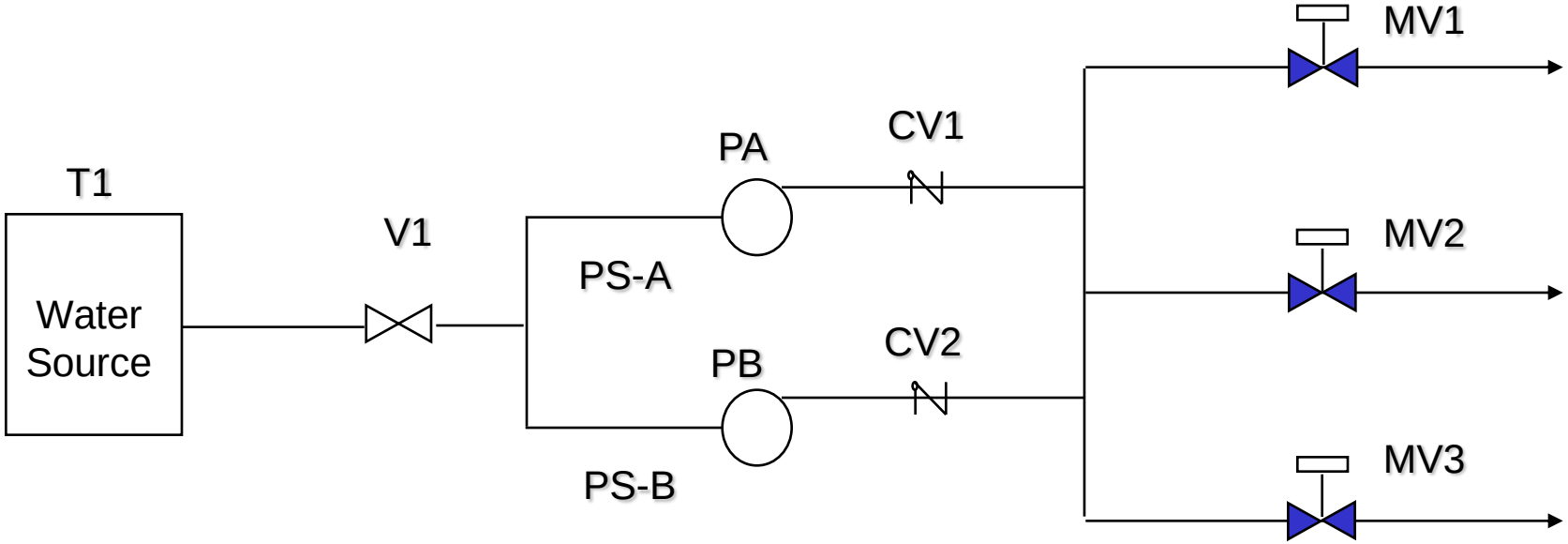


Fault Tree Analysis Definition

*"An analytical technique, whereby an **undesired state** of the system is specified (usually a state that is critical from a safety standpoint), and the system is then analyzed **in the context of its environment and operation** to find all **credible** ways in which the undesired event can occur."*

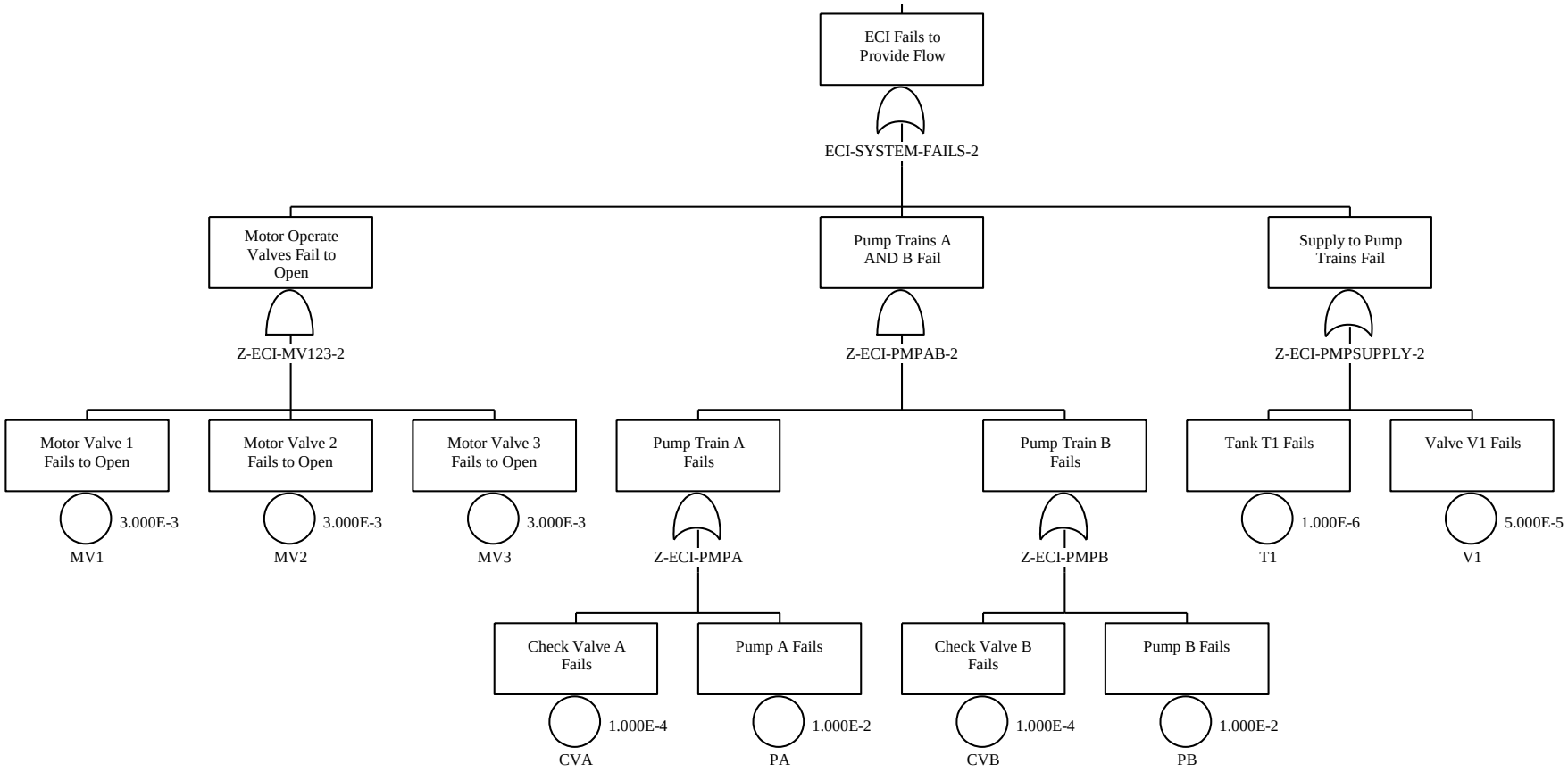
NUREG-0492

Emergency Cooling Injection System Fault Tree Example

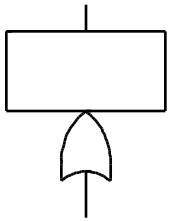
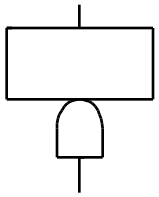
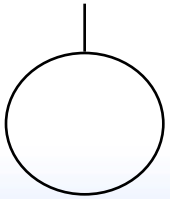


Flow from any one pump through any one MV is success

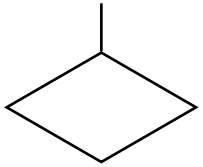
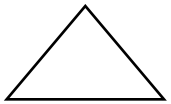
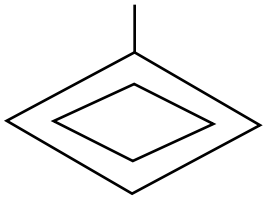
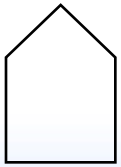
T_ - tank
V_ - manual valve, normally open
PS_ - pipe segment
P_ - pump
CV_ - check valve
MV_ - motor-operated valve, normally closed


ECI Fails to Provide Flow
ECI-SYSTEM-FAILS-2
Motor Operate Valves Fail to Open
Pump Trains A AND B Fail
Supply to Pump Trains Fail
Z-ECI-MV123-2
Z-ECI-PMPAB-2
Z-ECI-PMPSUPPLY-2
Motor Valve 1 Fails to Open
Motor Valve 2 Fails to Open
Motor Valve 3 Fails to Open
Pump Train A Fails
Pump Train B Fails
Tank T1 Fails
Valve V1 Fails
3.000E-3
3.000E-3
3.000E-3
1.000E-6
5.000E-5
MV1
MV2
MV3
Z-ECI-PMPA
Z-ECI-PMPB
T1
V1
Check Valve A Fails
Pump A Fails
Check Valve B Fails
Pump B Fails
1.000E-4
1.000E-2
1.000E-4
1.000E-2
CVA
PA
CVB
PB

Fault Tree Symbols

Symbol		Description
	"OR" Gate	Logic gate providing a representation of the Boolean union of input events. The output will occur if at least one of the inputs occur.
	"AND" Gate	Logic gate providing a representation of the Boolean intersection of input events. The output will occur if all of the inputs occur.
	Basic Event	A basic component fault which requires no further development. Consistent with level of resolution in databases of component faults.

Fault Tree Symbols (cont.)

Symbol		Description
	Undeveloped Event	A fault event whose development is limited due to insufficient consequence or lack of additional detailed information
	Transfer Gate	A transfer symbol to connect various portions of the fault tree
	Undeveloped Transfer Event	A fault event for which a detailed development is provided as a separate fault tree and a numerical value is derived
	House Event	Used as a trigger event for logic structure changes within the fault tree. Used to impose boundary conditions on FT. Used to model changes in plant system status.

Relationship Between Fault Trees and Event Trees

- As discussed in Module E, event trees consist of a series of nodes. Each node represents the success or failure of a particular system, component, or operation.
- For complex systems, fault tree models are used to model system failure and estimate the system's probability of failure.
- Therefore, the top event of a fault tree corresponds to the failure branch of its associated event tree node.

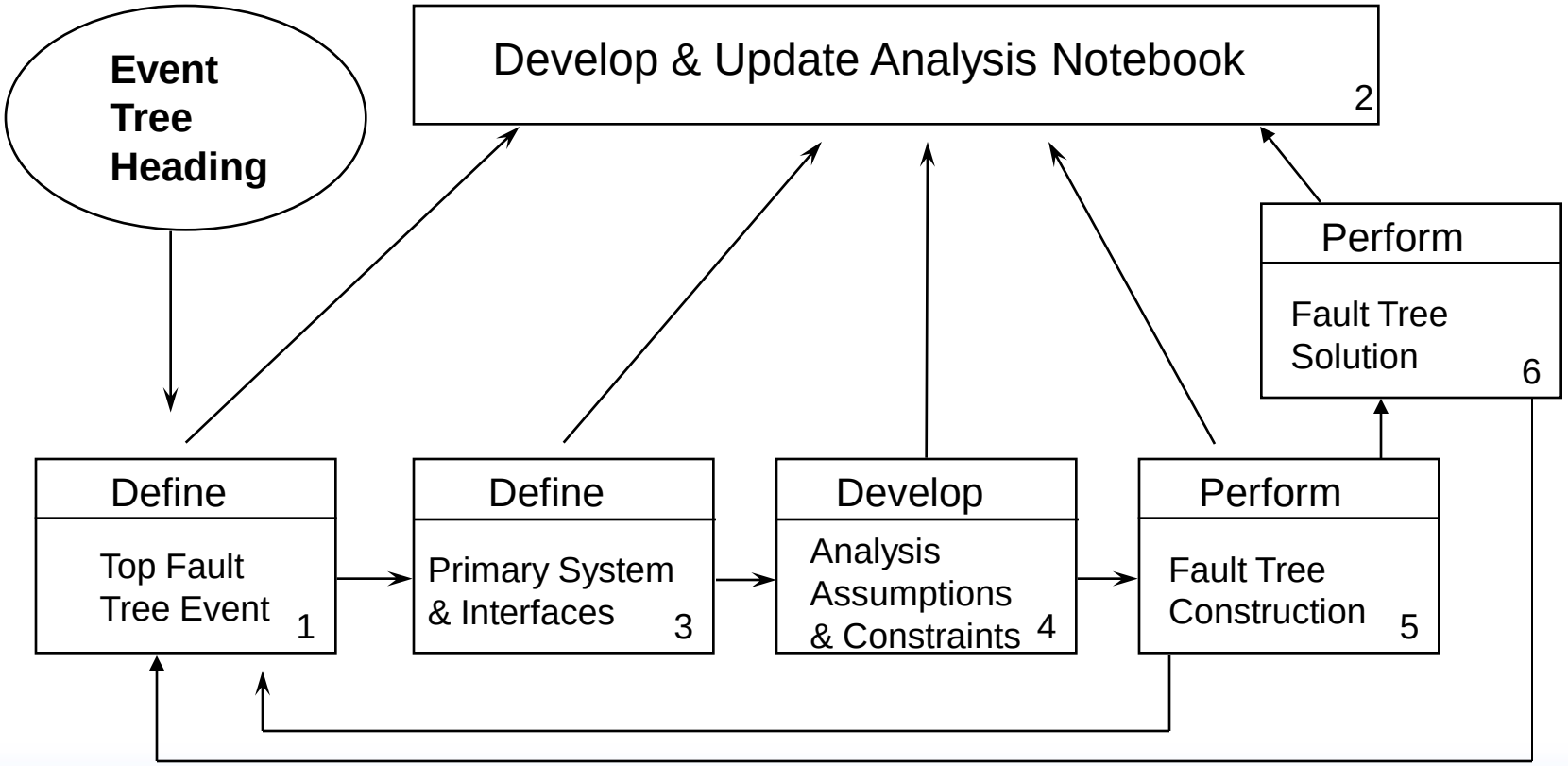
Fault Trees

- Deductive analysis (event trees are inductive)
- Top down approach starting with undesired event (top event) definition
- Explicitly models multiple failures
- Provides event relationships (i.e., combinations of events leading to undesired event)
- Used to estimate top event unreliability (i.e., probability top event fails to perform intended function)

Purpose of Fault Tree Analysis

- **Fault trees can be used to identify the ways in which a system, component, function, or operation can fail.**
- **Fault tree models can be used to determine:**
  - **Interrelationships between fault events, failure combinations producing undesired event**
  - **System "weaknesses"**
    - **Qualitative**
    - **Quantitative**
  - **System unreliability (system failure probability)**

Fault Tree Development and Analysis Process



1. Define Top Event

- **Undesired event or state of system**
  - **Often corresponds to an event on an event tree**
  - **Based on success criterion for system**
    - **Typically initiating event dependent (e.g., HPI would have different success criteria for small LOCA vs. medium LOCA)**
    - **Success criteria determined from thermal/hydraulic calculations (i.e., computer code runs made to determine how much injection is needed to keep core covered given particular IE)**
  - **Success criterion used to determine failure criterion**
    - **Fault tree top event**
  - **Will often have multiple versions of system failure fault tree**
    - **For different IEs**

2. Develop and Update System Notebooks

- **Fault tree development is an iterative process, that is related to the other PRA processes. A system notebook should be started at the onset of fault tree development; it should be maintained and updated periodically.**
  - **A system notebook should contain the following:**
    - **scope of analysis,**
    - **system definition and boundaries,**
    - **system design information,**
    - **the drawings or diagrams used for model development,**
    - **system operational information,**
    - **applicable Technical Specifications,**
    - **test and maintenance information and data,**
    - **analytical assumptions,**
    - **component failure rate data, and**
    - **fault tree results.**
  - **System notebooks were typically developed during the IPE process. Although the system notebooks are not generally included in the IPE submittal, the system notebooks should be available for review by the inspectors.**

3. Define the System and Interfaces

- **Define system/component boundaries based on:**
  - **the information required from the analysis and**
  - **the basic event level (i.e., the level of resolution of available data)**
  - **function of the system being modeled**
  - **Note: boundaries may not be consistent with those used in the plant**
- **Identify shared components with other systems.**
- **Identify dependencies on other systems.**

4. Develop Analysis Assumptions and Constraints

- **Analytical assumptions must be made to compensate for incomplete knowledge of: plant response, system response, system operation, failure modes and mechanisms, and potential recovery actions.**
- **The rationale for assumptions should be specified and documented. Whenever possible, it should be supported by engineering analysis.**
- **Time and/or budget constraints, as well as the tools available for performing the analysis, can often contribute to defining the scope of the analysis.**

5. Fault Tree Construction

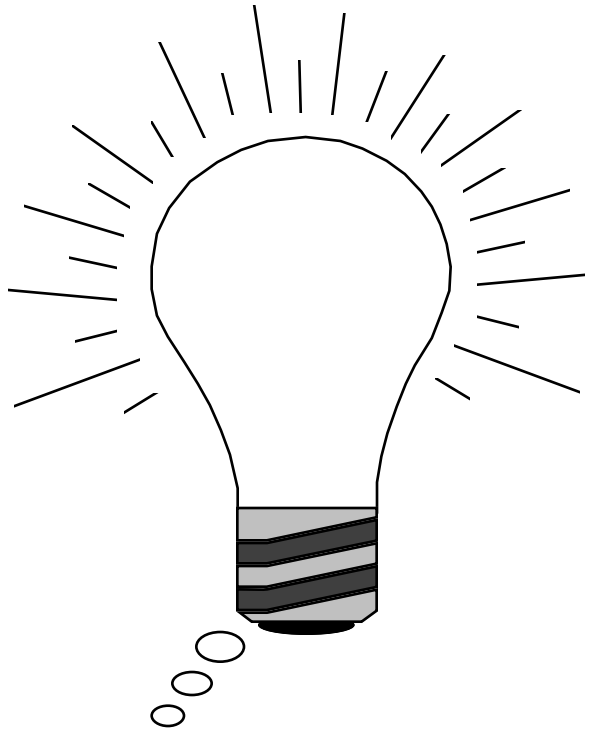
- **Fault tree construction requires the step-by-step postulation of system faults, starting at the top event and working down to the basic events whose failures contribute to the top event failure.**
- **A standardized symbology is employed.**
- **Postulation should be consistent with the level of resolution in the available data and the analytical assumptions.**
- **Fault tree construction is an iterative process requiring constant feedback from the other PRA processes as well as the other steps in the fault tree development process.**

6. Fault Tree Solution

- Due to the complexity of most fault trees, computers are used to generate results. This produces a list of the various combinations of basic event failures that cause the top event to occur.
- Fault tree results - the list of various combinations are called Minimal Cut Sets.
- Solution relies on "laws" of Boolean algebra.
- Because typical models are very large, solution most often approximated by performing minimal cut set truncation.

Minimal Cut Set

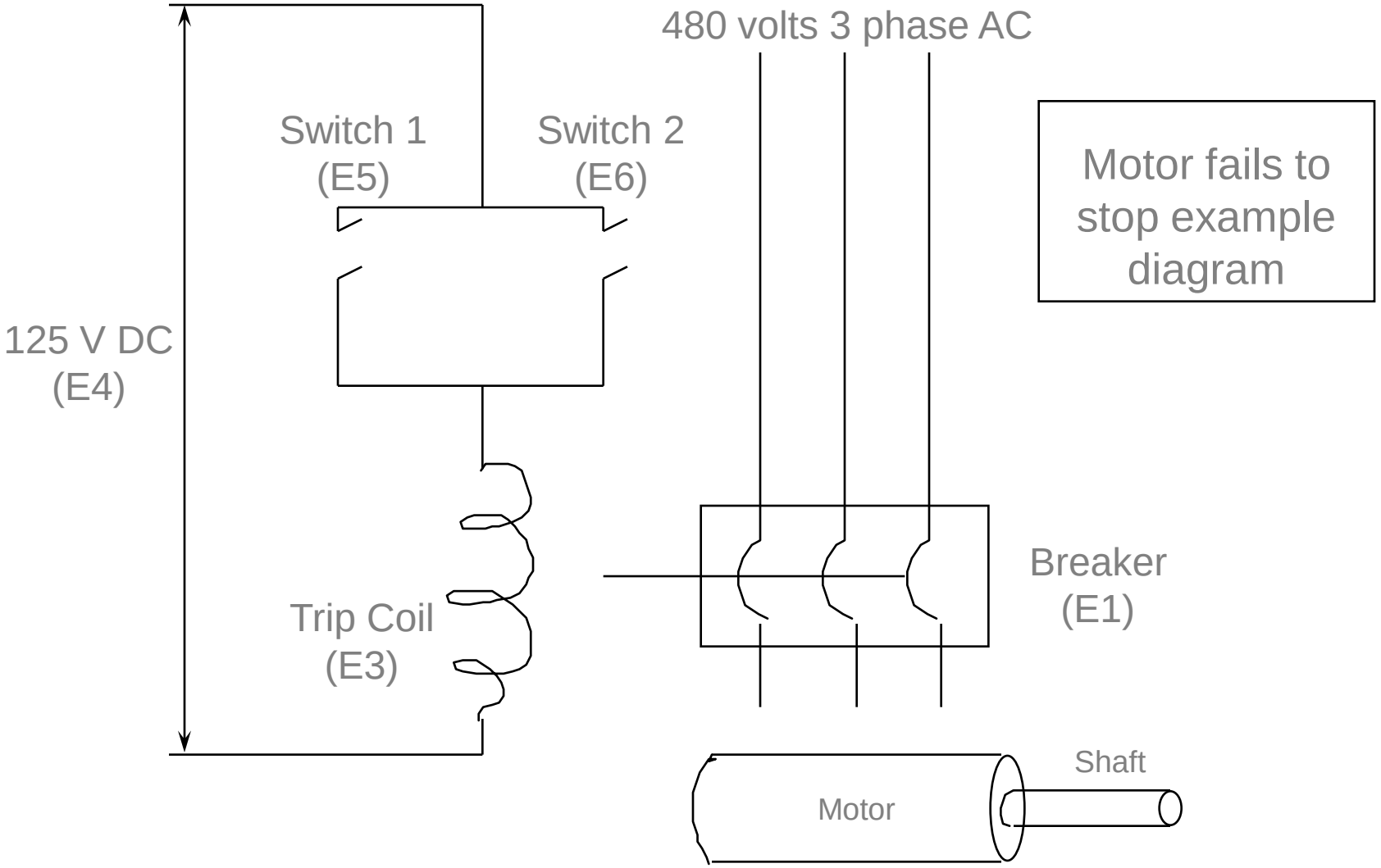
A group of basic failures (component failures and/or human errors) that are *collectively necessary* and *sufficient* to cause the TOP event to occur.

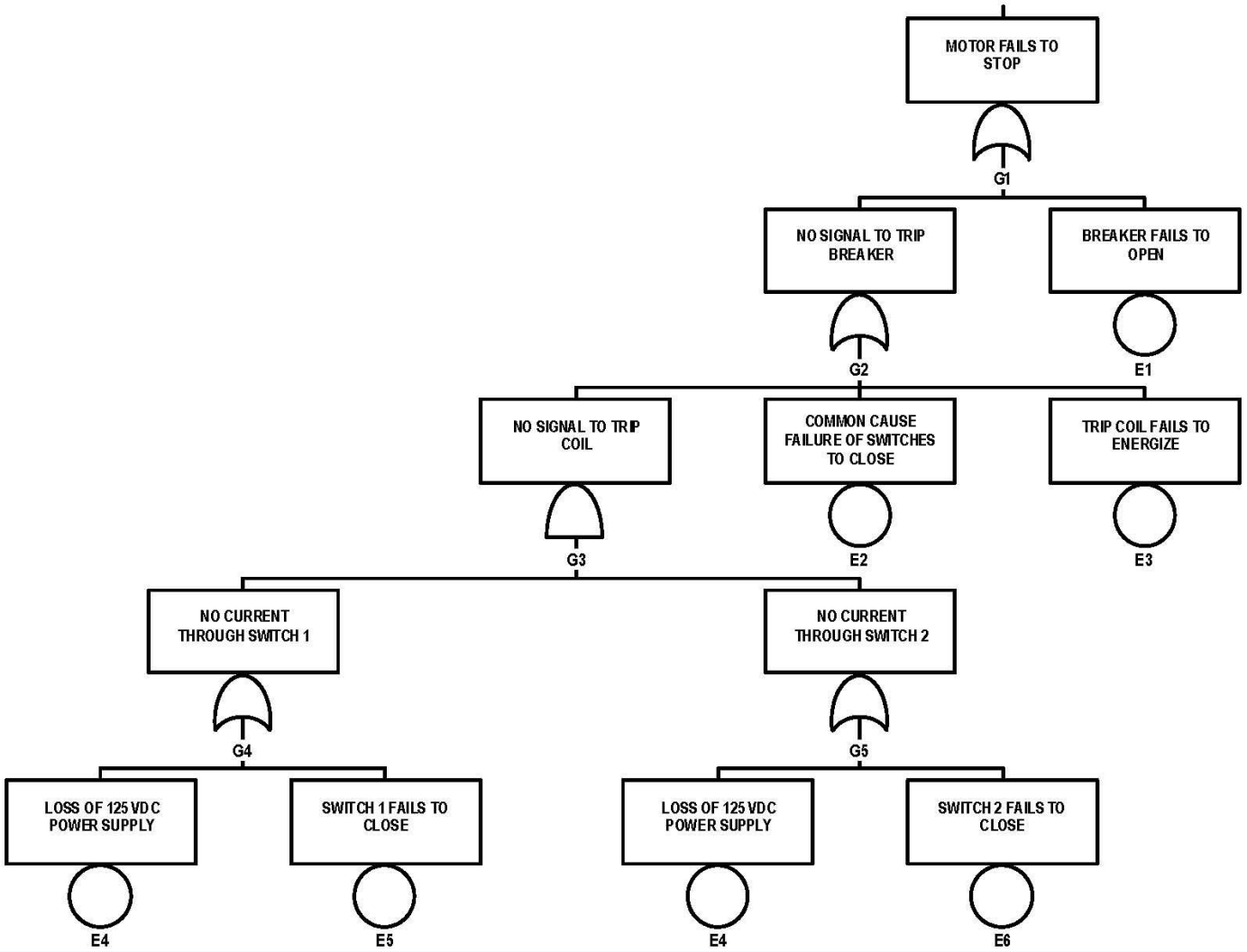


Understanding the concept of minimal cuts sets is one of the most important steps in understanding PRA

Demonstration of the Fault Tree Construction & Solution Process

- **Build fault tree for the schematic provided**
- **Some assumptions:**
  - **Ignore wire faults**
  - **Do not model details of 125 V DC power supply**
- **Will solve fault tree and discuss "meaning" of the solution process**


480 volts 3 phase AC
Switch 1
(E5)
Switch 2
(E6)
Motor fails to stop example diagram
125 V DC
(E4)
Trip Coil
(E3)
Breaker
(E1)
Shaft
Motor


MOTOR FAILS TO STOP
G1
NO SIGNAL TO TRIP BREAKER
BREAKER FAILS TO OPEN
G2
E1
NO SIGNAL TO TRIP COIL
COMMON CAUSE FAILURE OF SWITCHES TO CLOSE
TRIP COIL FAILS TO ENERGIZE
G3
E2
E3
NO CURRENT THROUGH SWITCH 1
NO CURRENT THROUGH SWITCH 2
G4
G5
LOSS OF 125 VDC POWER SUPPLY
SWITCH 1 FAILS TO CLOSE
LOSS OF 125 VDC POWER SUPPLY
SWITCH 2 FAILS TO CLOSE
E4
E5
E4
E6

Boolean Fault Tree Reduction

- First, express a fault tree's logic as a Boolean Equation.
- Then, apply the rules of Boolean Algebra to reduce the terms.
- This results in a reduced form of the Boolean Equation, which can be used to quantified the fault tree in terms of its minimal cut sets.
- Boolean reduction is typically done automatically by the fault tree software during the quantification process.

Fault Tree Results

- **Fault tree solution results in a list of minimal cut sets.**
- **Each minimal cut set is a combination of basic events.**
- **Each minimal cut set has an individual probability of occurrence that is equal to the product of the basic event failure probabilities.**
- **The probability that the top event will occur is approximately the sum of the individual cut set probabilities.**

Rules of Boolean Algebra

Mathematical Symbolism	Engineering Symbolism	Designation
(1a) $X \cap Y = Y \cap X$	X * Y = Y * X	Commutative Law
(1b) $X \cup Y = Y \cup X$	X + Y = Y + X	
(2a) $X \cap (Y \cap Z) = (X \cap Y) \cap Z$	X * (Y * Z) = (X * Y) * Z	Associative Law
(2b) $X \cup (Y \cup Z) = (X \cup Y) \cup Z$	X + (Y + Z) = (X + Y) + Z	
(3a) $X \cap (Y \cup Z) = (X \cap Y) \cup (X \cap Z)$	X * (Y+Z) = (X * Y) + (X * Z)	Distributive Law
(3b) $X \cup (Y \cap Z) = (X \cup Y) \cap (X \cup Z)$	X + (Y * Z) = (X + Y) * (X + Z)	
(4a) $X \cap X = X$	X * X = X	Idempotent Law
(4b) $X \cup X = X$	X + X = X	
(5a) $X \cap (X \cup Y) = X$	X * (X + Y) = X	Law of Absorption
(5b) $X \cup (X \cap Y) = X$	X + (X * Y) = X	

Reduction of Example Fault Tree

- **Top down logic equations (+ = "OR", ∗ = "AND")**

  G1 = G2 + E1

  G2 = E2 + G3 + E3

  G3 = G4 ∗ G5

  G4 = E4 + E5

  G5 = E4 + E6

- **Back-substitute**

  G3 = (E4 + E5) ∗ (E4 + E6)

  G2 = E2 + [(E4 + E5) ∗ (E4 + E6)] + E3

  G1 = E2 + [(E4 + E5) ∗ (E4 + E6)] + E3 + E1

Reduction of Example Fault Tree (cont.)

- **Expand parentheses**

  G1 = E2 + E4 * E4 + E4 * E6 + E5 * E4 + E5 * E6 + E3 + E1

- **Reduce terms using rules of Boolean Algebra**
  - **Idempotent Law applies to E4 * E4 = E4**

    G1 = E2 + [E4 * E4] + E4 * E6 + E5 * E4 + E5 * E6 + E3 + E1

    G1 = E2 + [E4] + E4 * E6 + E5 * E4 + E5 * E6 + E3 + E1
  - **Law of Absorption applies to E4 + (E4 * "Y") = E4**

    G1 = E2 + [E4 + (E4 * E6)] + E5 * E4 + E5 * E6 + E3 + E1

    G1 = E2 + [E4] + E5 * E4 + E5 * E6 + E3 + E1

    G1 = E2 + [E4 + (E5 * E4)] + E5 * E6 + E3 + E1

    G1 = E2 + [E4] + E5 * E6 + E3 + E1

- **Reduced equation is list of minimal cut sets, each minimal cut set separated by "+"**

  G1 = E1 + E2 + E3 + E4 + (E5 * E6)

  Pr(G1) ≈ Pr(E1) + Pr(E2) + Pr(E3) + Pr(E4) + [Pr(E5) * Pr(E6)]

**** Fault Tree Exercise ****

- **Using the AFW fault tree from North Anna IPE (provided in Volume 2 of course material), identify various fault tree elements;**
  - **top event,**
  - **the various types of logic gates and gate names,**
  - **the use of house events,**
  - **transfers (including transfers to support systems),**
  - **undeveloped events, and**
  - **basic events and basic event names, noting examples of human error and common cause failure.**
- **Review your IPE for fault tree models and note the various fault tree elements.**