Quantitative Fire Risk Analysis

Ir Dr Eric WM Lee

Assistant Head of Architectural Engineering

Department of Architecture and Civil Engineering

City University of Hong Kong

Main Features of QRA

- Provide a numerical measure for the risk;
- Assist in an quantitative evaluation of the risk control measures;
- Enable a comparison of the effectiveness of different risk control strategies;
- Demonstrate the performance of the risk control measures and risk targets being achieved and maintained.

Content of this lecture

- Fault Tree Analysis (FTA)
- Event Tree Analysis (ETA)

Fault Tree Analysis

Introduction

- Fault trees use deductive logic in that it transfers the general problem to the specific causes.
- Two kinds of symbols are used in a fault tree:
  - Logic symbols
  - Event symbols
- Many symbols and styles, we stay with the simple ones here.
- Fault tree is based on probability theory in solving Boolean algebra.
- The graphical model can compute failure probabilities and system importance measures.

Review of Boolean Algebra

- A Boolean operator can be completely described using a truth table.
- The truth table for the Boolean operators AND and OR are shown at the right.
- The AND operator is also known as a Boolean product. The OR operator is the Boolean sum.

X AND Y

X	Y	XY
0	0	0
0	1	0
1	0	0
1	1	1

X OR Y

X	Y	X+Y
0	0	0
0	1	1
1	0	1
1	1	1

Review of Boolean Algebra

- The truth table for the Boolean NOT operator is shown at the right.
- The NOT operation is most often designated by an overbar. It is sometimes indicated by a prime mark (‘) or an “elbow” (¬).

NOT x

x	$\overline{x}$
0	1
1	0

Laws in Boolean Algebra

AB = A and B A+B = A or B

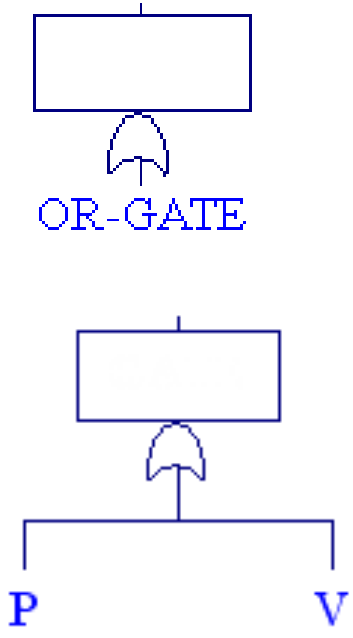
Identity Name	AND Form	OR Form
Identity Law	$1x = x$	$0 + x = x$
Null Law	$0x = 0$	$1 + x = 1$
Idempotent Law	$xx = x$	$x + x = x$
Inverse Law	$x\bar{x} = 0$	$x + \bar{x} = 1$

Laws in Boolean Algebra

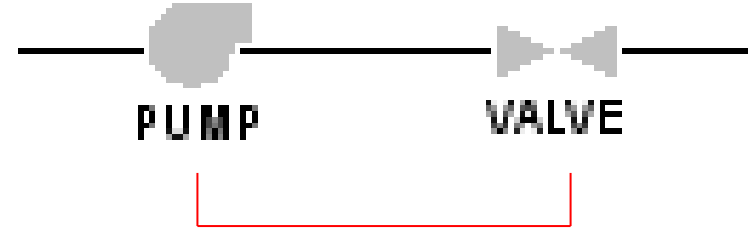
Identity Name	AND Form	OR Form
Commutative Law	$xy = yx$	$x+y = y+x$
Associative Law	$(xy)z = x(yz)$	$(x+y)+z = x + (y+z)$
Distributive Law	$x+yz = (x+y)(x+z)$	$x(y+z) = xy+xz$

Identity Name	AND Form	OR Form
Absorption Law	$x(x+y) = x$	$x + xy = x$
DeMorgan's Law	$\overline{(xy)} = \bar{x} + \bar{y}$	$\overline{(x+y)} = \bar{x}\bar{y}$
Double Complement Law	$\overline{(\bar{x})} = x$	

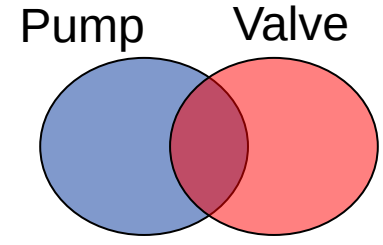
NOTATION – OR Gate



Series system

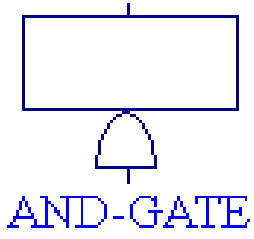


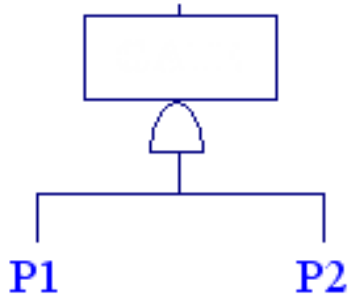
System fails when either component fails



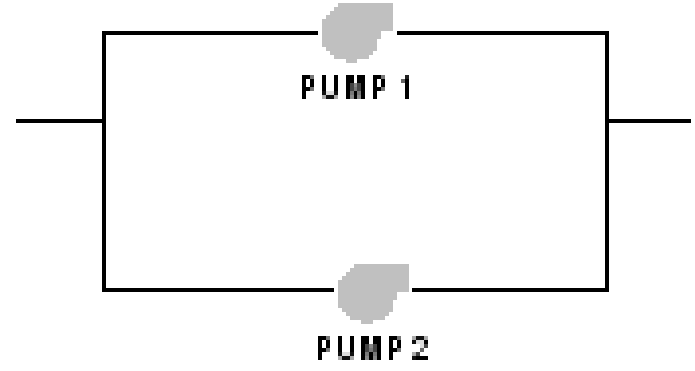
P(system failure) =
P(pump failure ∪ valve failure)

NOTATION – AND Gate

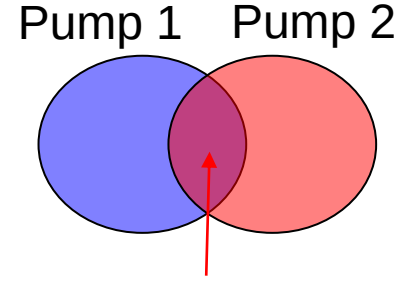




Parallel system

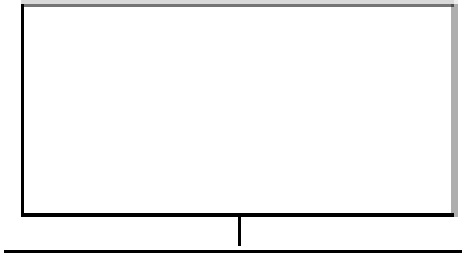


System fails when both components fail
(with one-out-of-two success criterion)

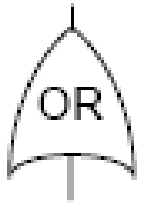


P(system failure) =
P(pump 1 failure ∩ pump 2 failure)

Fault Tree Symbols – Logic Symbols



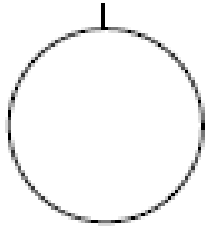
TOP Event – forseeable, undesirable event, toward which all fault tree logic paths flow,or **Intermediate event** – describing a system state produced by antecedent events.



"Or" Gate – produces output if any input exists. Any input, individual, must be (1) necessary and (2) sufficient to cause the output event.

Most Fault Tree Analyses can be carried out using only these four symbols.

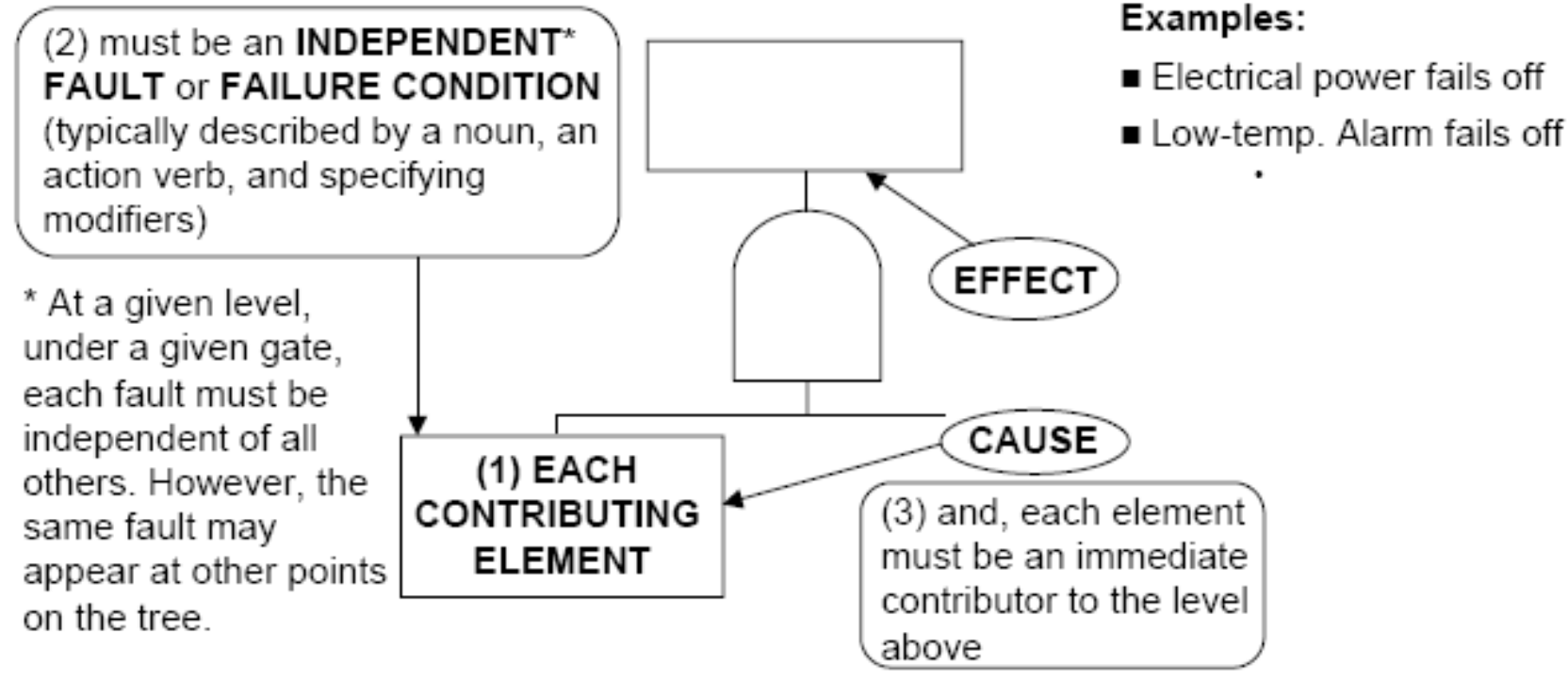
"And" Gate – produces output if all inputs co-exist. All inputs, individually must be (1) necessary and (2) sufficient to cause the output event



Basic Event – Initiating fault/failure, not developed further. (Called "Leaf," "Initiator," or "Basic.") The Basic Event marks the limit of resolution of the analysis.

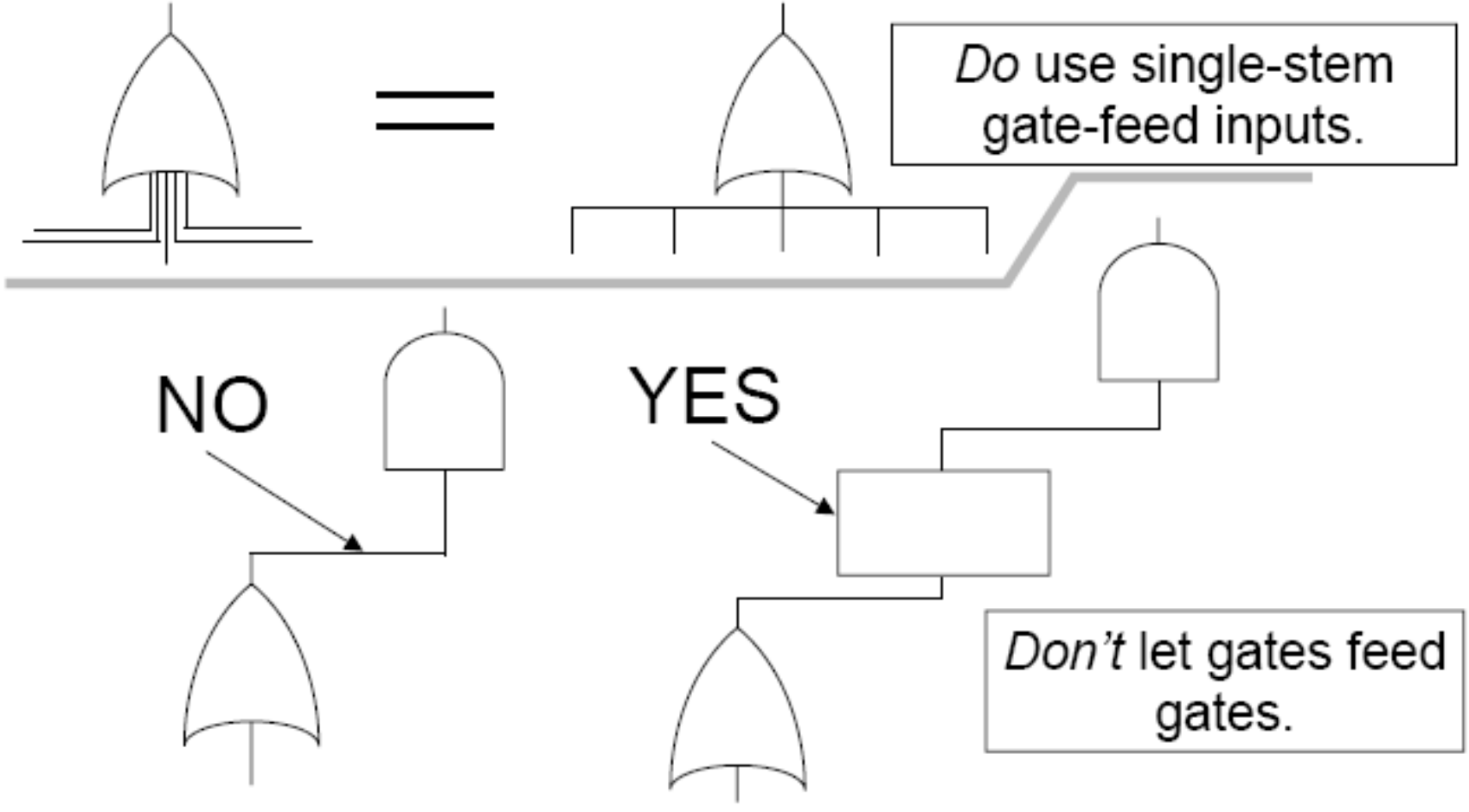
Events and **Gates** are **not** component parts of the system being analyzed. They are symbols representing the logic of the analysis. They are bi-modal. They function flawlessly.

Fault Tree Symbols – Event Symbols

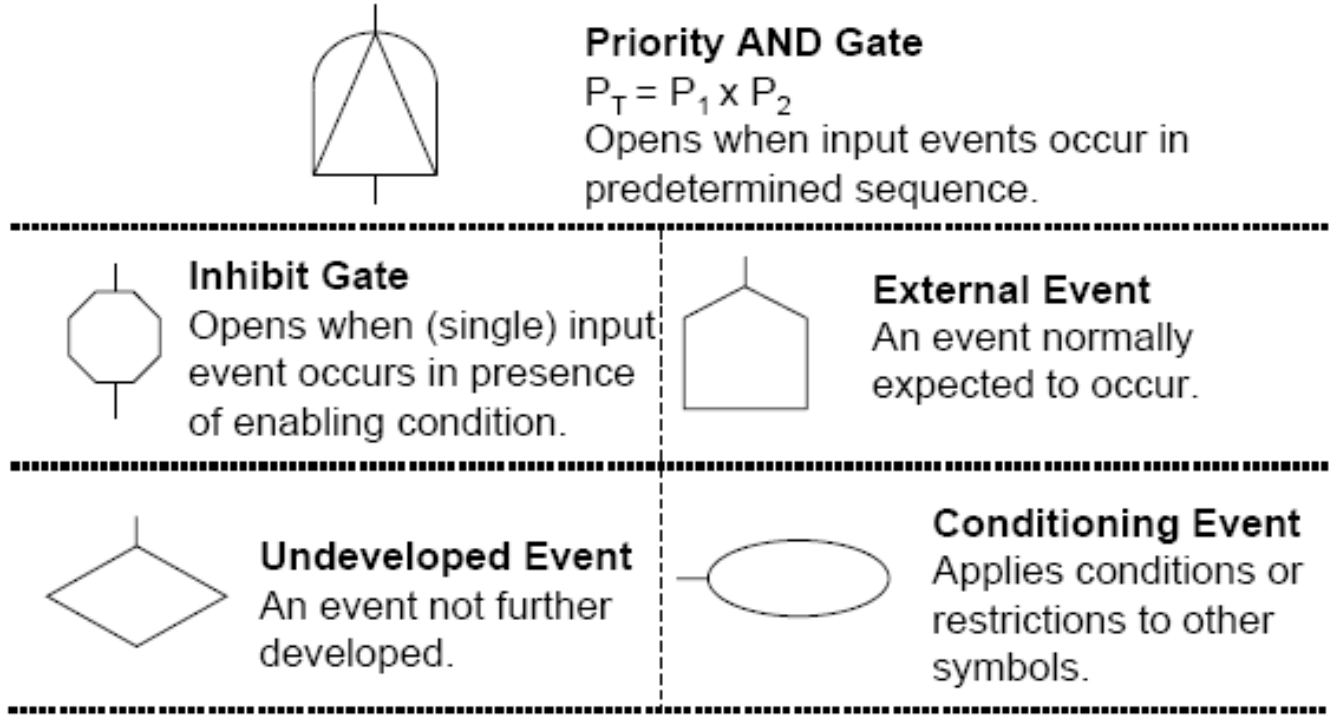


NOTE: As a **group** under an AND gate, and **individually** under an OR gate, contributing elements must be both **necessary** and **sufficient** to serve as **immediate** cause for the output event.

Fault Tree Symbols – Event Symbols



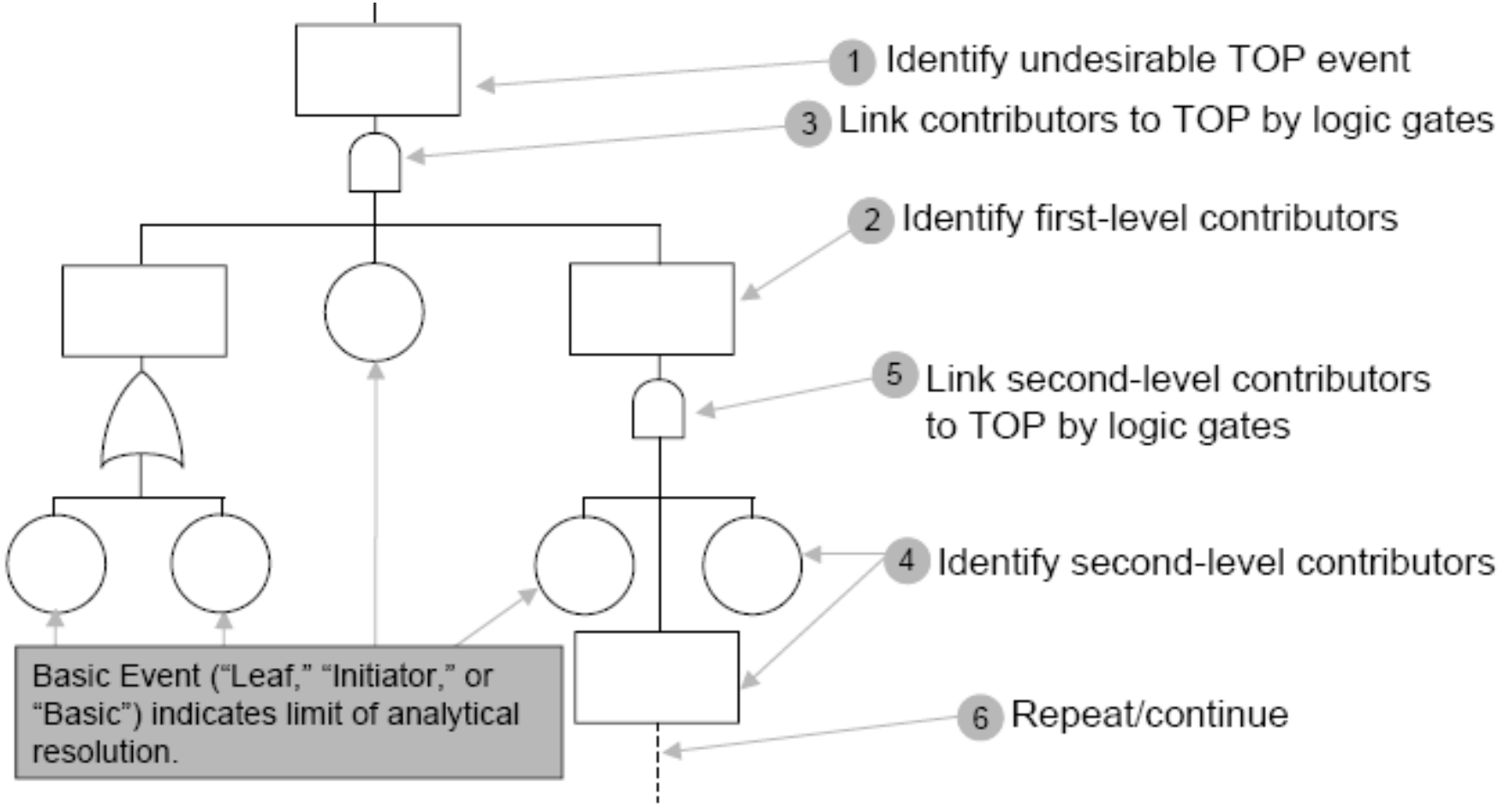
Fault Tree Symbols – More Symbols…



Fault Tree Construction

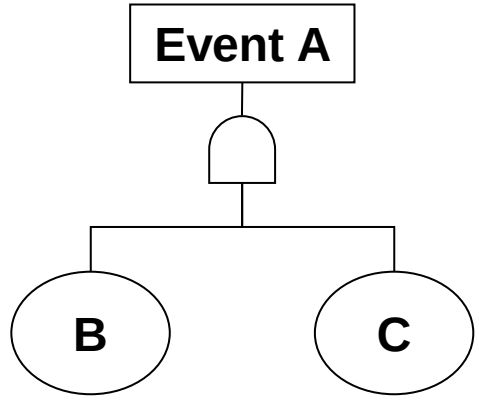
- Identify the Undesired Top Event. A different tree is required for each unique Top Event
- Start with Top Event and follow through scenario to systematically identify event initiators
- Separate tree into functional level, system level, subsystem level, component level, fault level, etc.
- Bottom of the tree are basic events or developed events
- Constructing the logic
- Identify and sketch the Intermediate Events to develop logical branches
- Spotting/correcting some common errors
- Can be qualitative or quantitative, adding quantitative data if needed

Fault Tree Construction

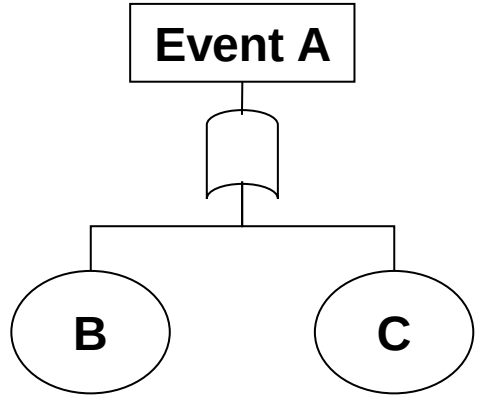


Fault Tree Structure

- **Event A occurs because of Event B and Event C occur**

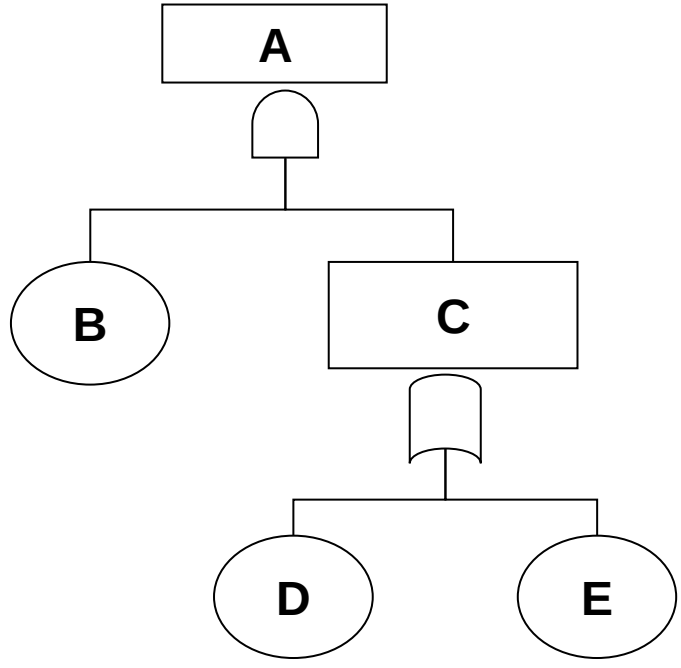


- **Event A occurs because of Event B or Event C occur**

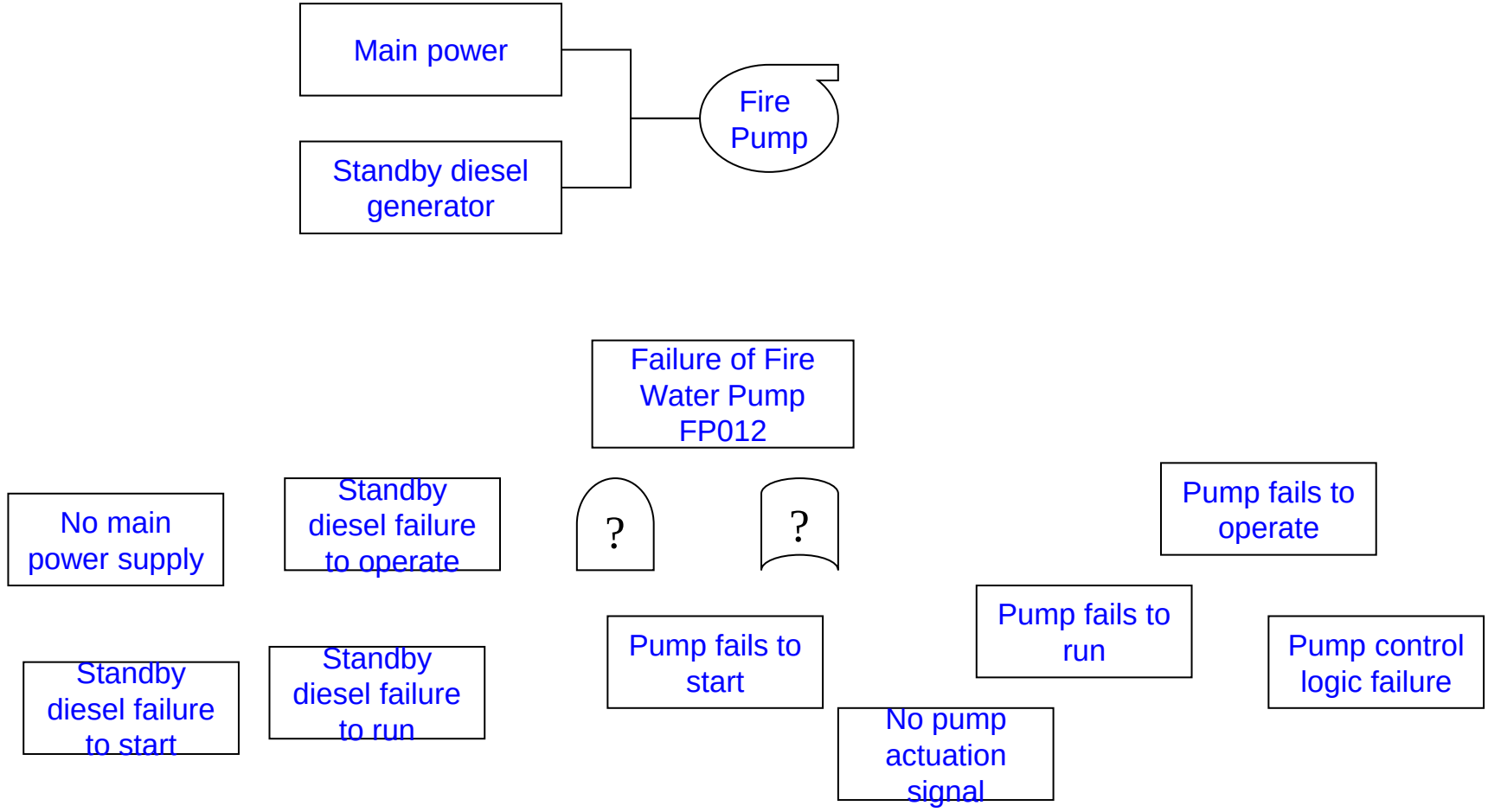


Fault Tree Structure

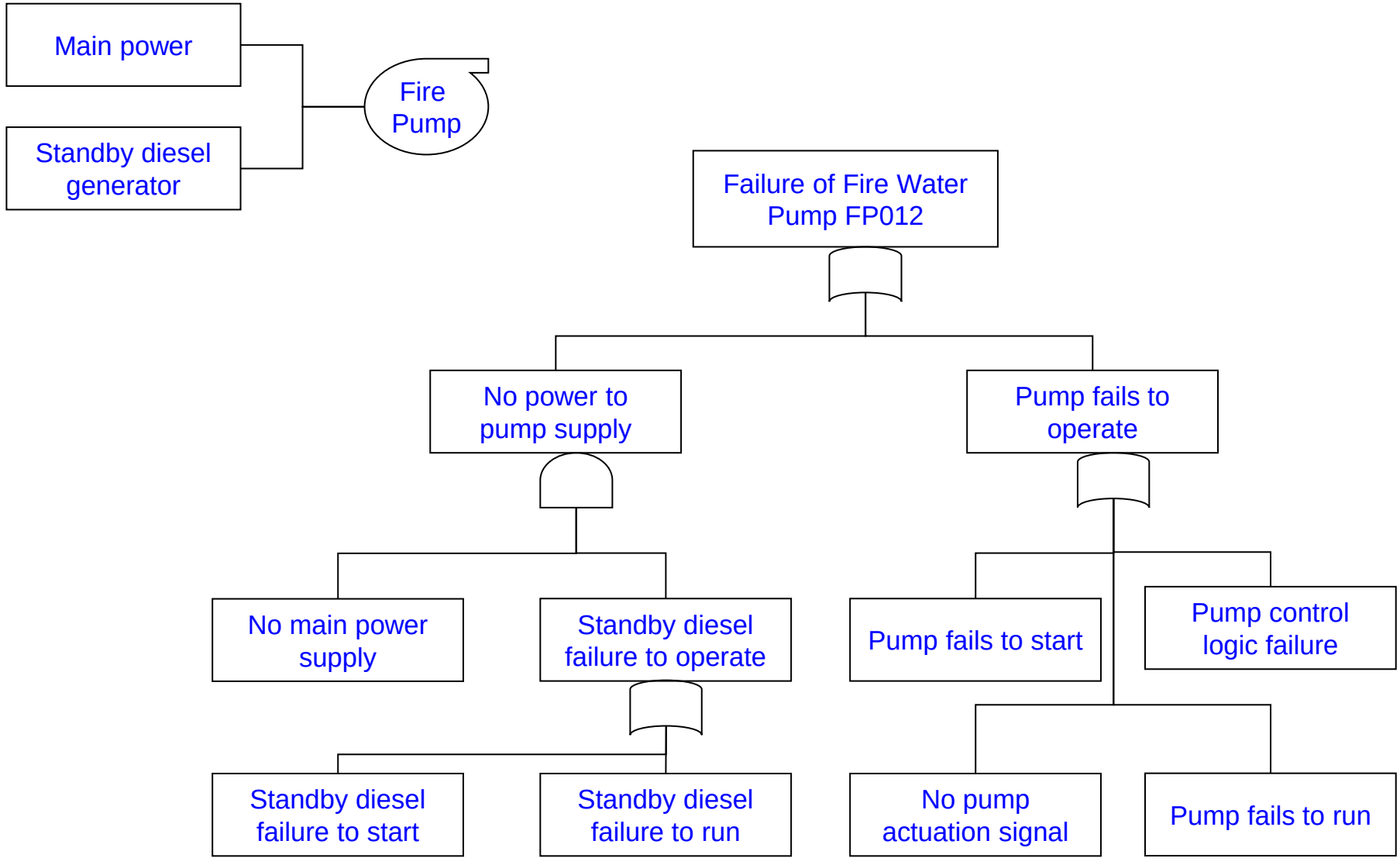
- **Event A occurs because of Event B and Event C occur**
- **Event C occurs because of Event D or Event E occur**



Example: Fire Pump Failure

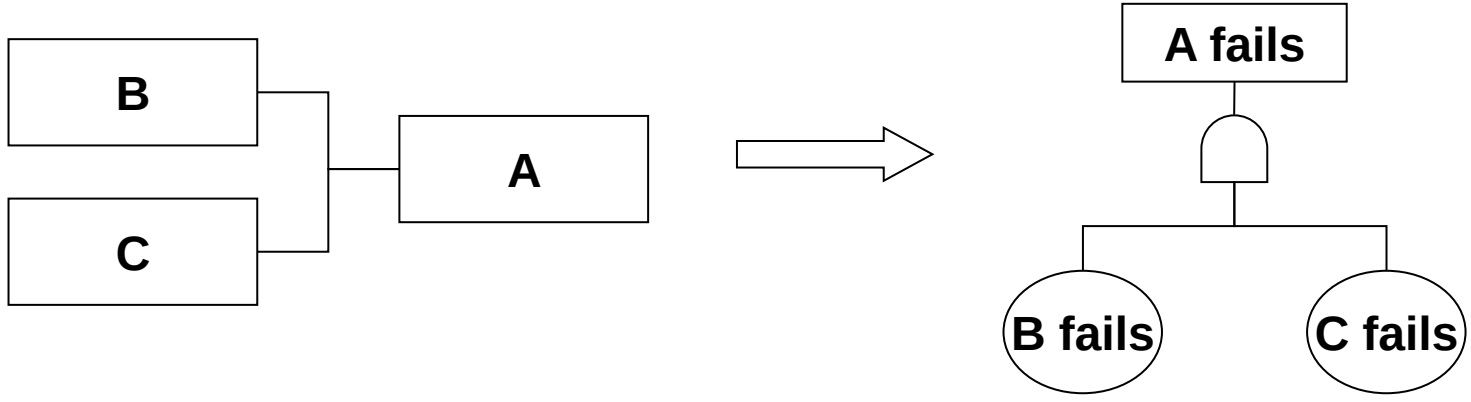


Example: Fire Pump Failure

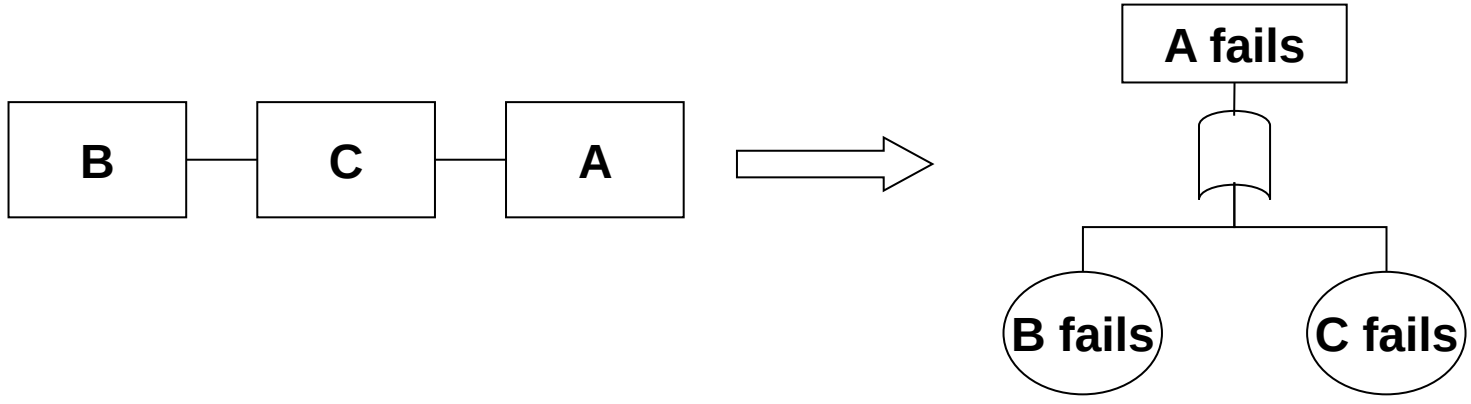


Fault Tree Structure

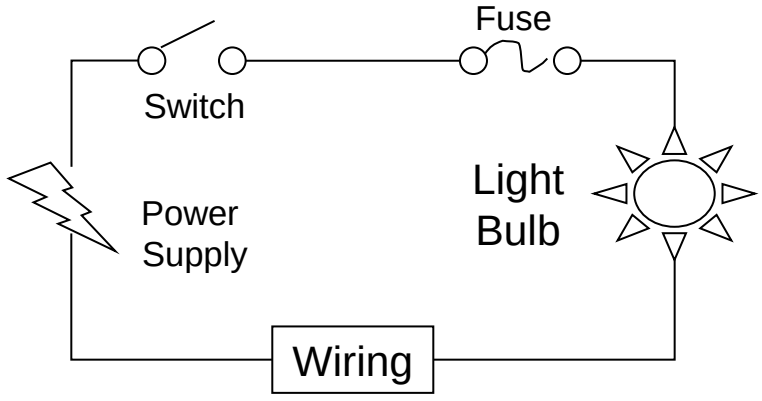
A parallel system (system works if either component works)



A series system (system works when all components work)

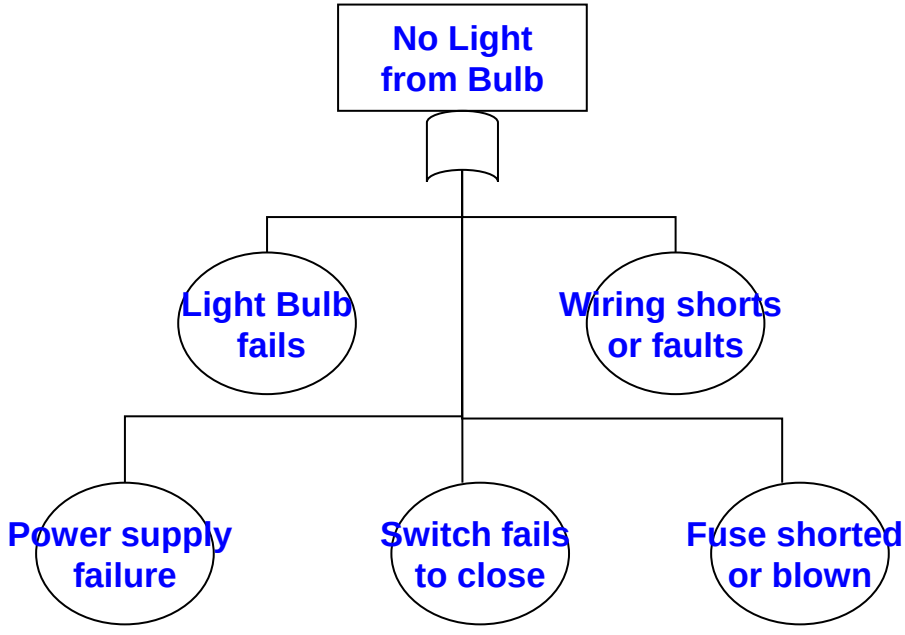


Example-No Light from Bulb

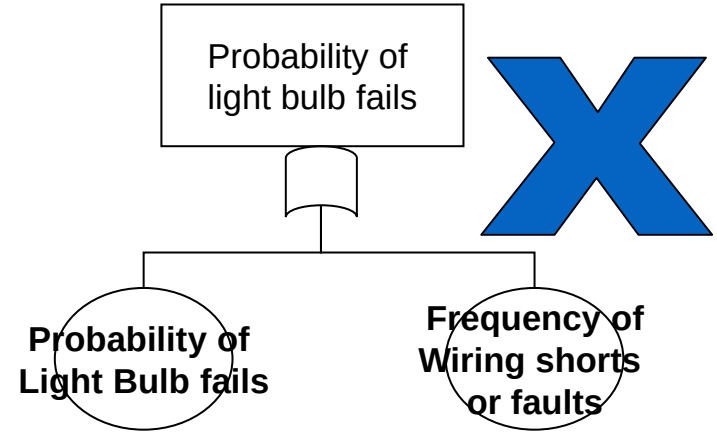


Develop fault event with top event: No light from bulb

Initial conditions: Switch closed
Not-considering events: failure external to system



Do not put down:

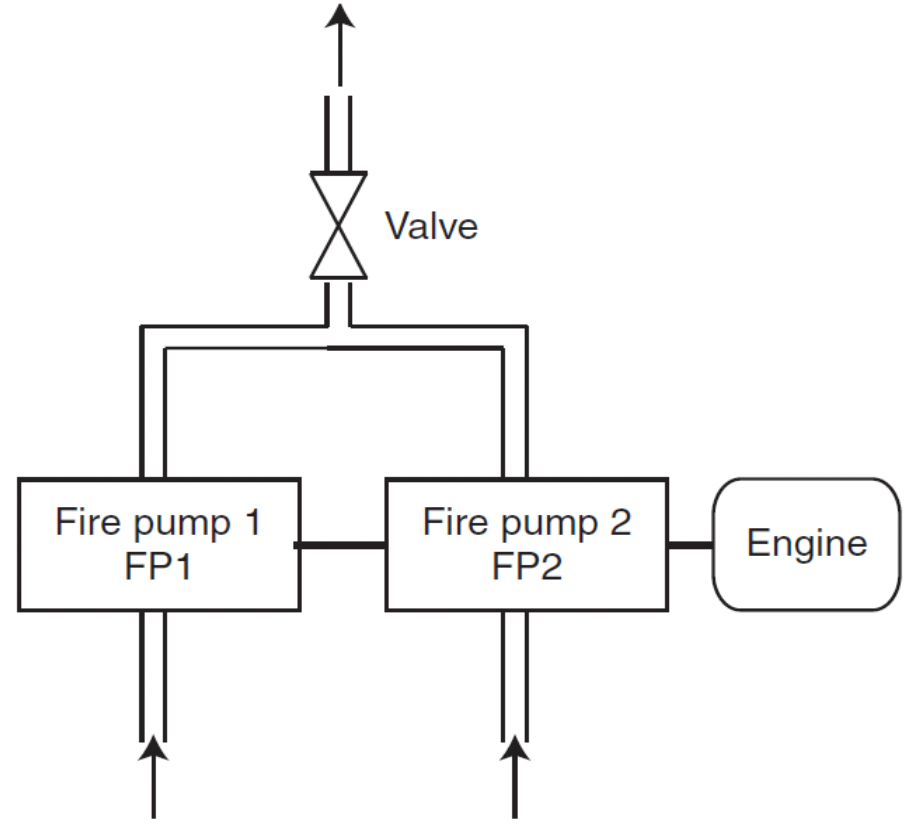


Example: Redundant Fire Pumps

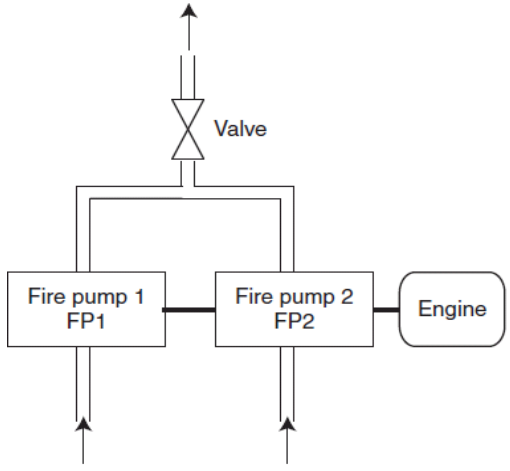
Develop fault tree for the top event: No water from fire water system.

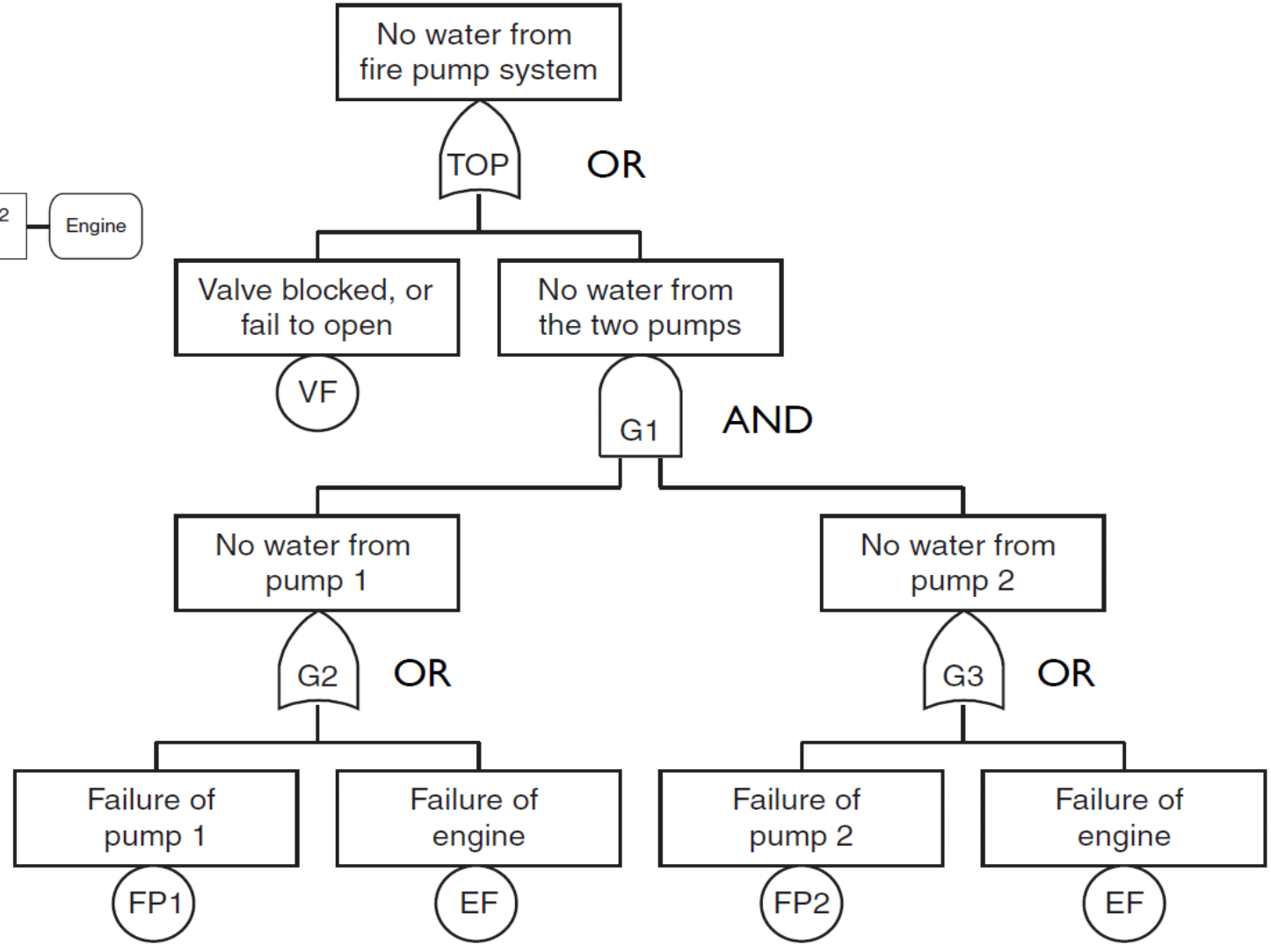
Causes for top event:

- VF: valve failure
- FP1: failure of fire pump 1
- FP2: failure of fire pump 2
- EF: failure of engine
- G1: no output from any of the fire pumps
- G2: no water from FP1
- G3: no water from FP2

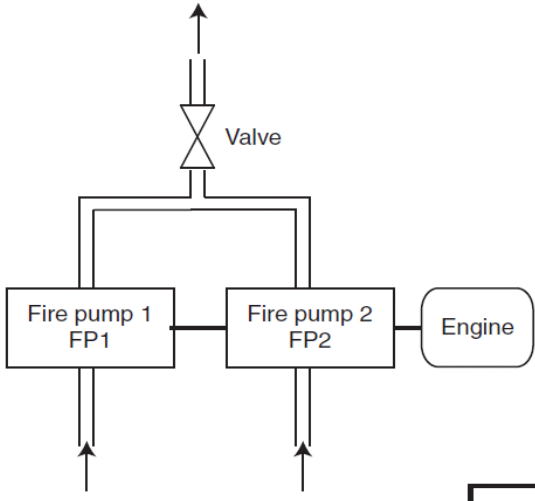


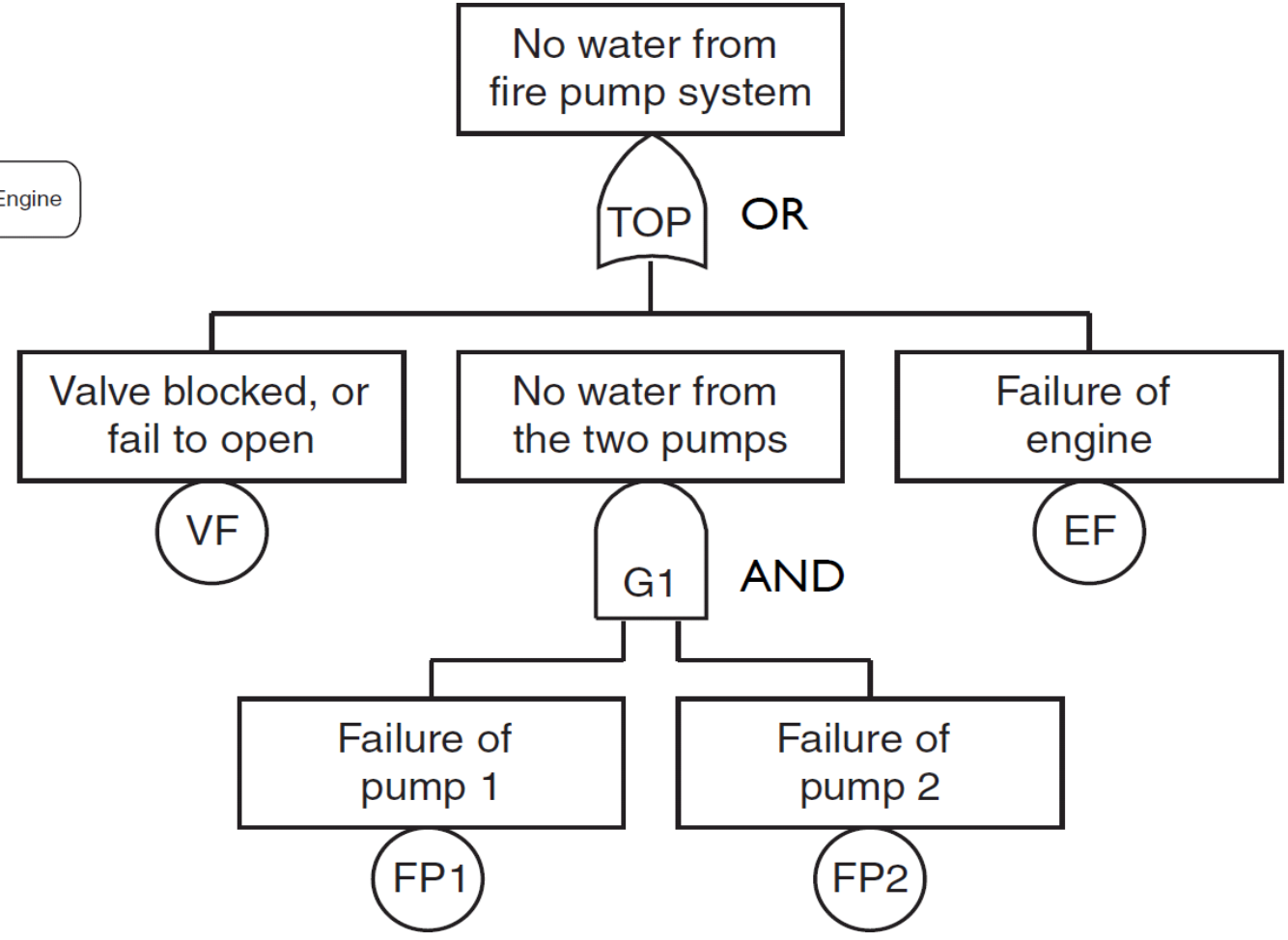
Example: Redundant Fire Pumps





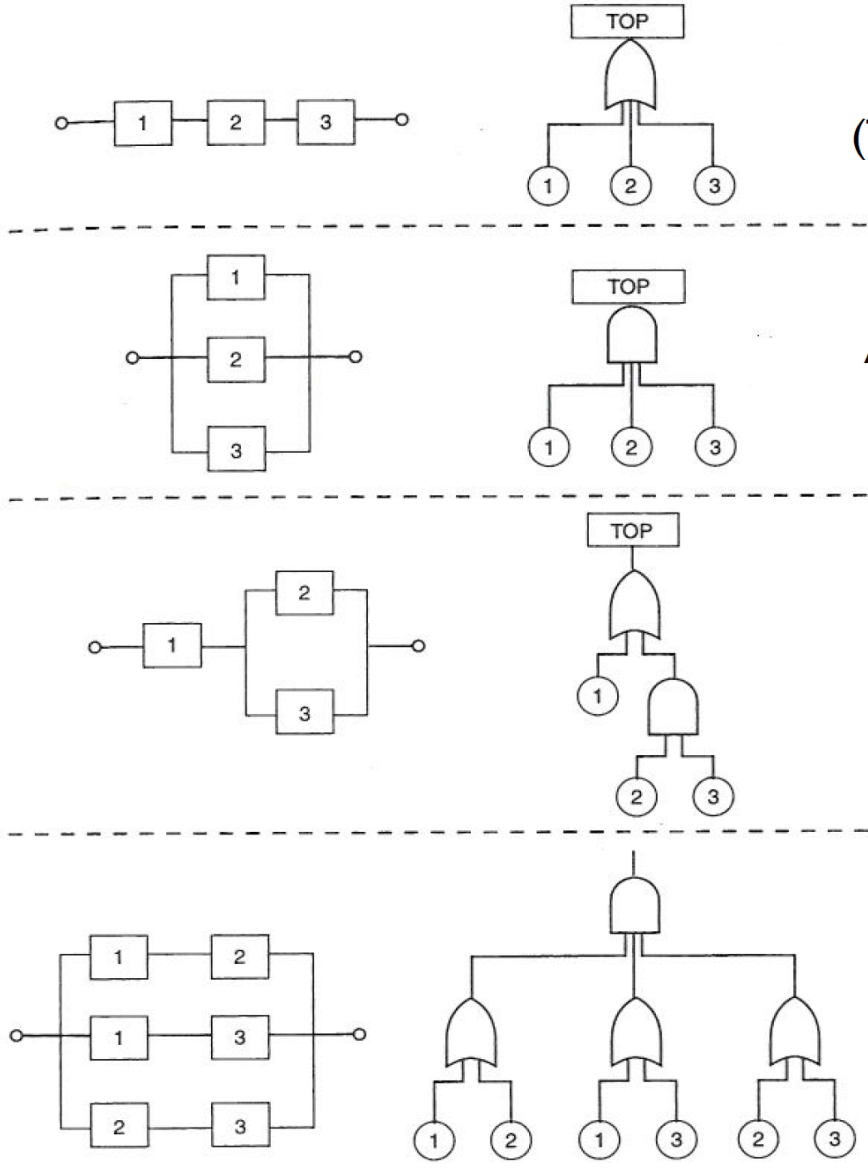
Example: Redundant Fire Pumps (simplified)





Much Simpler!

RBD to FTA Mapping

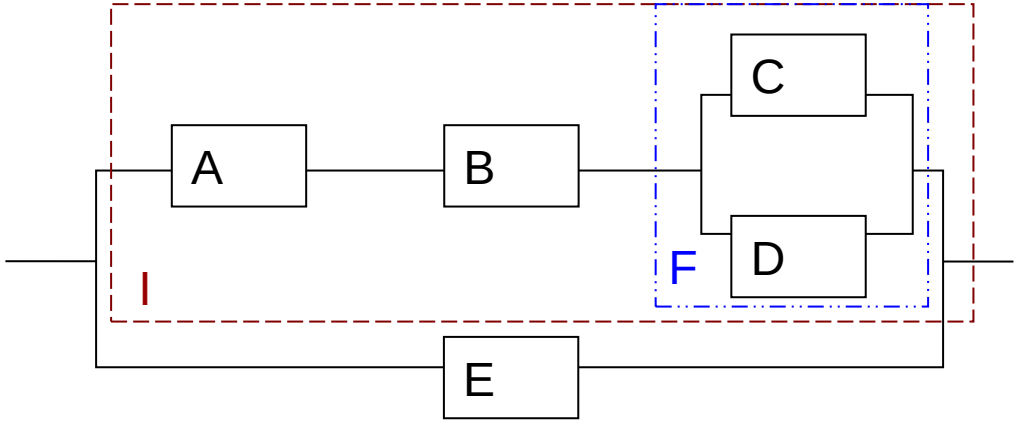


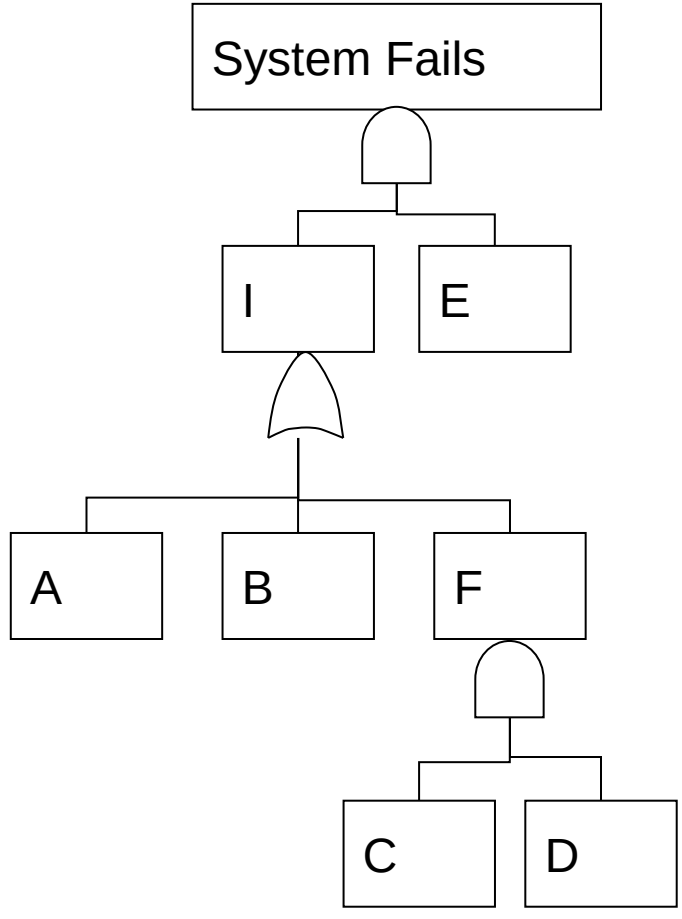
OR-gate = Series configuration
(TOP event occurs if at least one fails)

AND-gate = Parallel configuration
(TOP event occurs if all fail)

NOTE difference with when we were computing reliabilities! WHY?

Example





FTA: Qualitative Assessment

- In qualitative analysis, FTA is used to locate weak points and evaluate and improve system design.

- **Cut Sets:** A set of basic events whose simultaneous occurrence ensures that the TOP event occurs

- Qualitative assessment by investigating minimal cut sets which is a cut set with minimum number of events that can still cause the top event.

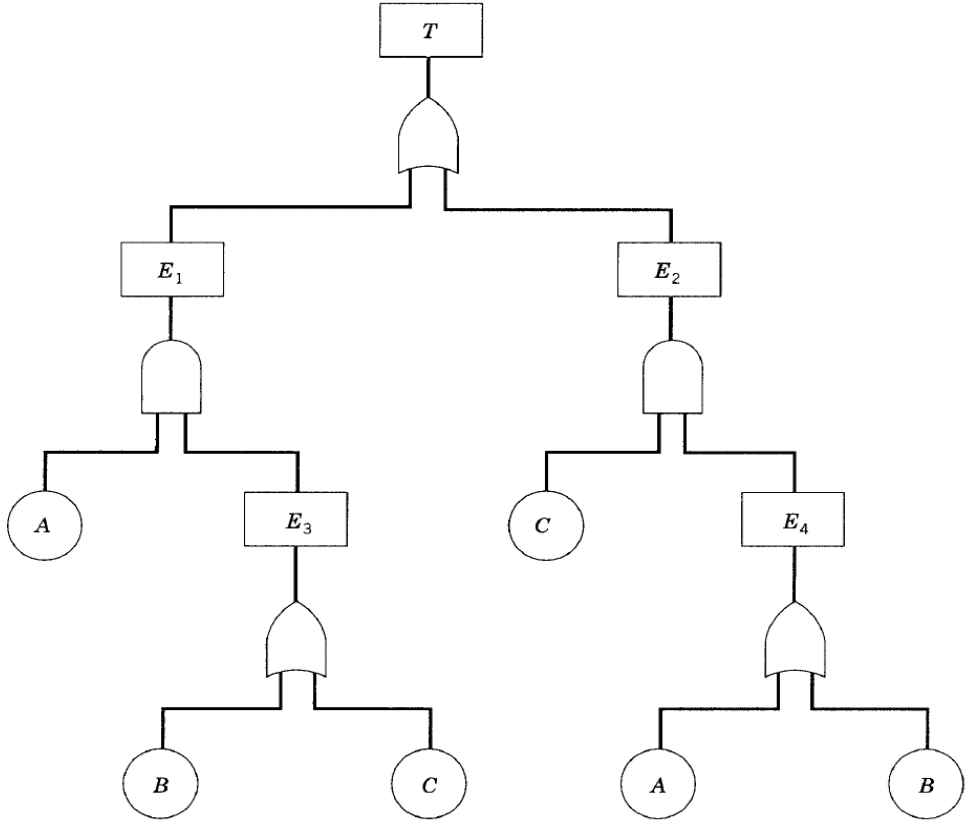
Minimum Cut Sets

- A minimum cut set is defined as the smallest combination (i.e. intersection) of primary failures which, if they all occur, will cause the top event to occur.

- If even one of the failures in the minimum cut set does not happen, the TOP event will not take place.

- A minimum cut set is one that does not contain within itself another cut set.

Find the Minimum Cut Set by Boolean Reduction

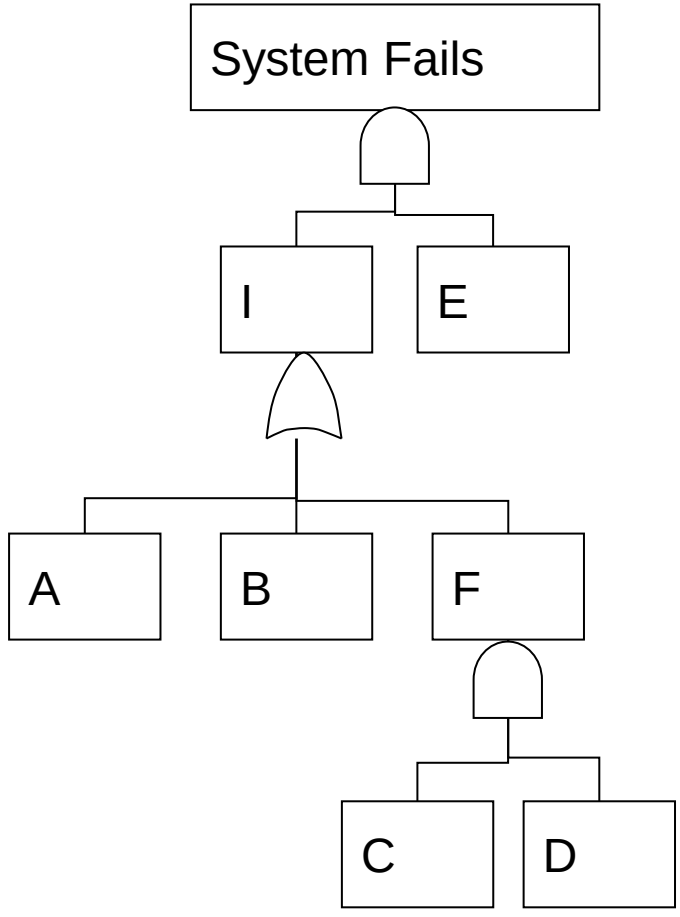


T = E1UE2
= (A∩E3) U (C∩E4)
= (A∩(BUC)) U (C∩(AUB))
= (A∩B) U (A∩C) U (C∩A) U (C∩B)
= (A∩B) U (A∩C) U (B∩C)

Cut Sets:
A∩B, A∩C, B∩C

Minimum Cut Set:
A∩B, A∩C, B∩C

Example - Finding the Minimum Cut Set



Suggested solution:

S = I∩E

= (AUBUF) ∩ E

= (A∩E) U (B∩E) U (F∩E)

= (A∩E) U (B∩E) U ((C∩D)∩E)

= (A∩E) U (B∩E) U (C∩D∩E)

Cut Set:

A∩E, B∩E, C∩D∩E

Minimum Cut Set:

A∩E, B∩E

FTA: Qualitative Assessment

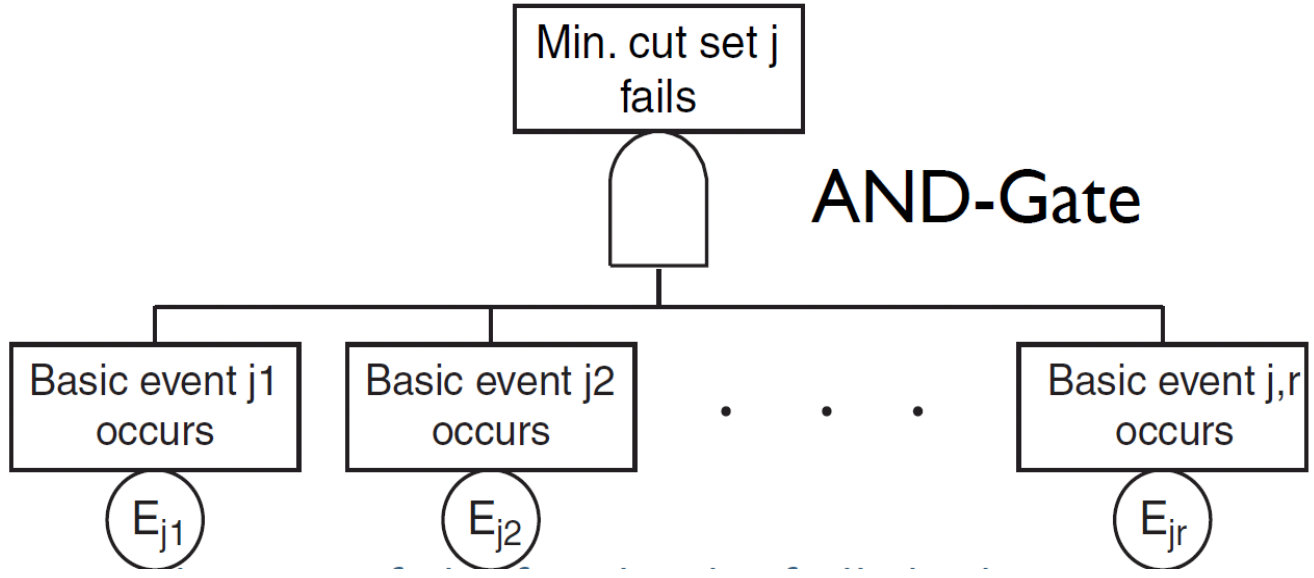
- Qualitatively evaluate and analyze the Cut Sets for design problems/concerns (i.e. root cause, design weak points, common cause problems);
- Lower order Cut Sets are more important;
- Component importance by number of times it appears in different Cut Sets;

FTA: Quantitative Assessment

- Quantitatively evaluate the probability of event occurrence;
- Process requires components failure rate;
- Probabilistic Risk Assessment (PRA)
- Quantitative measures for
  - Cut sets
  - Component criticality & Importance
  - Critical path ranking

Cut Set Assessment

- A minimal cut set fails if and only if all the basic events in the set fail at the same time



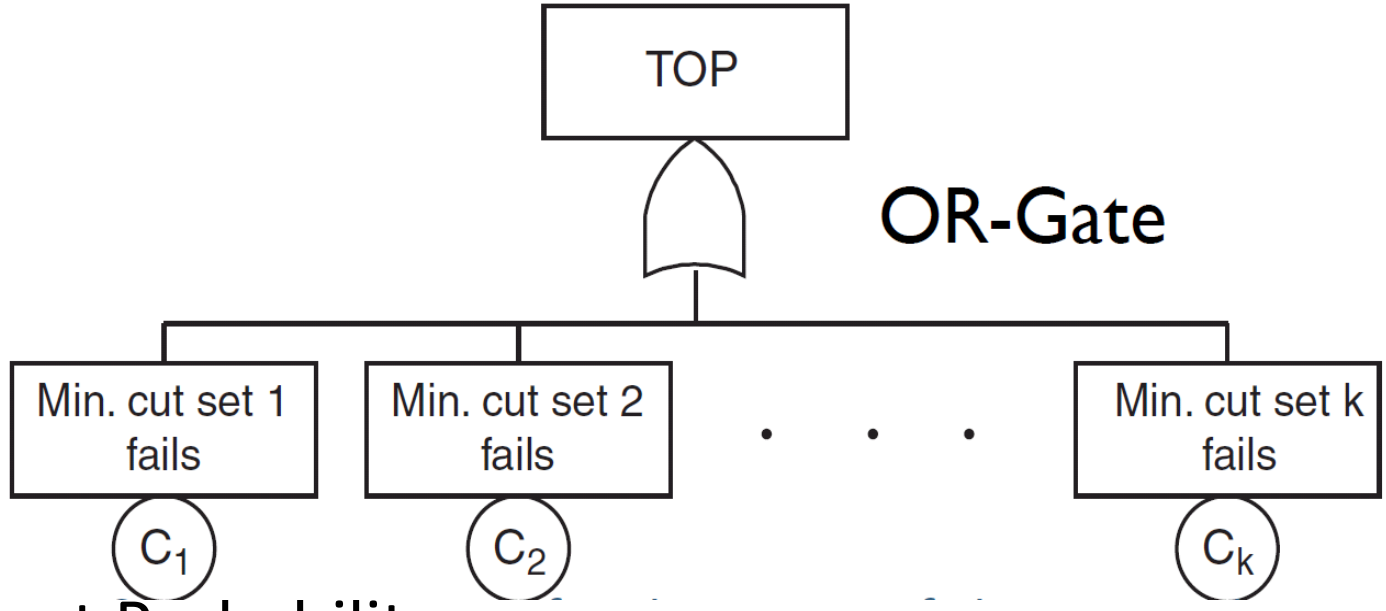
- Probability that cut set j fails at time t:

$$\tilde{Q}_j(t) = \prod_{i=1}^{r} q_{j,i}(t)$$

where all the r basic events in the minimal cut set j are independent

Top Event Probability

- The TOP event occurs if at least one of the minimal cut sets fails.

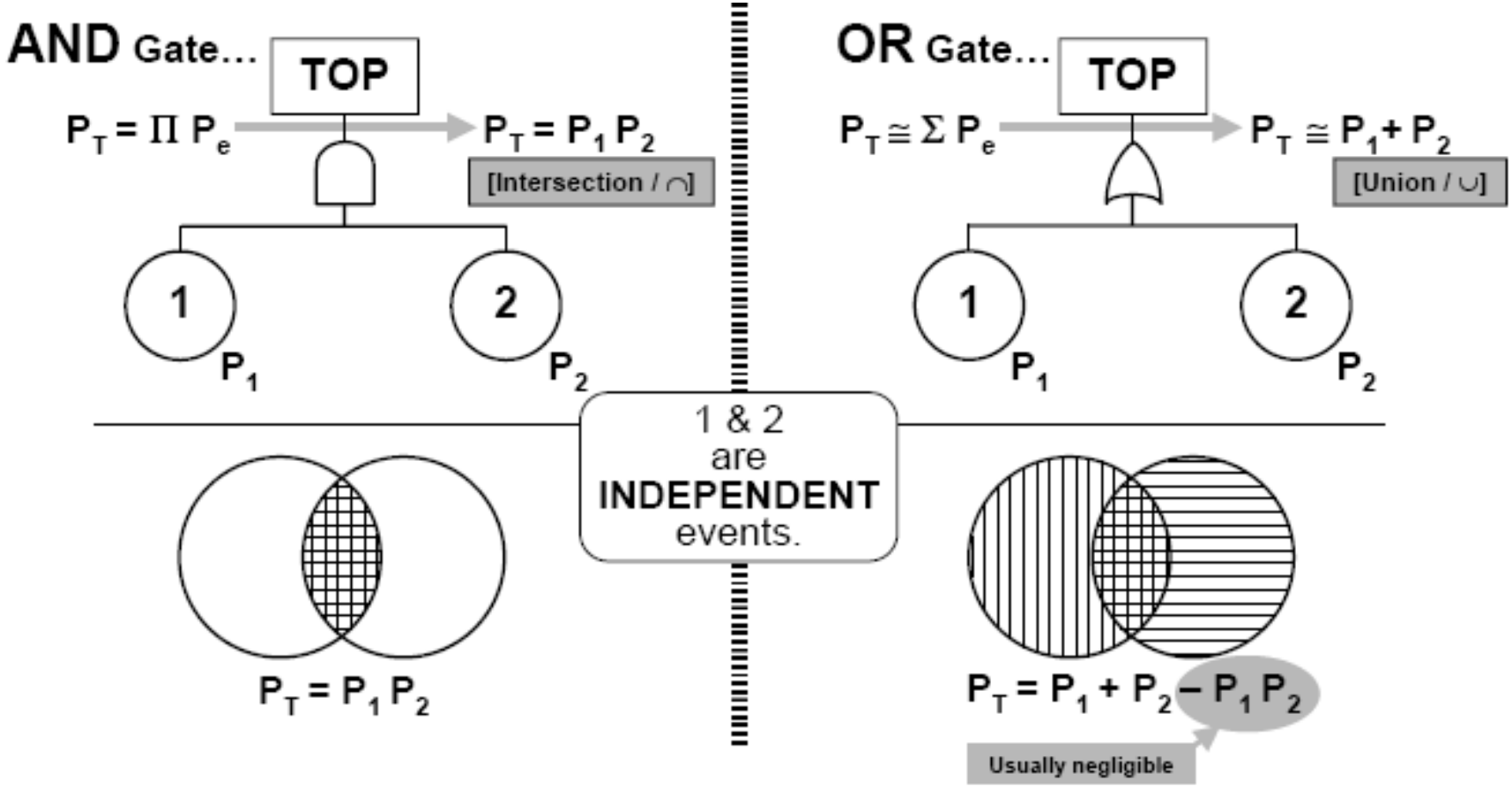


- TOP event Probability:

The inequality sign implies that the basic events are not always independent (same basic event can be a part of several cut sets.)

$$Q_0(t) \le 1 - \prod_{j=1}^{k} (1 - \tilde{Q}_j(t))$$

Fault Tree Calculations



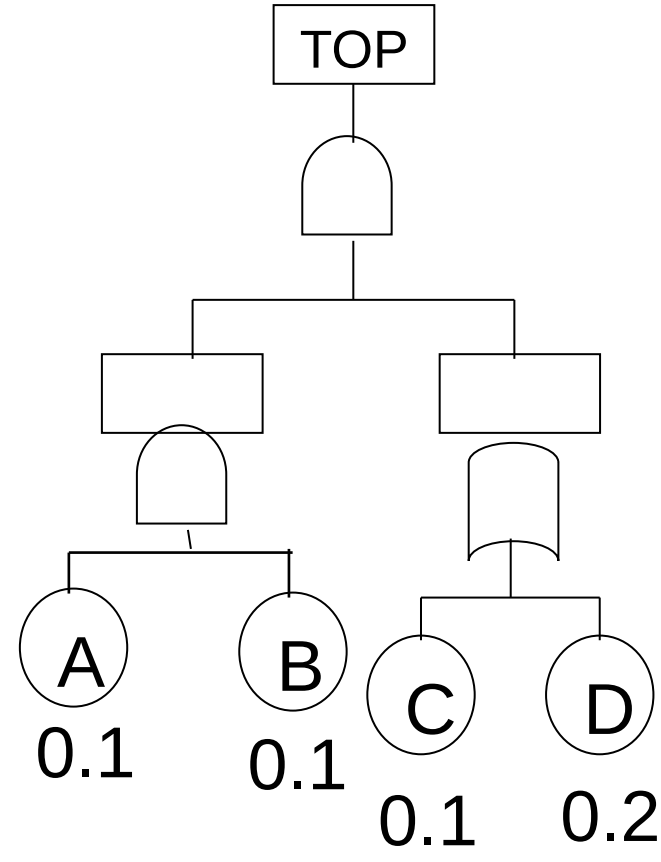
Fault Tree Calculations

Approximation:

- P(Top) ≈ P(A) x P(B) x [P(C) + P(D)]
- P(Top) ≈ 0.1x0.1x(0.1+0.2) = 0.003

Exact:

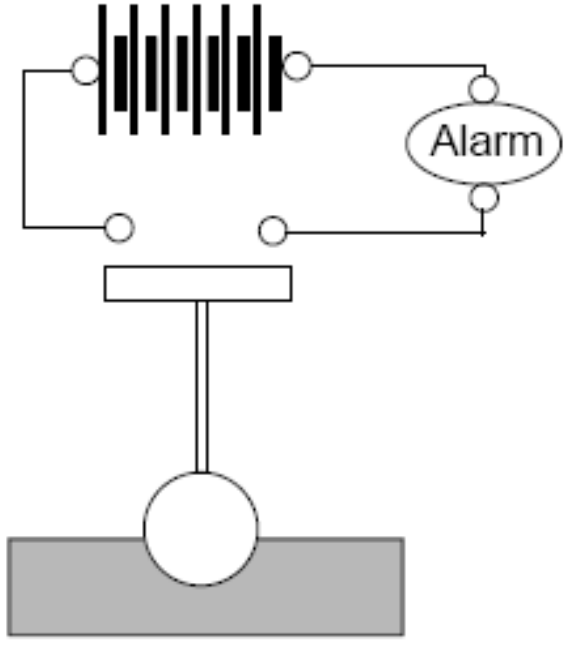
- P(Top) = P(A) x P(B) x [P(C) + P(D) – P(C)xP(D)]
- P(Top) ≈ 0.1x0.1x(0.1+0.2 – 0.1x0.2) = 0.0028



Rare-event approximation
(OK if the individual component failure probabilities are less than 0.1)

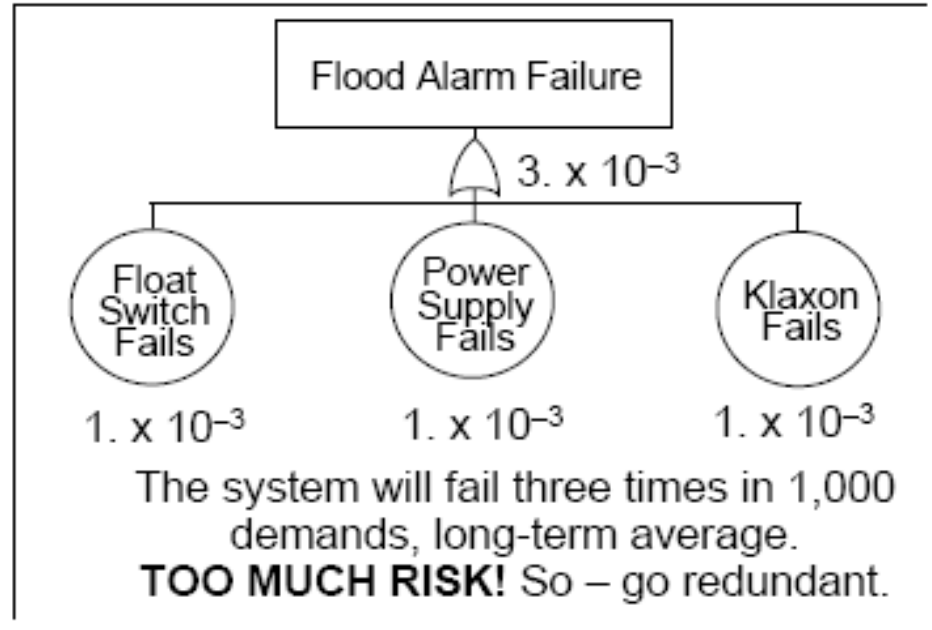
- If events are not independent, then we need conditional probabilities

A Flood Alarm System



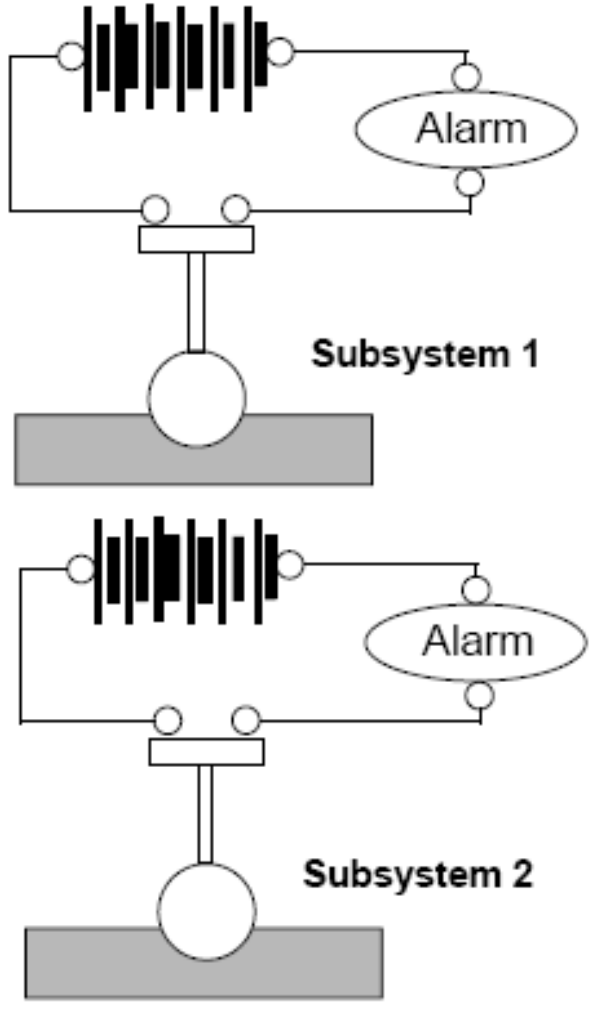
A system design goal is $P_F < 5 \times 10^{-6}$, per flood.

A subgrade compartment is protected against flooding by a simple alarm system. Each of the three components shown has a failure probability of 10^{-3} per demand. What is the probability of failure to alarm upon flooding?

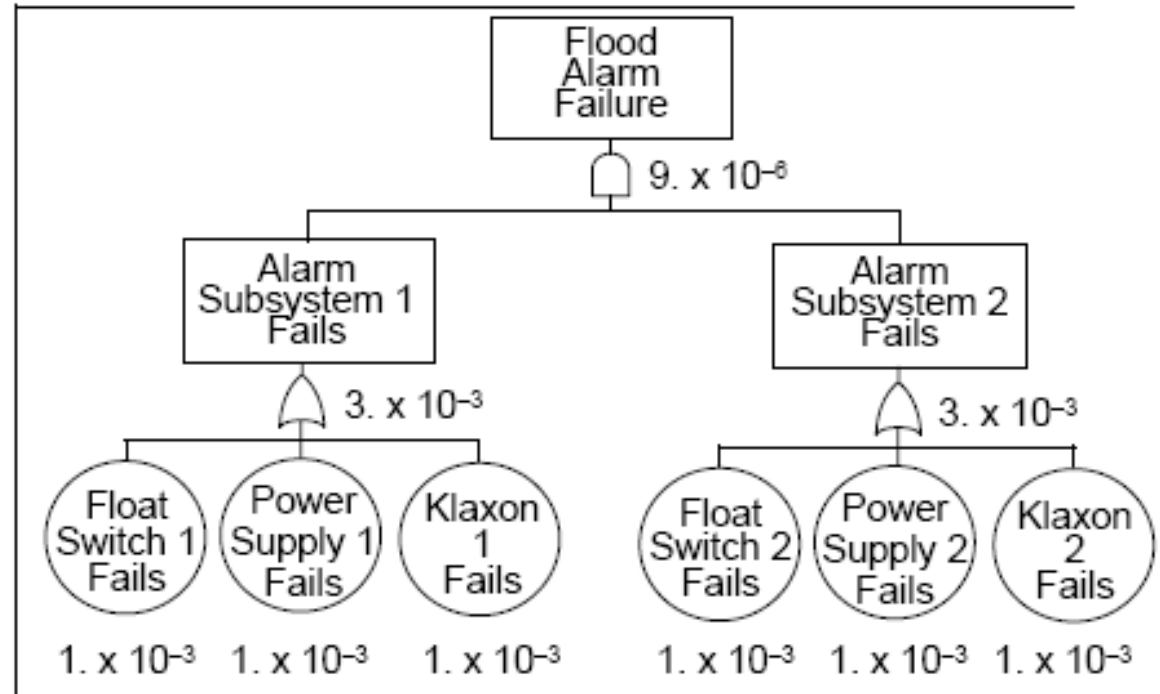


The system will fail three times in 1,000 demands, long-term average.
TOO MUCH RISK! So – go redundant.

A Flood Alarm System
Two System Redundancy

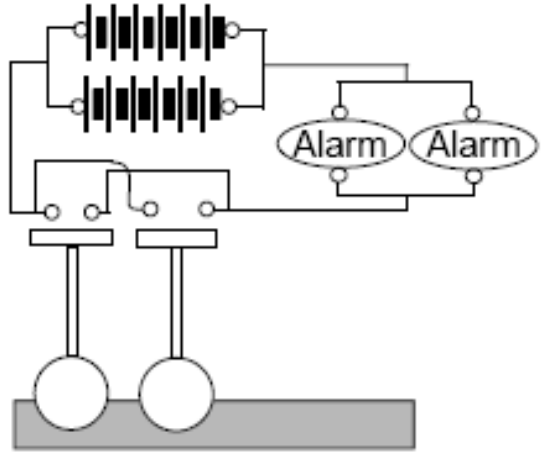


Two subsystems identical to the first system are now used. Ignoring common-cause effects, what now is the probability of failure to alarm upon flooding?

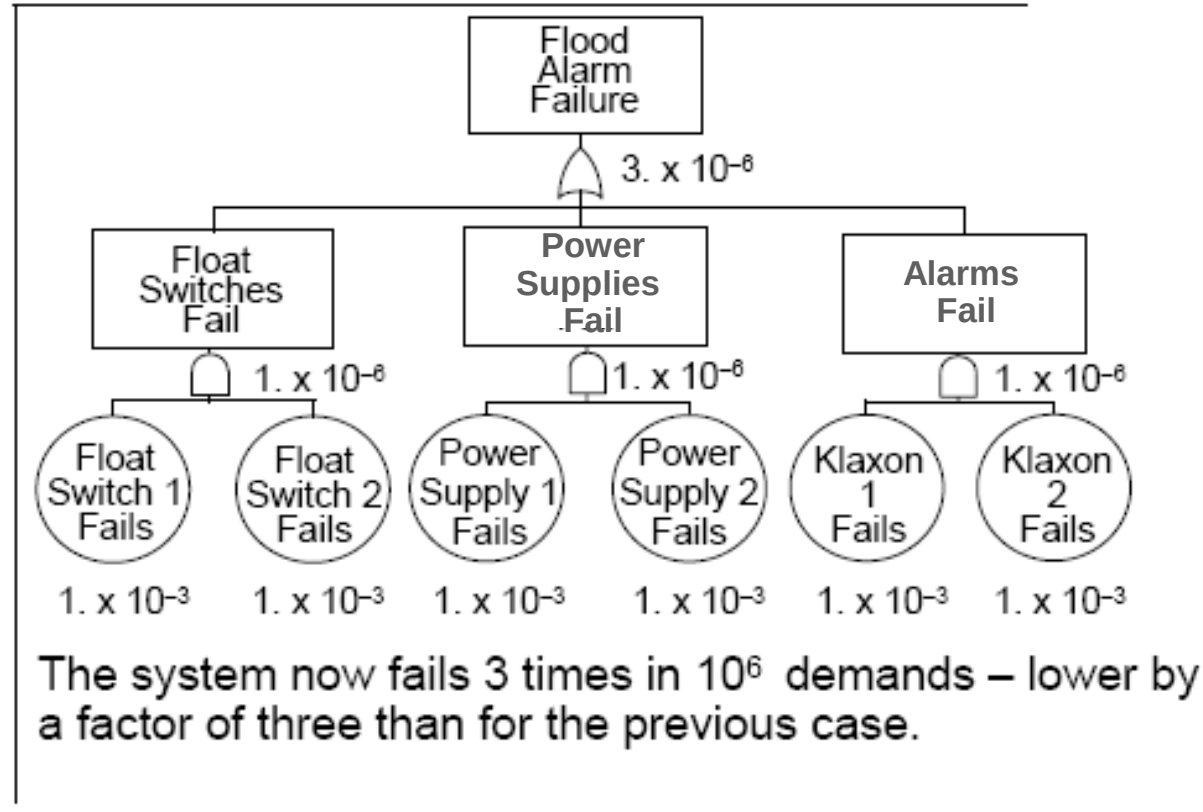


The system will fail 9 times in 10^6 demands…
STILL TOO HIGH! Can it be <u>further reduced</u>, perhaps using the same components?

A Flood Alarm System
Component Level Redundancy



Components themselves are made redundant, rather than the whole system. What **NOW** is the probability of alarm failure upon flooding?



The system now fails 3 times in 10^6 demands – lower by a factor of three than for the previous case.

Summary of FTA

- Find the root causes of a hazard or undesired event during design development in order that they can be eliminated or mitigated.
- Establish the root causes of a mishap that has occurred and prevent them from recurring.
- Identify the undesired event causal factor combinations and their relative probability.
- Determine high-risk fault paths and their mechanisms.
- Identify risk importance measures for components and fault events.
- Support a probabilistic risk assessment (PRA) of system designs.
- FTA is NOT (fully) suitable for modeling dynamic scenarios

Event Tree Analysis

Event Trees

- Use inductive logic to postulate and quantify accident scenarios or accident sequences which can generate from a single initiating event
- The initiating event might be a *failure* of the system or an *external event* to the system
- Each event following the initiating event is conditional on the occurrence of its precursor event;
- ETA identifies all possible accident scenarios
- Each branch of the event tree represents a separate outcome (event sequence)

Type of Event Tree

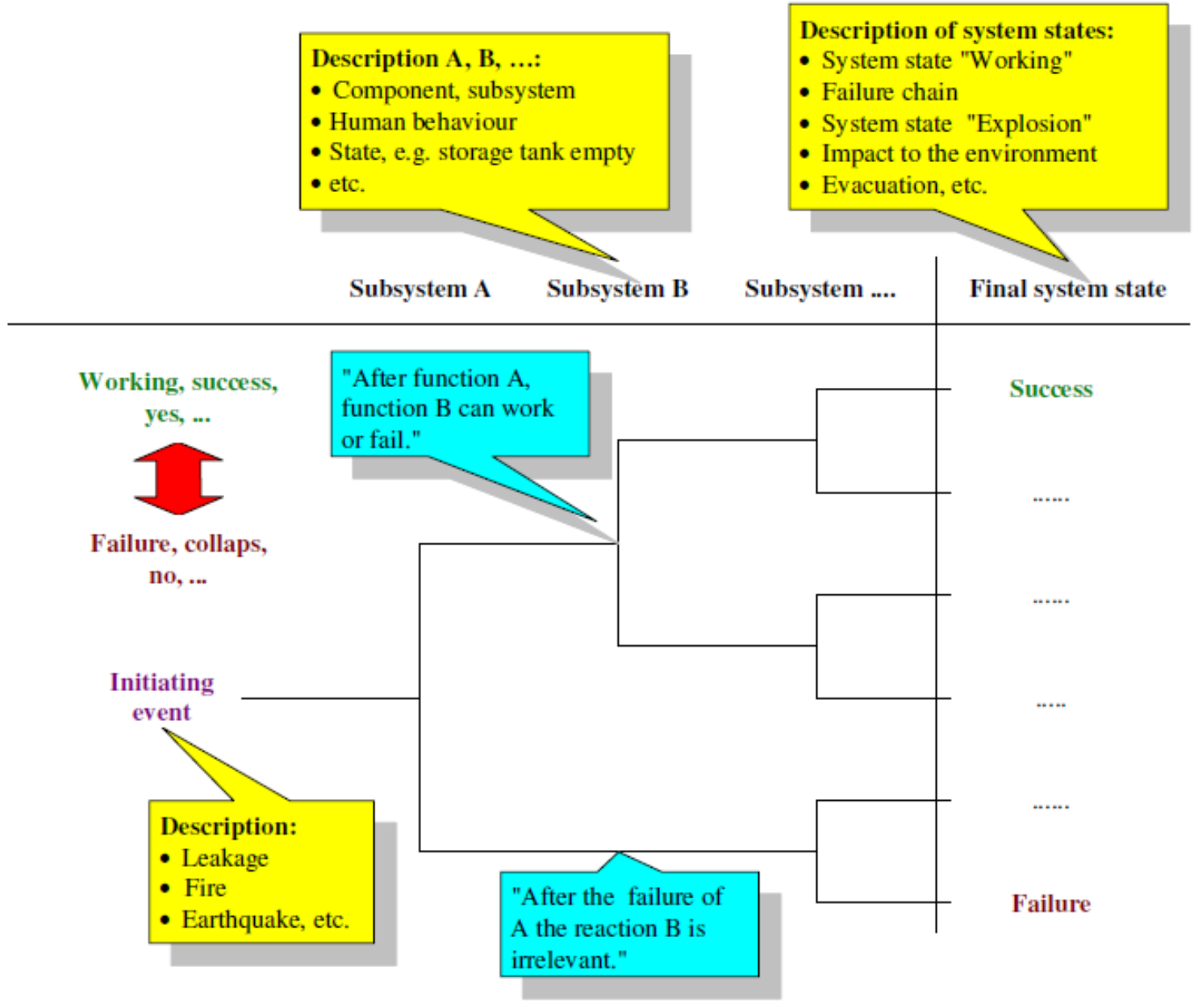
- System Event Tree
- Functional Event Tree
- Phenomenological Event Tree

Structure of Event Tree

- *An initiation event*:
  - The first significant deviation from a normal situation that may lead to unwanted consequences
  - Examples: gas leak, falling object, start of a fire, ...

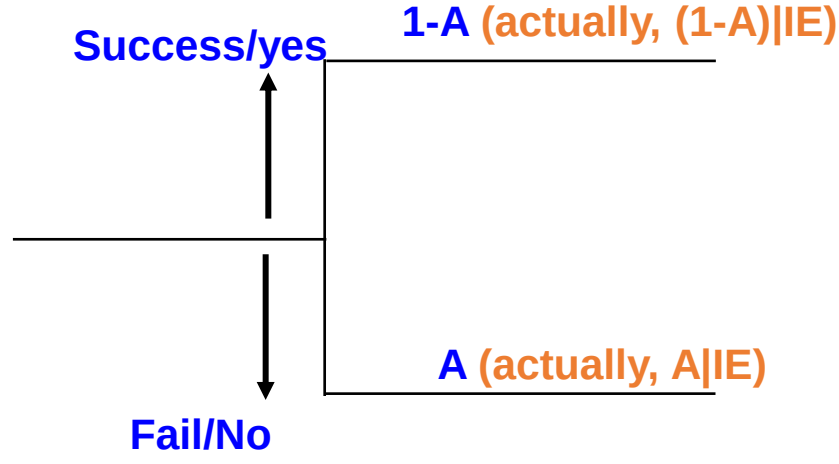
- *Barriers*:
  - Most well designed systems have barriers implemented to stop or reduce the consequences of potential accidental events
  - Also called: safety functions or protection layers

Structure of Event Tree



Event Tree

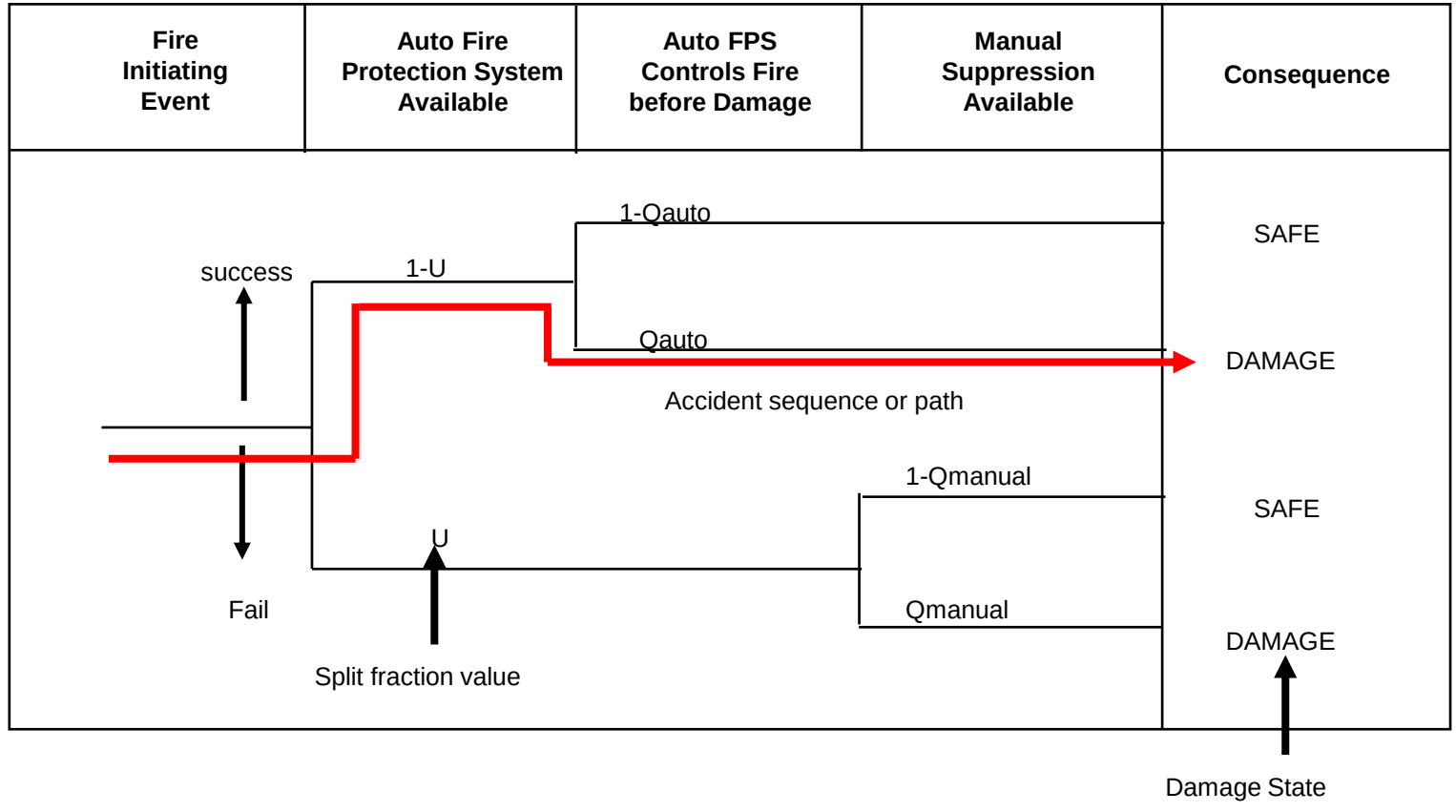
- Event headings are usually state or system, function of safety barriers, actions or events that can alter the course of the accident scenario; easier if you put key actions first
- Event tree and fault tree are inter-changeable in most cases



  - "A" is a probability called the "split fraction"
  - The sum of all split fractions coming out from a branch is 1

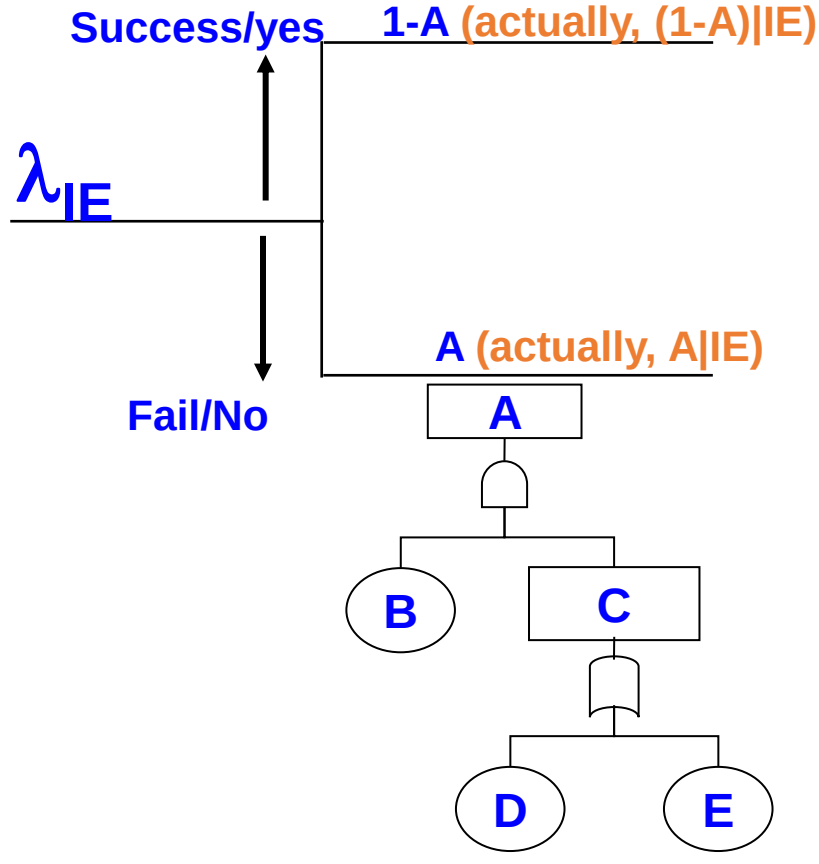
Event Tree Analysis

- Event tree heading may have more than 2 outcomes or branches.
- Start with an initiating event, not a damage state and follow through scenario to identify possible scenarios which need to be managed
- Most people confuse event tree with decision tree

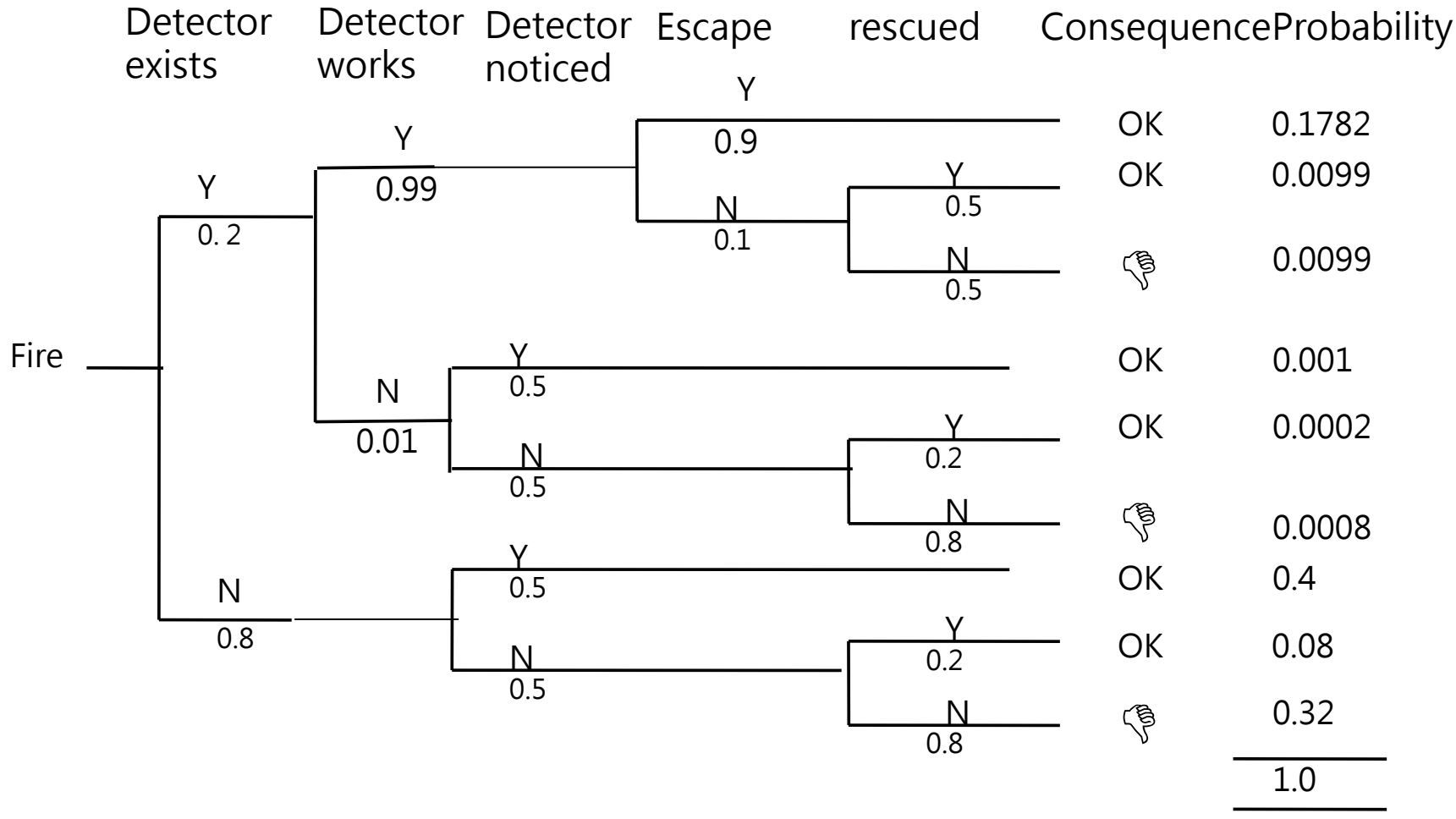


Event Tree Analysis

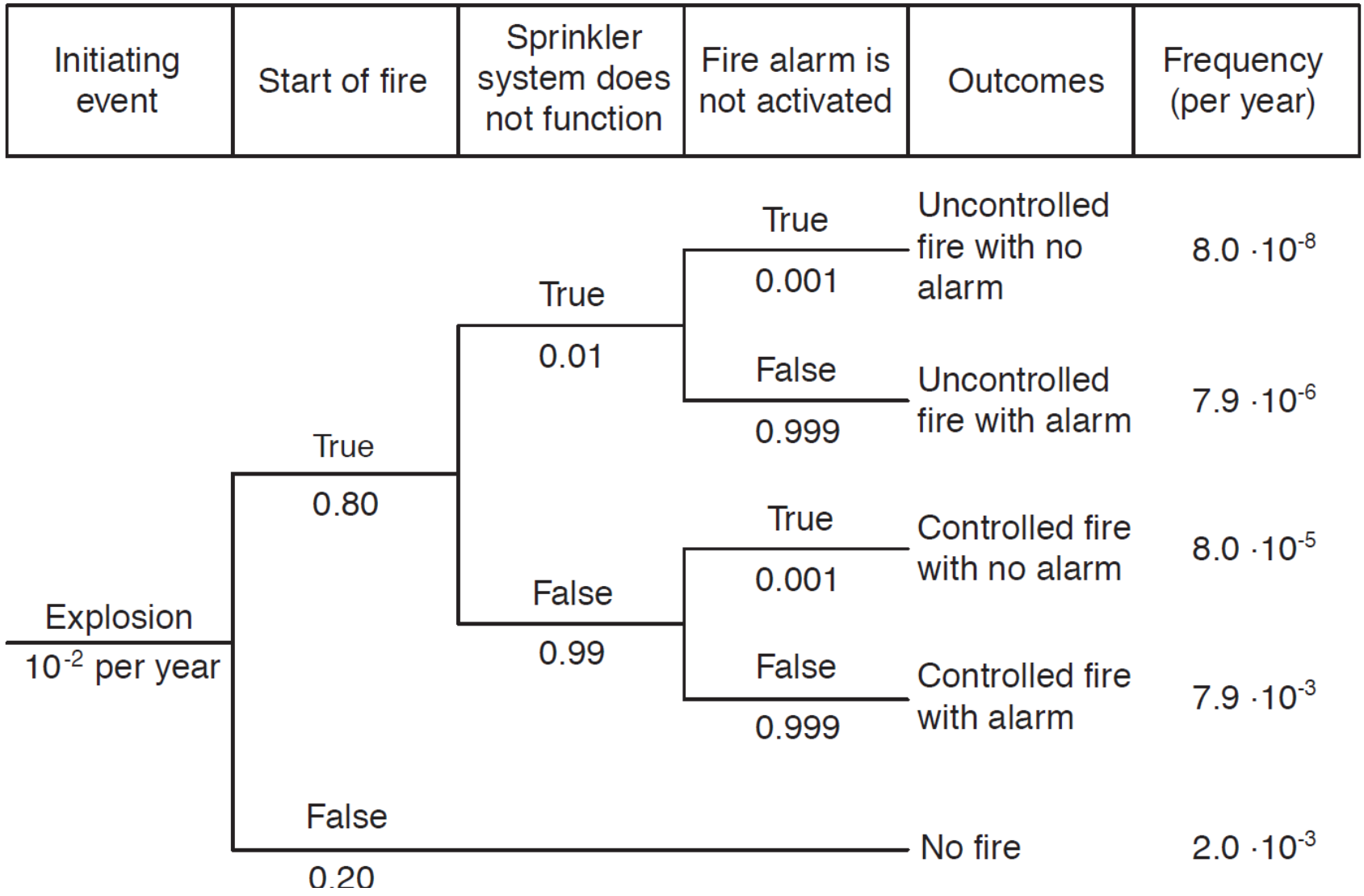
- The split fraction of an Event Tree Heading “A” is The Top event unavailability of the fault tree used to model the failure of the Event “A”



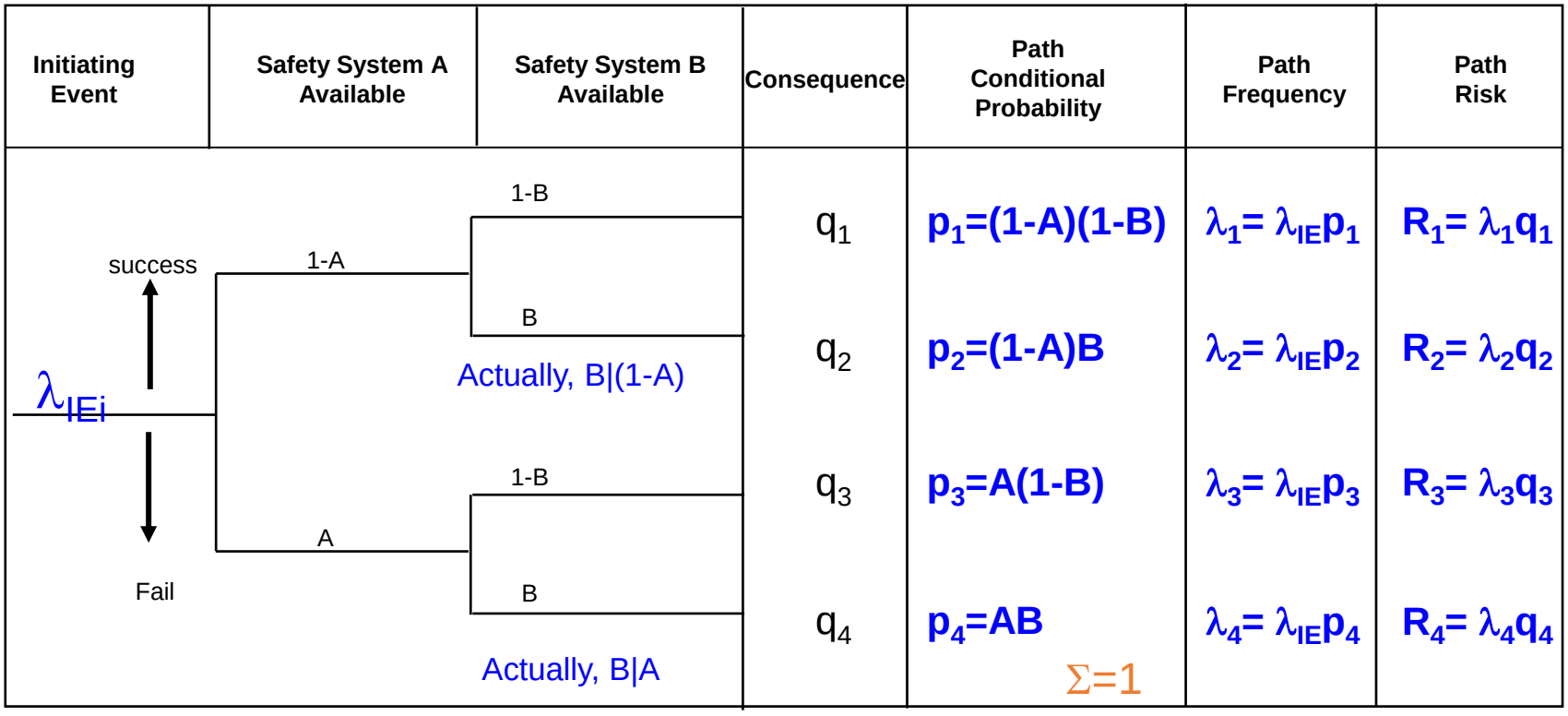
Example – Building with Fire Detector



Another example-Fire scenario caused by explosion



Event Tree Analysis



Given: λ_{IEi} = 2.3/yr; A=0.4, B=0.1, q_4= 24 fatalities

P_4= 0.4*0.1 = 0.04; $\lambda_4 = \lambda_{IE} P_4$ = 2.3*0.04/yr = 0.092/yr;

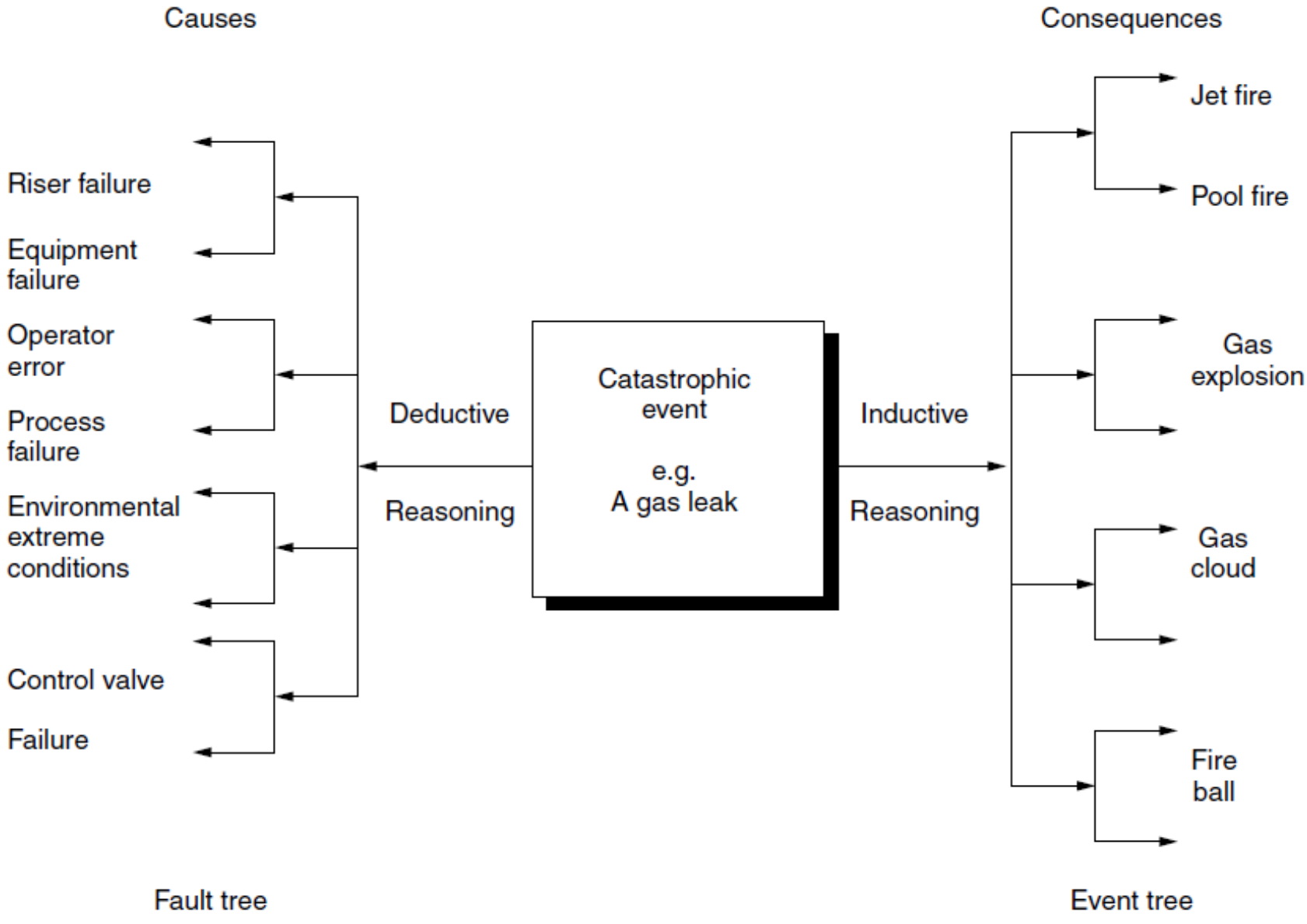
R_4=0.092*24 = 2.2 fatalities/yr

Total Risk (given IE_i) = $\lambda_{IEi} \Sigma R_{i|IEi}$; **Total System Risk = $\Sigma_j (\lambda_{IEj} \Sigma_i R_i)$**

FTA and ETA

- Event tree analysis and fault tree analysis are closely linked
  - Fault trees are often used to quantify system events that are part of event tree sequences;
  - The logical processes employed to evaluate event tree sequences and quantify the consequences are the same as in fault tree analyses;
  - Both produce Boolean logic expressions that are essential for probabilistic quantification

FTA and ETA


Causes
Consequences
Riser failure
Equipment failure
Operator error
Process failure
Environmental extreme conditions
Control valve
Failure
Deductive
Reasoning
Catastrophic event
e.g.
A gas leak
Inductive
Reasoning
Jet fire
Pool fire
Gas explosion
Gas cloud
Fire ball
Fault tree
Event tree

- End -