University of Nottingham
UK | CHINA | MALAYSIA

Lloyd's Register Foundation

Improved Risk Assessment

Professor John Andrews

Chair Risk and Resilience of Complex Systems

Annual Scientific Seminar

6th October 2021

Lloyd's Register Foundation (LRF)

'Our vision is to be known worldwide as a leading supporter of engineering-related research, training and education that makes a real difference in improving safety of the critical infrastructure on which modern society relies.'

'.. we promote scientific excellence and act as a catalyst working with others to achieve maximum impact.'

Next Generation Prediction Methodologies and Tools for System Safety Analysis (NxGen)

- Started in December 2019, 5 years duration
- 4 phases
  - Phase 1 – extend the capabilities of Fault Tree & Event tree Analysis
  - Phase 2 – extend the capabilities of phased mission analysis
  - Phase 3 – add dynamic capabilities to the modelling
  - Phase 4 – integration of stochastic models of the system failures with physical models

University of Nottingham
UK | CHINA | MALAYSIA

Industrial Partners

Network Rail

HIGH 1 SPEED

HS2

RSSB

First

NetworkRail
High Speed Ltd.

राष्ट्र की सेवा में परमाणु
प ऊ वि
DAE
ATOMS IN THE SERVICE OF THE NATION

igcar

राष्ट्र की सेवा में परमाणु
BARC
ATOMS IN THE SERVICE OF THE NATION

ROLLS RR ROYCE

BAE SYSTEMS

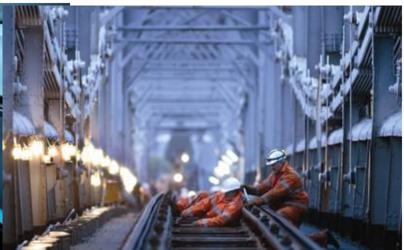
LR Lloyd's Register Foundation

LR Lloyd's Register









Background & Objectives

Background

- Current Risk Assessment tools include: Fault Tree Analysis, Event Tree Analysis
- The foundations of methodologies for safety critical systems were established in the 1960/70s.
  - Research has made considerable advances in the capabilities of analytical techniques since then.
  - Technology has advanced and system designs, their operating conditions and maintenance strategies are now significantly different to those of the 1970s.

Objectives

- Develop a single, generic methodology appropriate to meet the demands of modern industrial systems.
- Upwardly compatible - retain as much of the current methodology features as possible:
  - successfully supported safety assessments to date
  - companies want to retain the safety models they have evolved over time

University of Nottingham
UK | CHINA | MALAYSIA

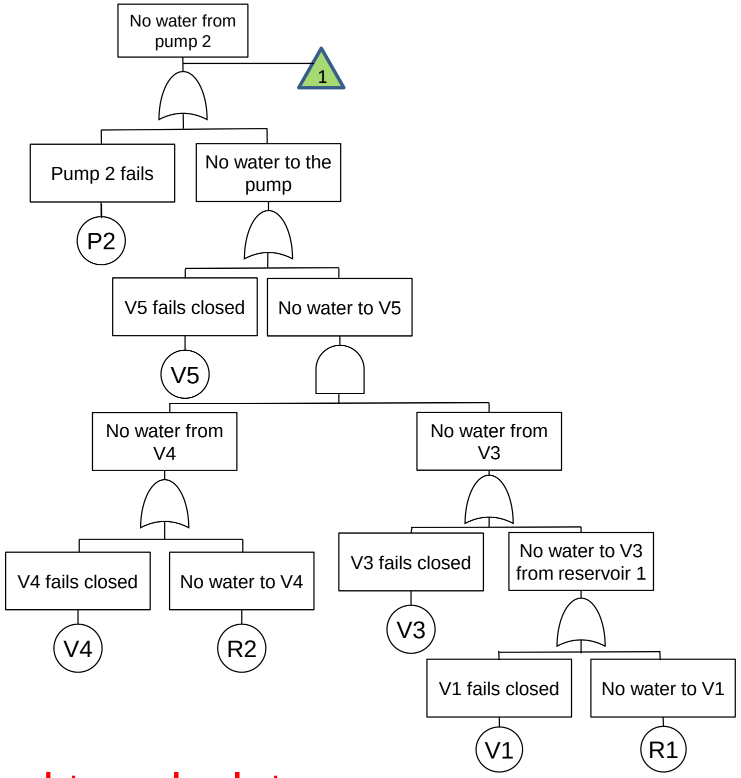
Traditional Approaches

Event Tree Analysis / Fault Tree Analysis

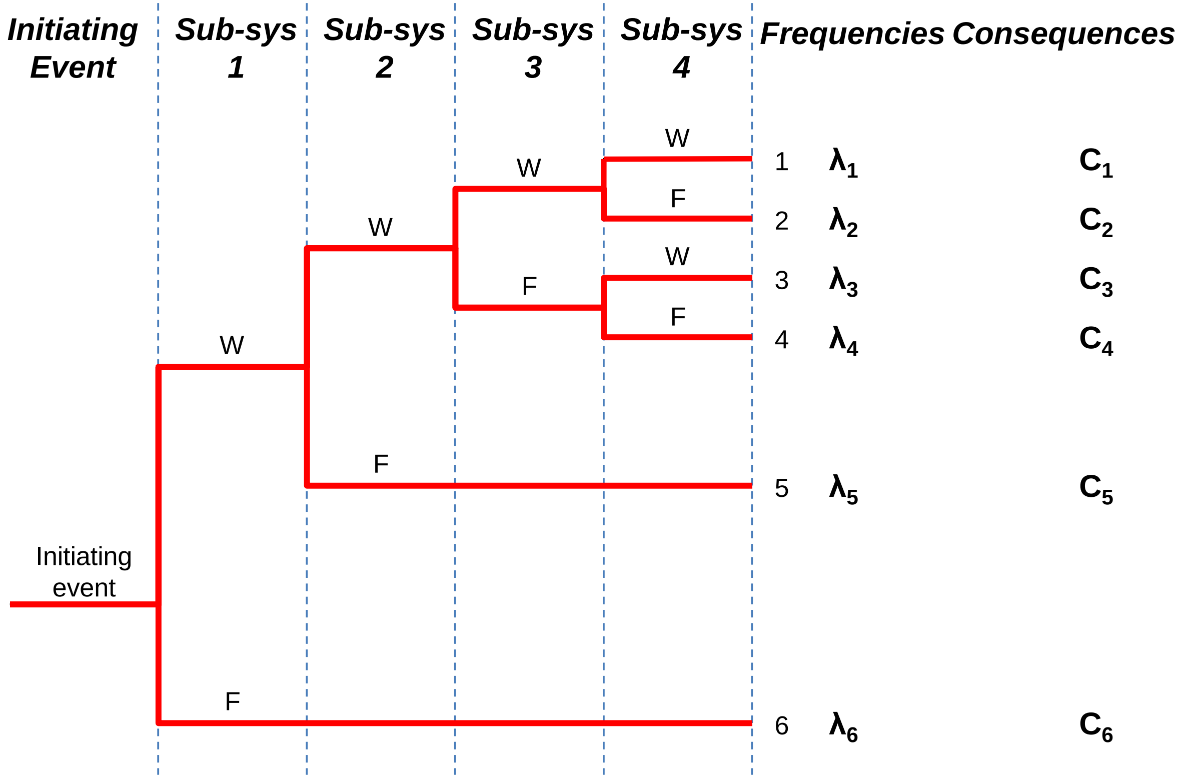
Traditional Approaches to Risk Modelling

Integrated Fault Tree Analysis / Event Tree Analysis Approach

Fault Tree Analysis



Event Tree Analysis

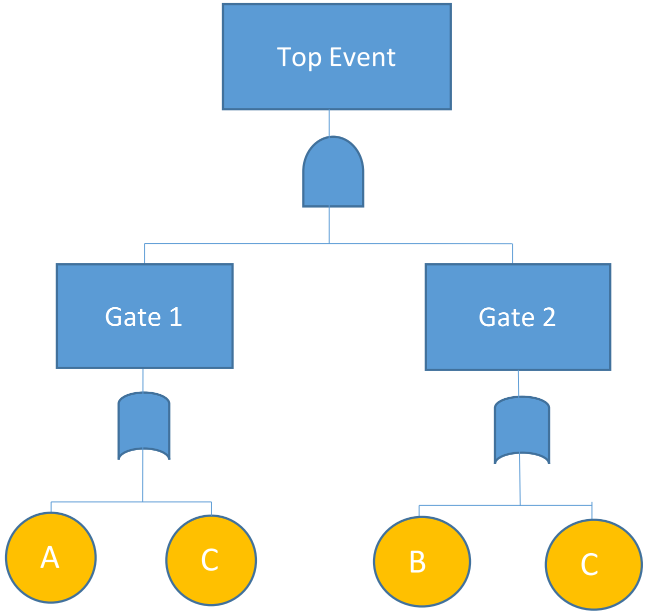


Used to calculate:

- Frequency of the initiating event
- Unavailability of enablers (responding safety systems)

$$Risk = \sum_{i=1}^{6} \lambda_i C_i$$

Fault Tree Analysis – Top Event Probability



$$TOP = (A + C).(B + C)$$

\+ OR

. AND

Minimal Cut Sets: {A, B}, {C}

Exact

$$Q_{SYS} = q_A\, q_B + q_C - q_A\, q_B\, q_C$$

Approximate

$$Q_{SYS} \leq 1 - (1 - q_A\, q_B)(1 - q_C)$$

Inclusion – exclusion expansion

$$Q_{SYS} = \sum_{i=1}^{N_C} P(C_i) - \sum_{i=2}^{N_C} \sum_{j=1}^{i-1} P(C_i \cap C_j) + \sum_{i=3}^{N_C} \sum_{j=2}^{i-1} \sum_{k=1}^{j-1} P(C_i \cap C_j \cap C_k) - \cdots$$
$$\cdots + (-1)^{N_C+1} P(C_1 \cap C_2 \cdots \cap C_{N_C})$$

Minimal Cut Set Upper Bound

$$Q_{SYS} \leq 1 - \prod_{i=1}^{N_C} (1 - P(C_i))$$

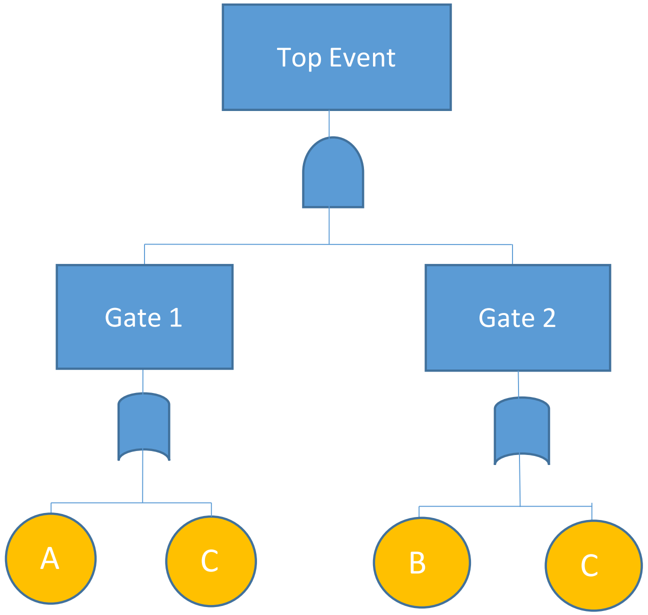
Initiator / Enabler events

Initiating Events: perturb system variables and place a demand on control / protection systems to respond

Enabling Events: are inactive control / protection systems which permit an initiating event to cause the top event

Critical System States: A critical state for a component i, is a state of the other components in the system such that the failure of component i causes the system to pass from the functioning to the failed state.

Fault Tree Analysis – failure intensity



Initiating events A, C

$$Q_{SYS} = q_A\, q_B + q_C - q_A\, q_B\, q_C$$

$$\boldsymbol{TOP = (A + C).(B + C)}$$

+ OR

. AND

Minimal Cut Sets: {A, B}, {C}

Criticality Function for the initiators:

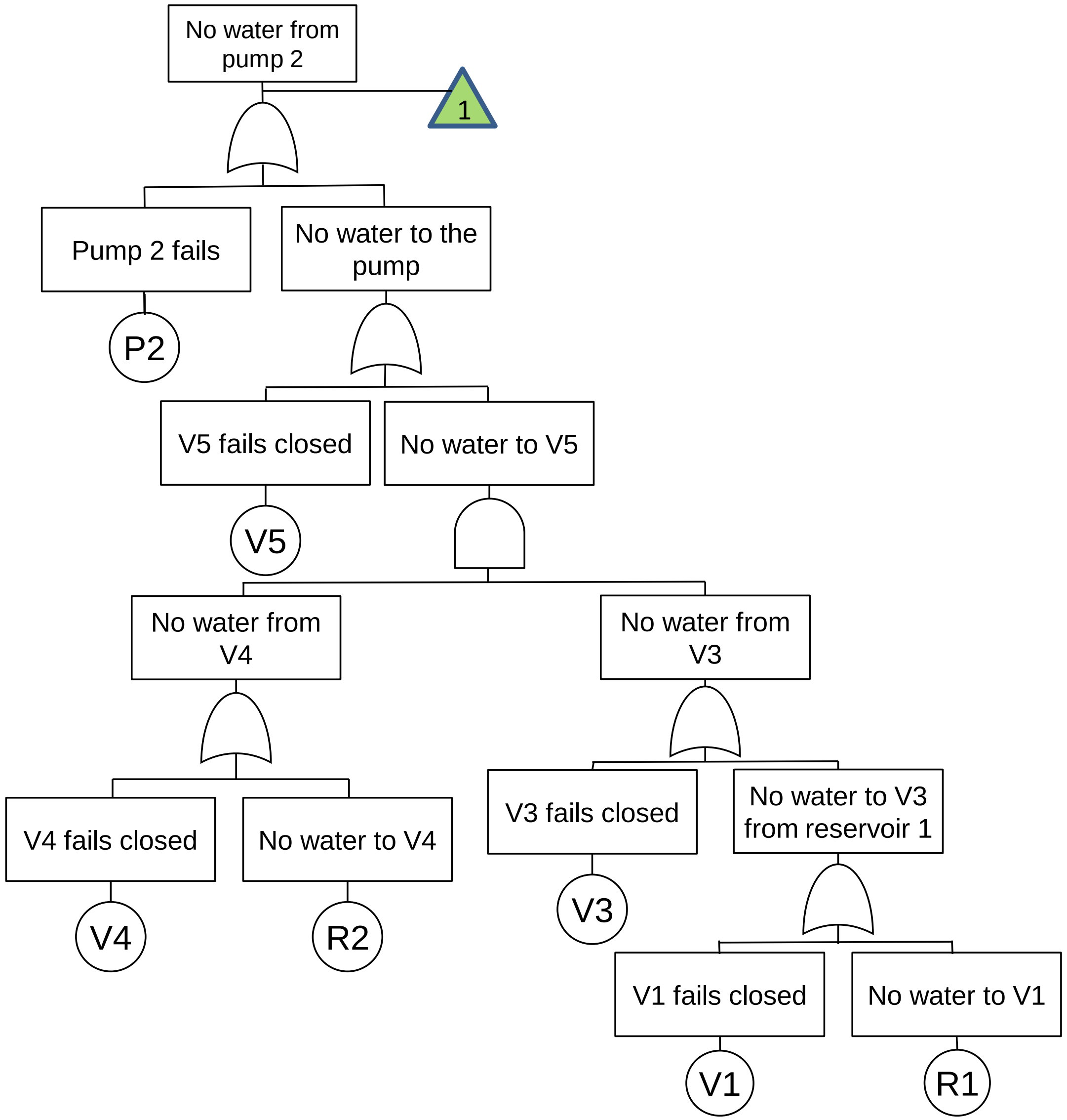
$$G_i(\boldsymbol{q}) = \frac{\partial Q_{SYS}}{\partial q_i}$$

$$G_A(\boldsymbol{q}) = q_B - q_B\, q_C = q_B(1 - q_C)$$

$$G_C(\boldsymbol{q}) = 1 - q_A\, q_B$$

$$w_{SYS}(t) = \sum_{\substack{i \\ initiators}} G_i(\boldsymbol{q}).w_i\,(t)$$

Fault Tree Analysis



Component failure models

- Limited maintenance process detail
  - No Repair: $Q(t) = F(t) = 1 - e^{-\lambda t}$
  - Revealed: $Q(t) = \frac{\lambda}{\lambda + \nu}\left(1 - e^{-(\lambda+\nu)t}\right)$
  - Unrevealed: $Q_{AV} = \lambda\left(\frac{\theta}{2} + \tau\right)$

PROJECT AIMS

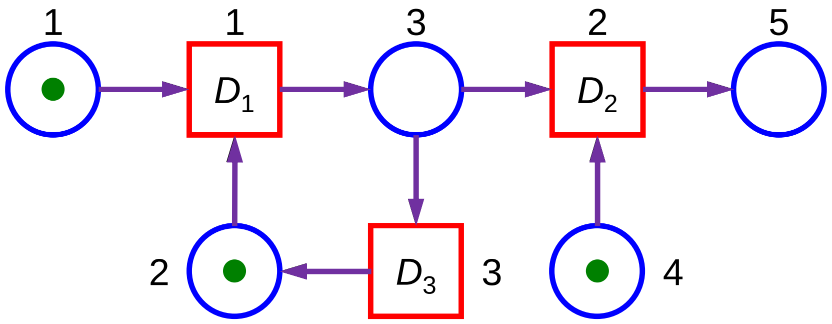
- Incorporate non-constant failure rates
- Incorporate dependent events
- Incorporate highly complex maintenance strategies

Supporting Methodologies:

Modelling Complexities / Dependencies

Petri Nets / Markov Methods

Petri Net Basics and Definitions



i

Places, p_i

- Marked with tokens

Edges

- From place to transition or transition to place.

j

D_j

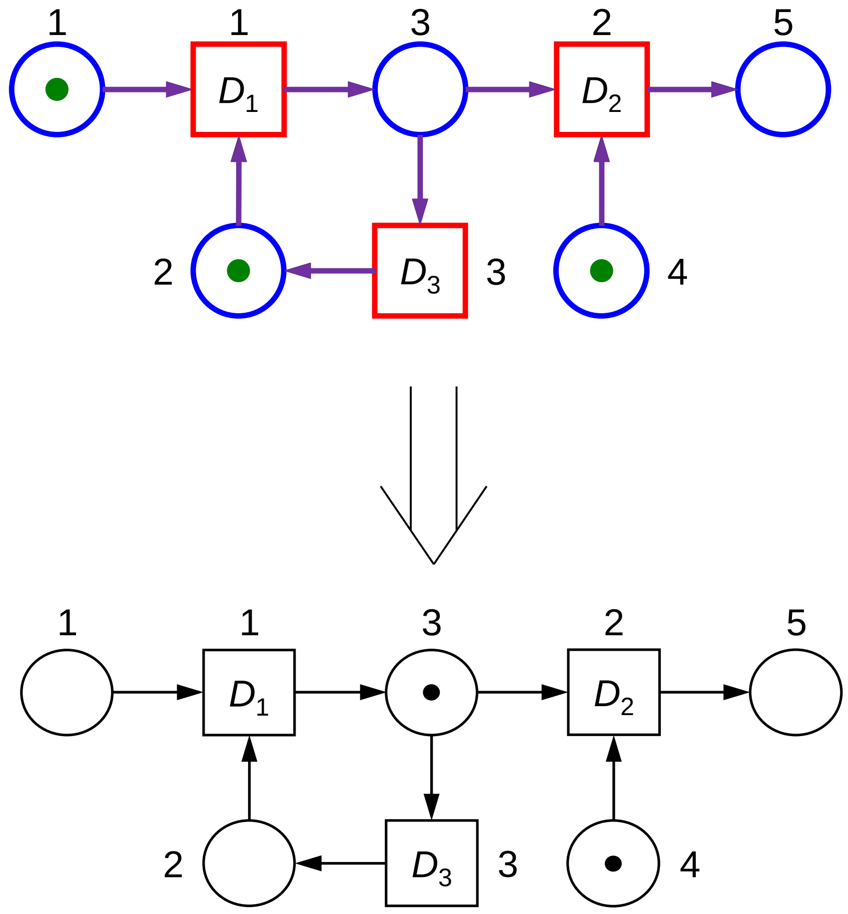
Transitions, t_j

- Time delay D_j determines token movement.
- Type:
  - immediate if $D_j = 0$
  - timed if $D_j \neq 0$

- Movement of tokens governed by the firing rule...

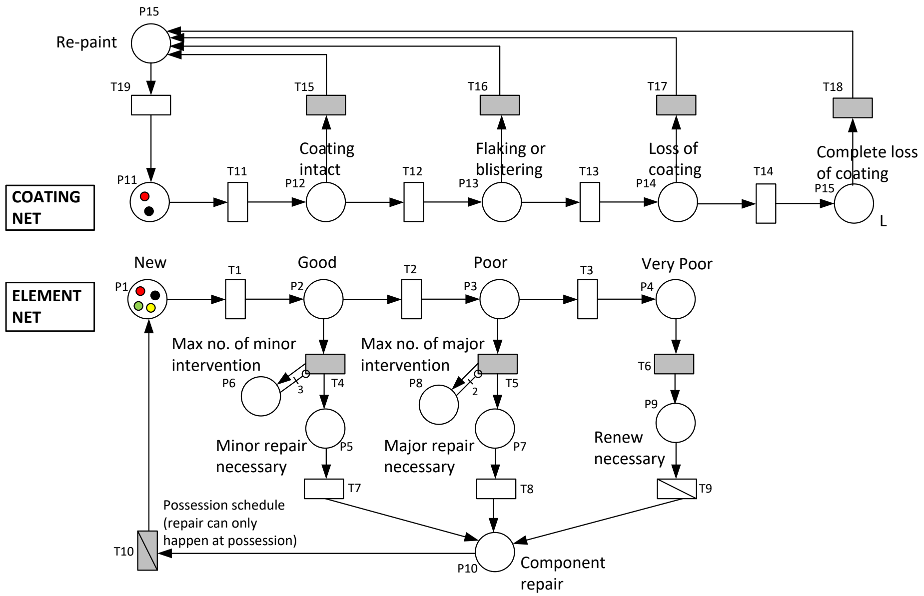
Petri Net Modelling

- If all input places of a transition are marked by at least one token then this transition is called **enabled**.
- After a delay $D \geq 0$ the transition **fires**. The firing removes one token from each of its input places and adds one token to each of its output places.



Modelling Methodology

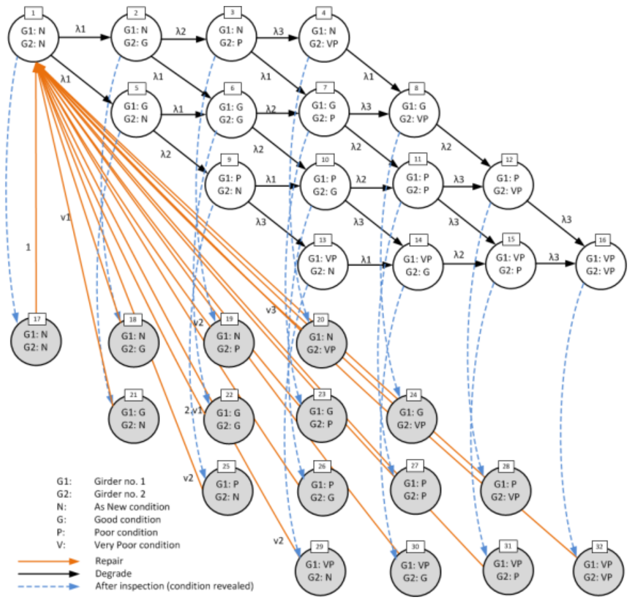
Petri-Net modelling (1962)



Features

- Any distribution of times to transition
- Capable of modelling very complex maintenance strategies
- Concise structure
- Solution by Monte Carlo simulation
- Produces distributions of durations and no of incidences of different states
- Modular – can form 'system' model by linking asset models

Markov modelling (1906)



Assumes:

- The future condition depends only on the current condition and not the history
- Constant rates of transition

Features

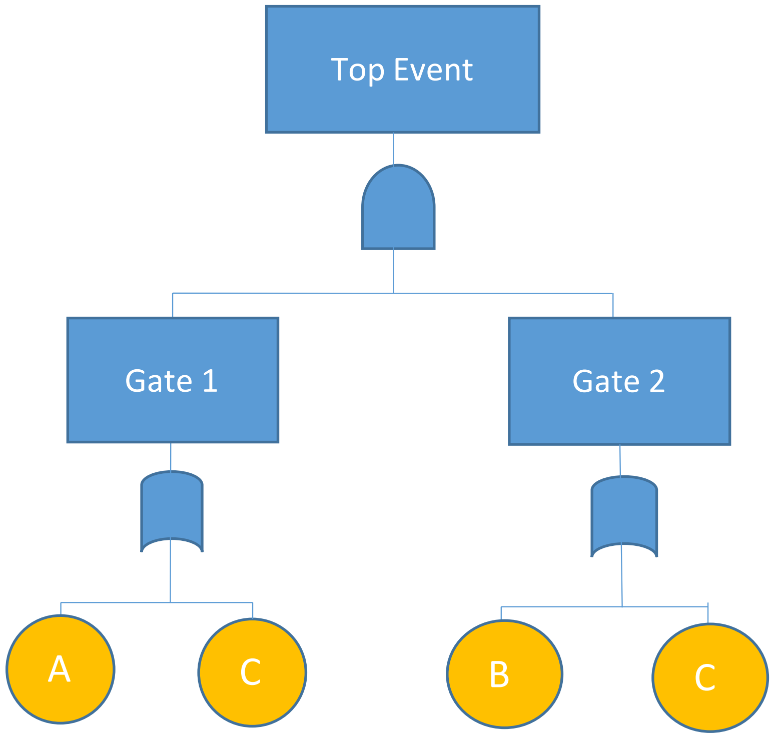
- State-space explosion
- Difficult to model decisions based on condition
- Can not combine asset models to form a 'system' model

Supporting Methodologies:

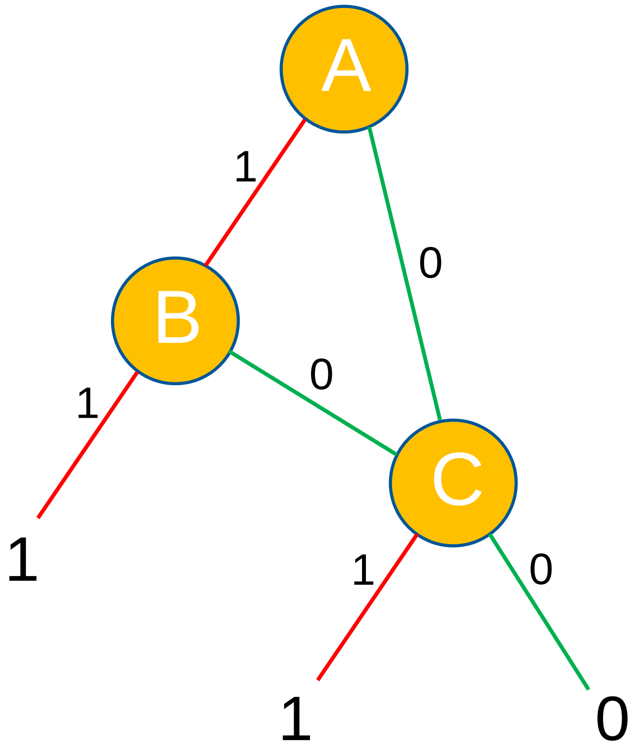
Fault Tree Quantification

Binary Decision Diagrams (BDDs)

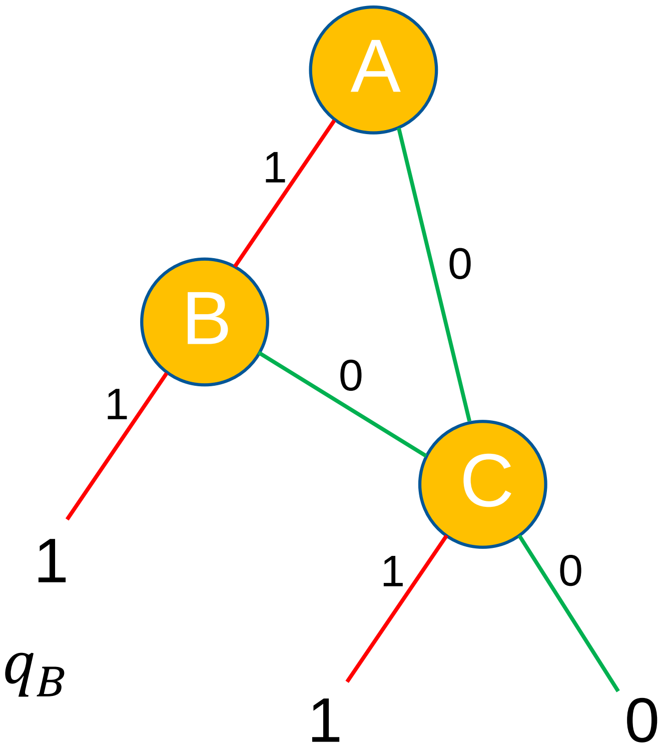
Binary Decision Diagrams – Top Event Probability



ORDERING A < B < C



Binary Decision Diagrams – Top Event Probability



$q_A \, q_B$

$q_A(1 - q_B)q_C$
$+(1 - q_A) \, q_C$

$+$ OR
$.$ AND

$$TOP = A.B + A.\bar{B}.C + \bar{A}.C$$

$$Q_{SYS} = q_A \, q_B + q_A(1 - q_B)q_C + (1 - q_A) \, q_C$$
$$= q_A \, q_B + q_C - q_A \, q_B \, q_C$$

- Exact
- Fast
- Efficient

} No need to derive the Min Cut Sets as an intermediate step

Binary Decision Diagram – Top Event Frequency

$$w_{SYS}(t) = \sum_{\substack{i \\ initiators}} G_i(\boldsymbol{q}).w_i\ (t)$$

The Criticality Function, G_i(q), is the probability that the system is in a critical state for component i such that the failure of component i causes system failure.

w_i(t) is the failure intensity of component i.

$$G_i(\boldsymbol{q}) = \frac{\partial Q_{SYS}}{\partial q_i} = Q_{SYS}(1_i, \boldsymbol{q}) - Q_{SYS}(0_i, \boldsymbol{q})$$

$Q_{SYS}(1_i, \boldsymbol{q})$ probability that the system fails with component i failed

$Q_{SYS}(0_i, \boldsymbol{q})$ probability that the system fails with component i working

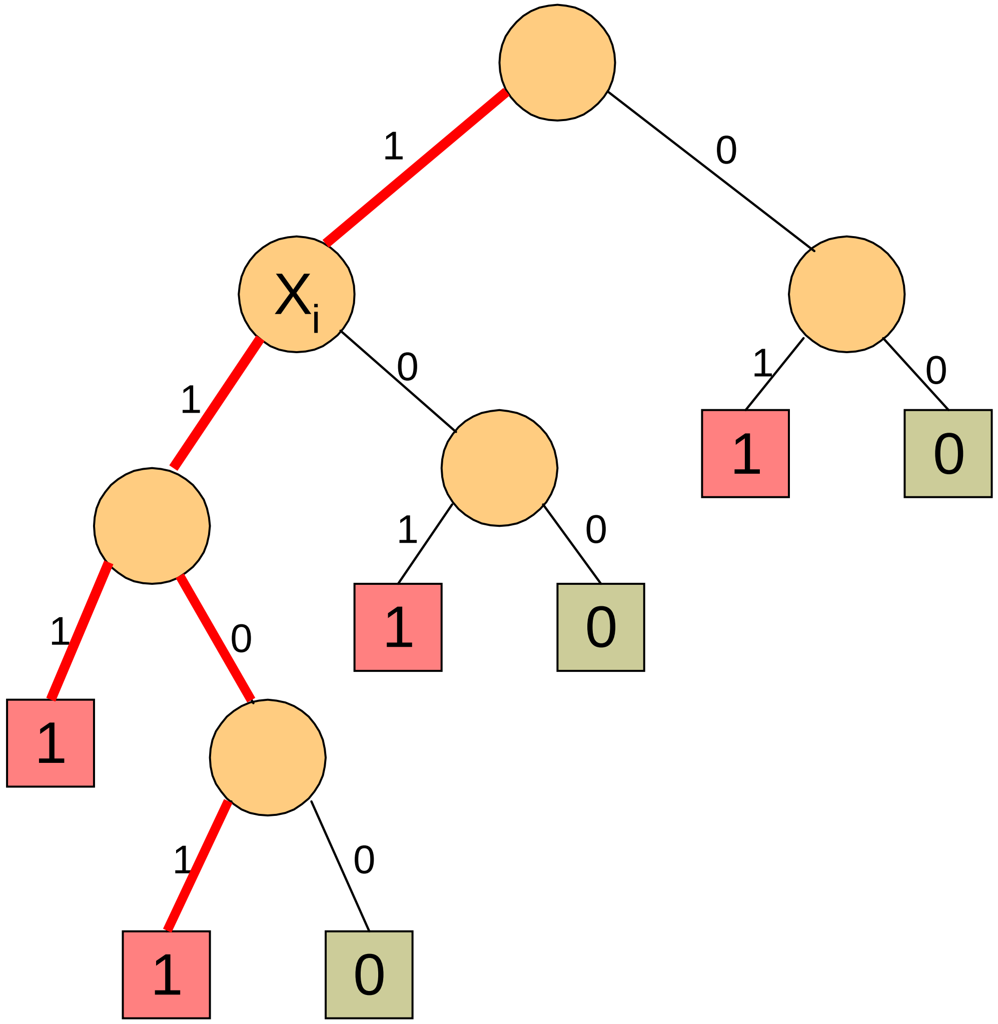
Note: the Criticality Function is also known as Birnbaum's Measure of importance

Criticality Function: Routes to a terminal-1

Criticality for X_i

Three Options:

1. paths through X_i on its 1-branch to a terminal-1
2. paths through X_i on its 0-branch to a terminal-1
3. paths which don't pass through X_i on way to a terminal-1

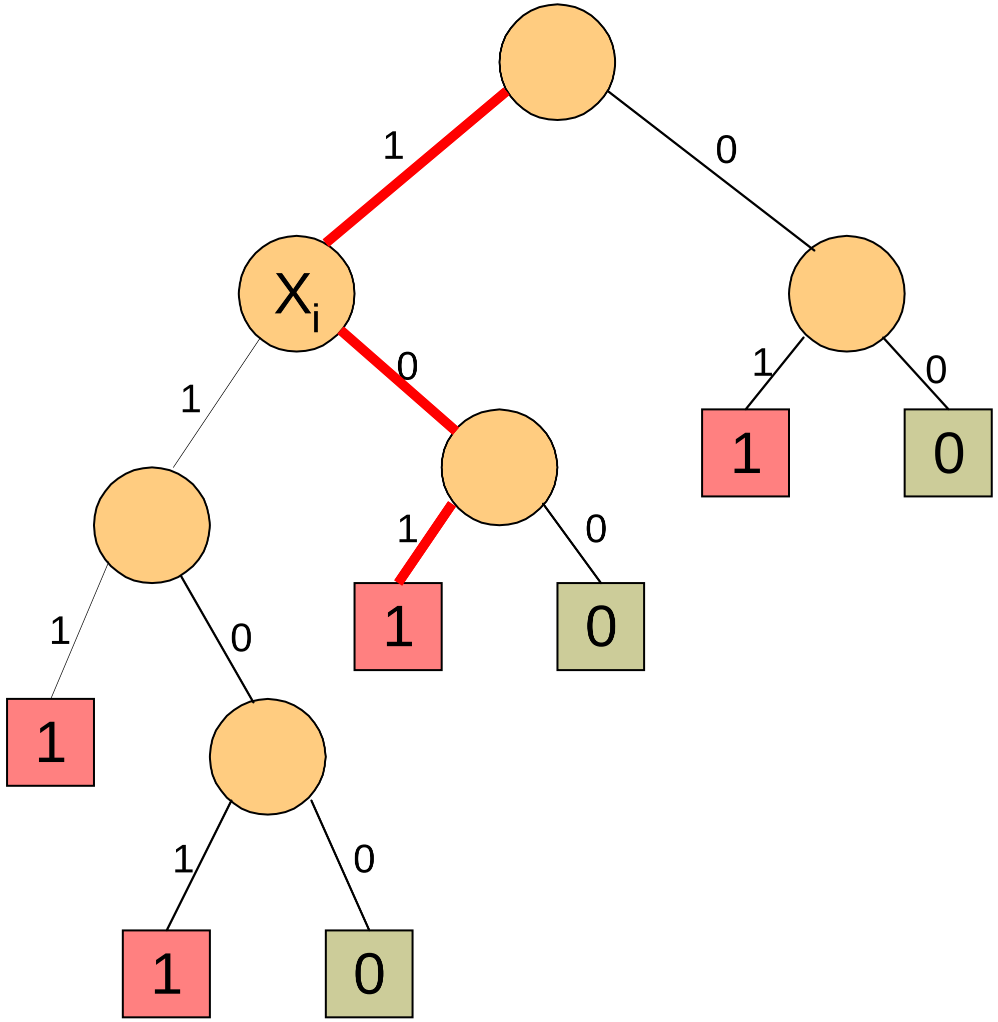


Criticality Function: Routes to a terminal-1

Criticality for X_i

Three Options:

1. paths through X_i on its 1-branch to a terminal-1
2. paths through X_i on its 0-branch to a terminal-1
3. paths which don't pass through X_i on way to a terminal-1

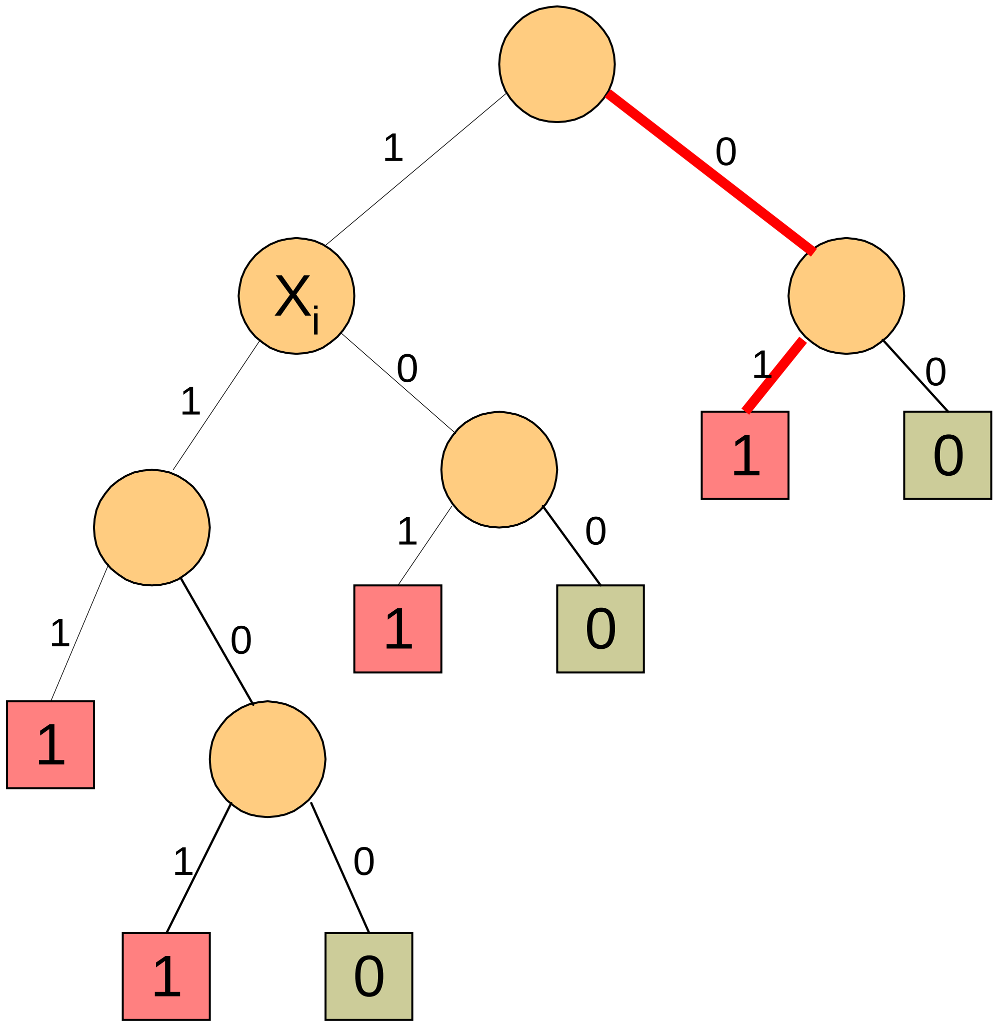


Criticality Function: Routes to a terminal-1

Criticality for X_i

Three Options:

1. paths through X_i on its 1-branch to a terminal-1
2. paths through X_i on its 0-branch to a terminal-1
3. paths which don't pass through X_i on way to a terminal-1



Criticality Function

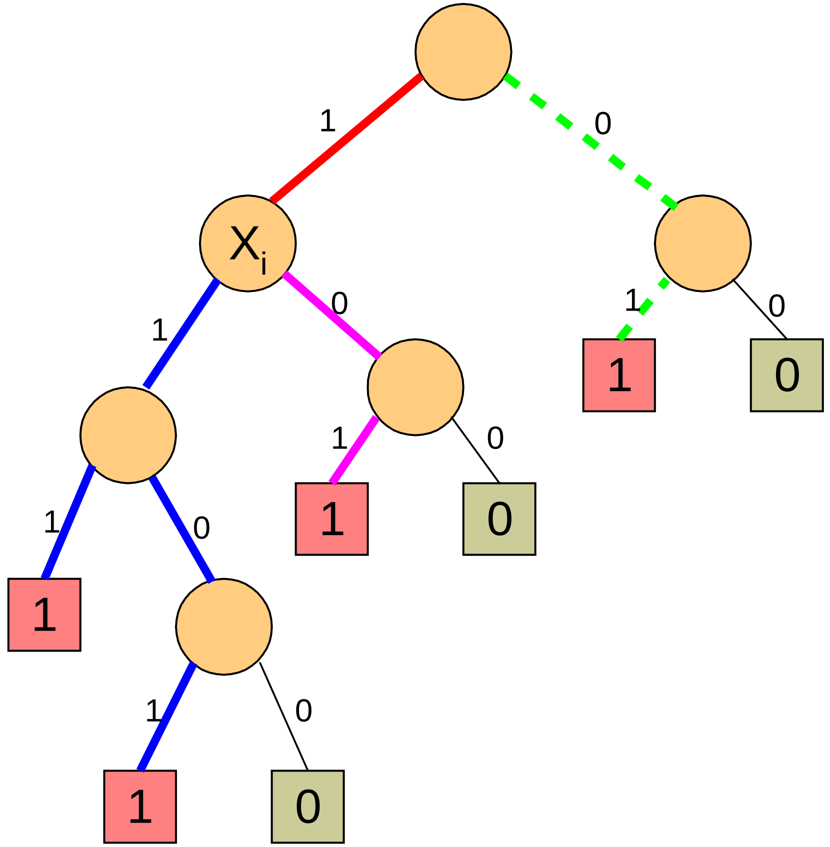
$$Q(1_i, \underline{q}) = \sum_{i=1}^{n} (pr_{xi}(\underline{q}).po^1_{xi}(\underline{q})) + Z(\underline{q})$$

$$Q(0_i, \underline{q}) = \sum_{i=1}^{n} (pr_{xi}(\underline{q}).po^0_{xi}(\underline{q})) + Z(\underline{q})$$

$pr_{xi}(\underline{q})$ is the probability of the path section from the root node to node x_i.

$po^1_{xi}(\underline{q})$ is the probability of the path section from the 1 branch of node x_i to a terminal 1 node (excluding probability of x_i).

$po^0_{xi}(\underline{q})$ is the probability of the path section from the 0 branch of node x_i to a terminal 1 node (excluding probability of x_i).



$Z(\underline{q})$ is the probability of the paths from the root node to the terminal 1 node not passing through the node for variable x_i.

Criticality Function

$$G_i(\boldsymbol{q}) = Q_{SYS}(1_i,\boldsymbol{q}) - Q_{SYS}(1_i,\boldsymbol{q})$$

$$Q_{SYS}(1_i,\boldsymbol{q}) = \sum_{all\ xi} \left(pr_{xi}(\boldsymbol{q}).po_{xi}^{1}(\boldsymbol{q})\right) + Z(\boldsymbol{q})$$

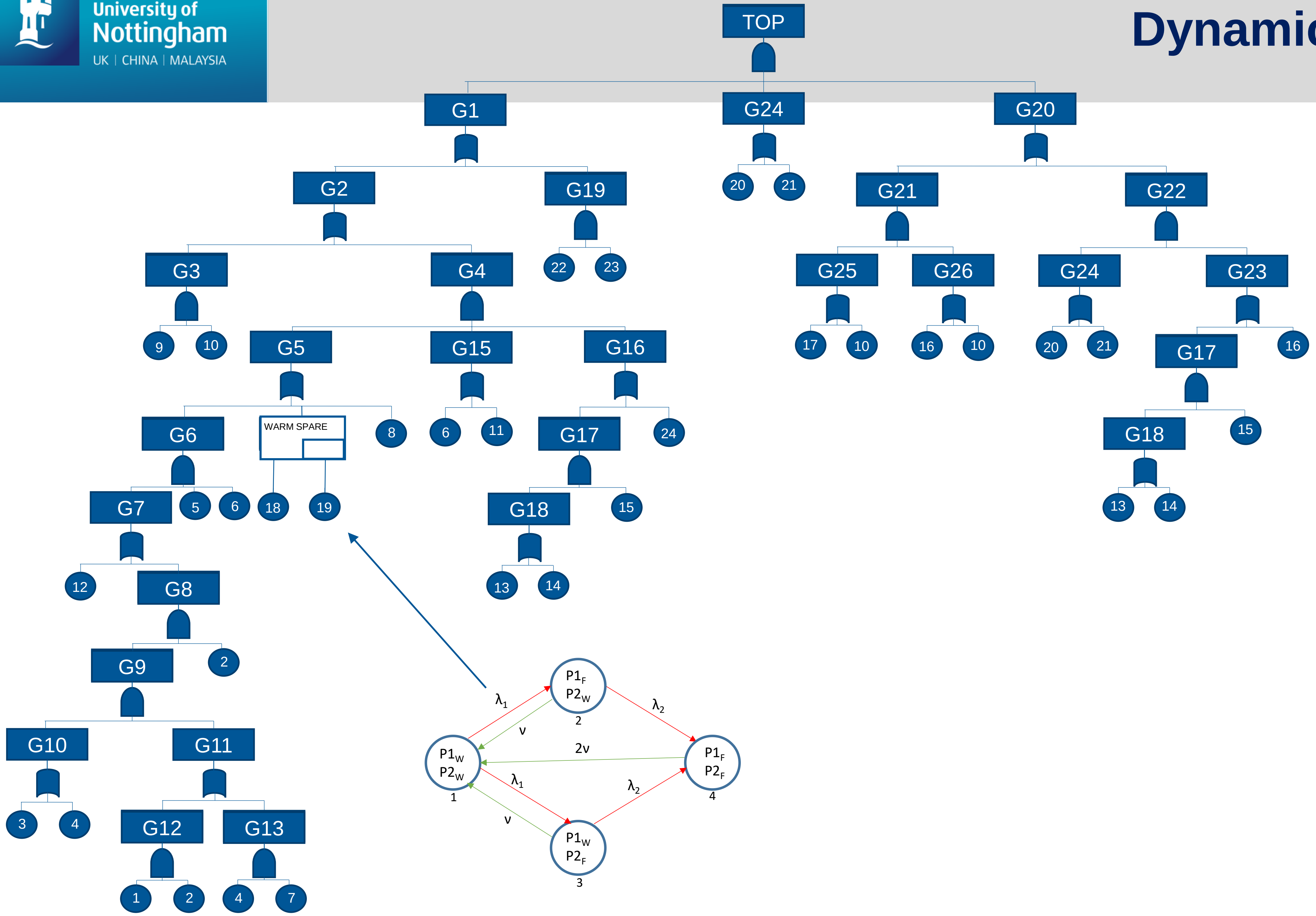
$$Q_{SYS}(0_i,\boldsymbol{q}) = \sum_{all\ xi} \left(pr_{xi}(\boldsymbol{q}).po_{xi}^{0}(\boldsymbol{q})\right) + Z(\boldsymbol{q})$$

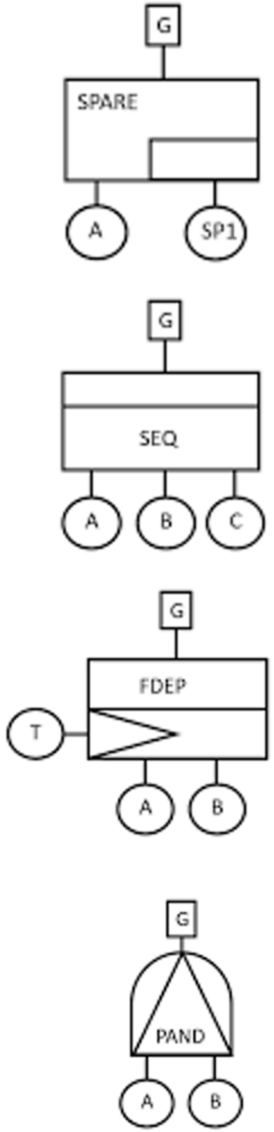
$$G_i(\boldsymbol{q}) = \sum_{all\ xi} pr_{xi}(\boldsymbol{q})\left[po_{xi}^{1}(\boldsymbol{q})) - po_{xi}^{0}(\boldsymbol{q}))\right]$$

$$w_{SYS}(t) = \sum_{\substack{i \\ initiators}} G_i(\boldsymbol{q}).w_i\,(t)$$

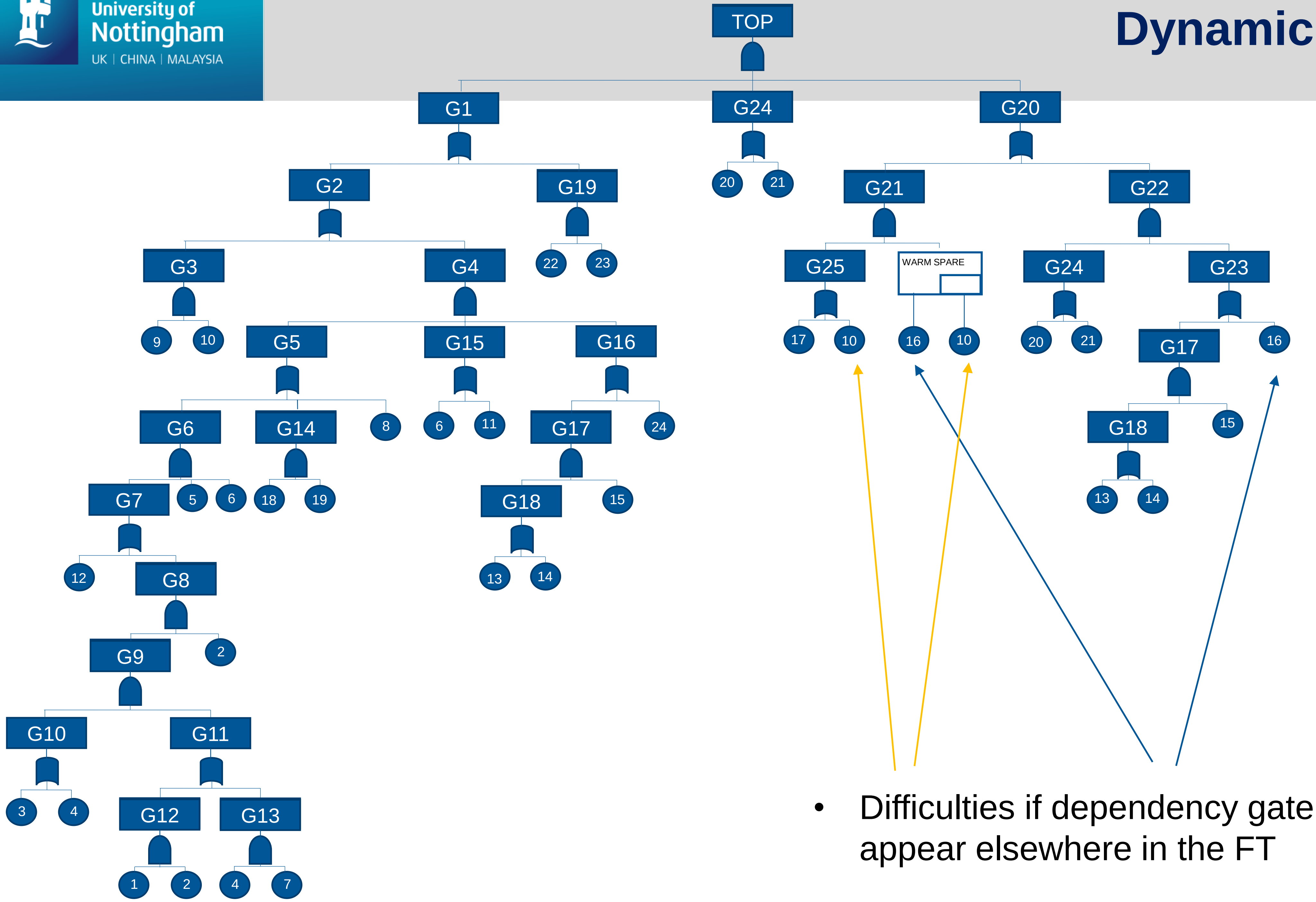
Approaches to Dependencies

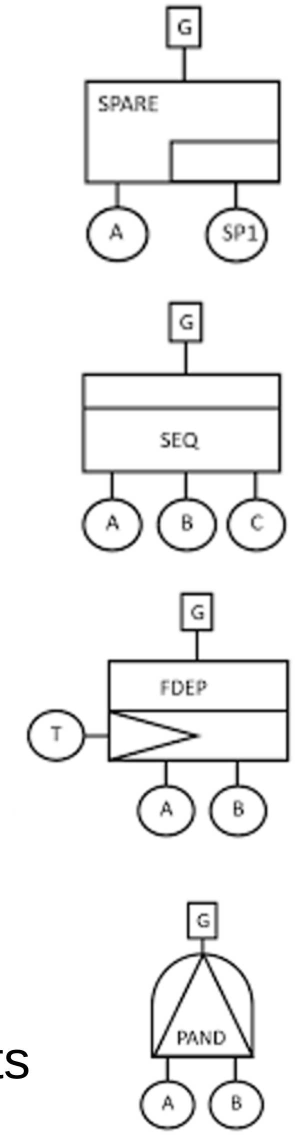
Dynamic Fault Trees





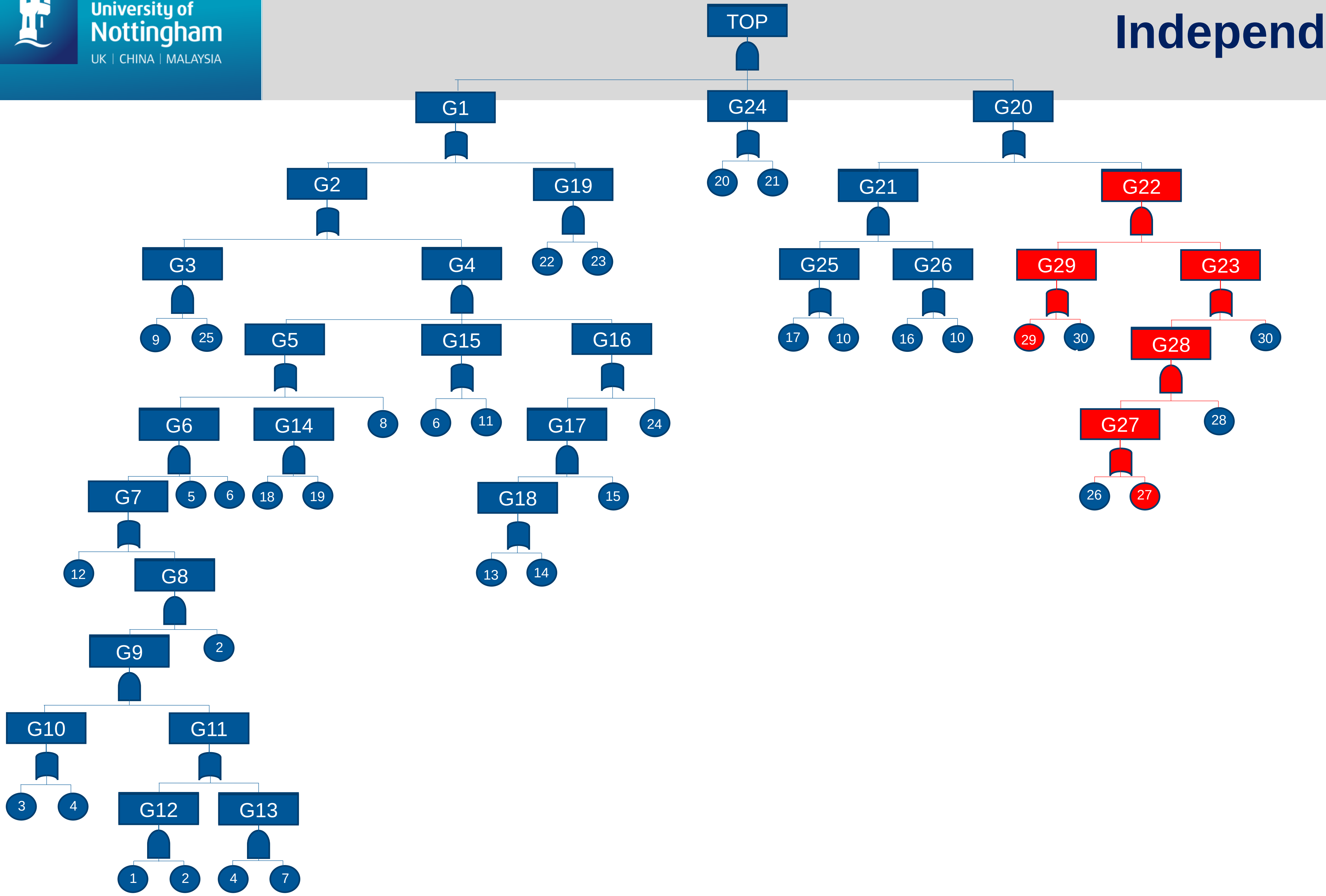
Dynamic Fault Trees





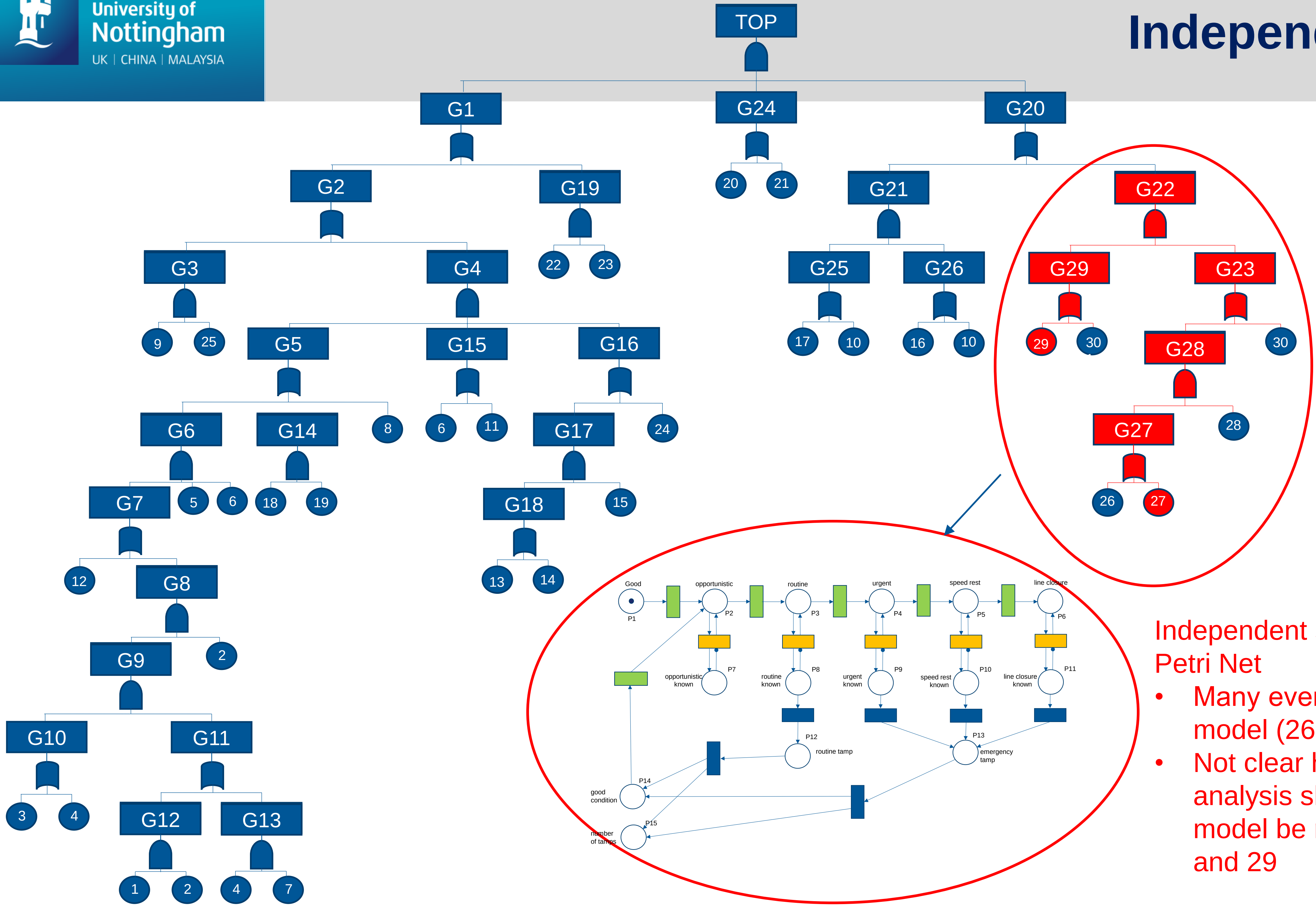
- Difficulties if dependency gate inputs appear elsewhere in the FT

Independent Modules



Dependencies between 27 and 29

Independent Modules



Dependencies between 27 and 29

Independent section solved using a Petri Net

- Many events don't need to be in this model (26, 28, 30)
- Not clear how to include them in the analysis should the dependency model be reduced to just events 27 and 29

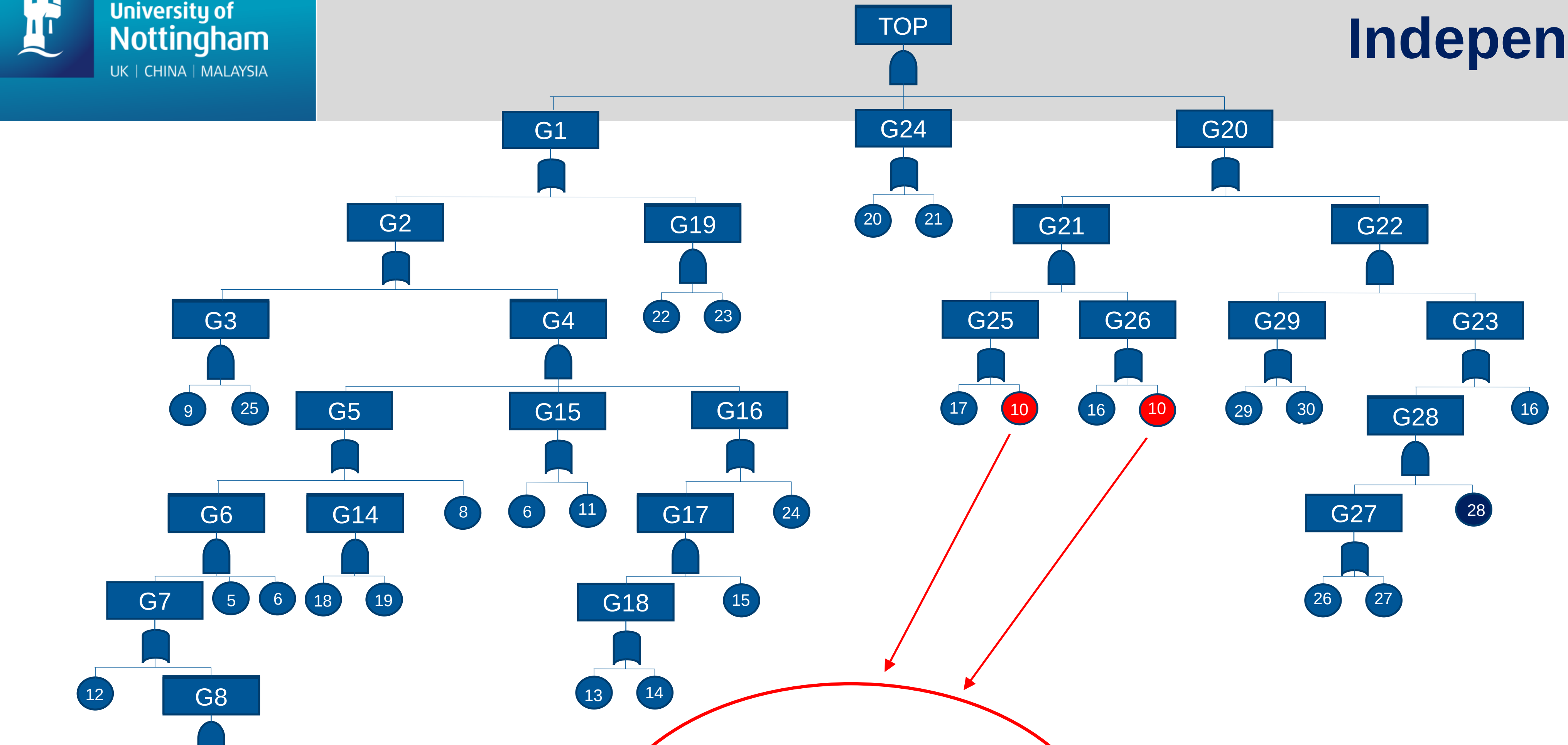
University of Nottingham
UK | CHINA | MALAYSIA

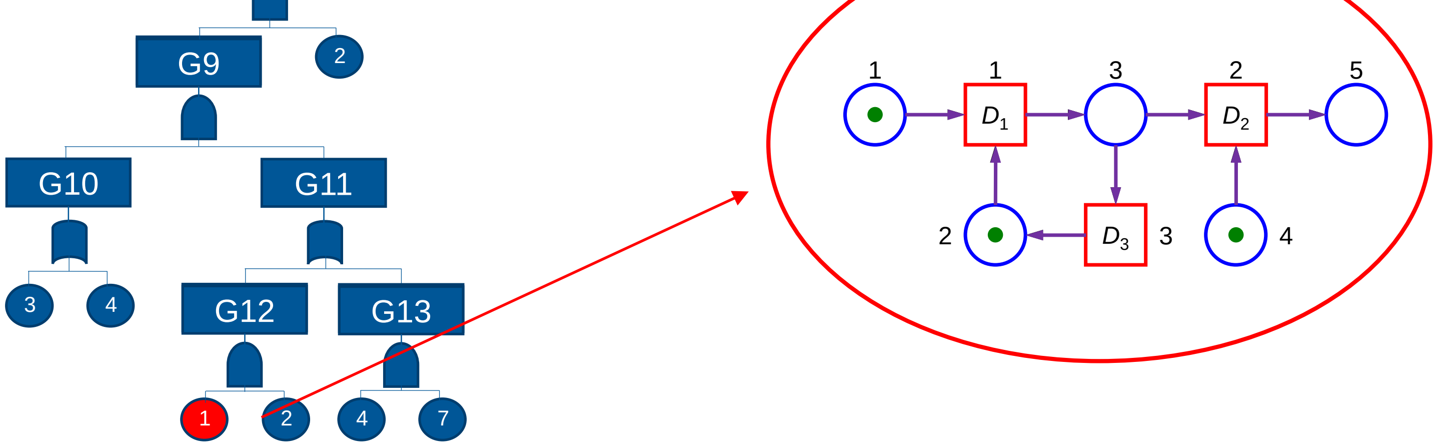
Modelling Requirements

Objectives

- Retain the FT and ET to represent the causality of system failures.
- Model the dependencies and complexities using Petri Nets or Markov as appropriate.
- Dependency models take substantial computer resource to solve – especially large models (their size should be minimised).
- No Matter where or how many of the dependent basic events occur in the FT - the simplest dependency model is used to analyse the results for those events alone

Independent Modules

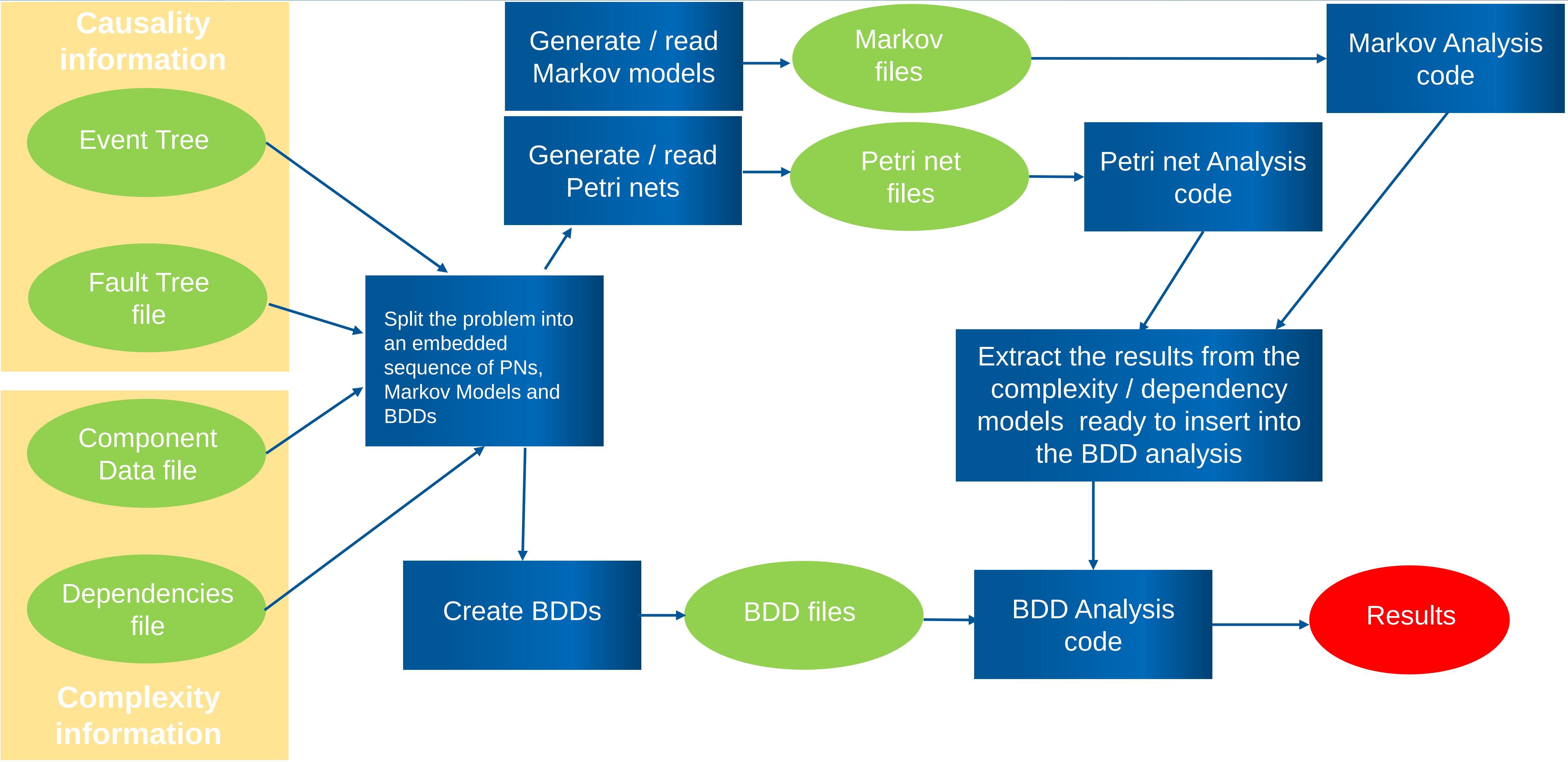




Small model containing only the dependent events

Results integrated back into the assessment of the remainder of the FT

Basic Structure of the Code



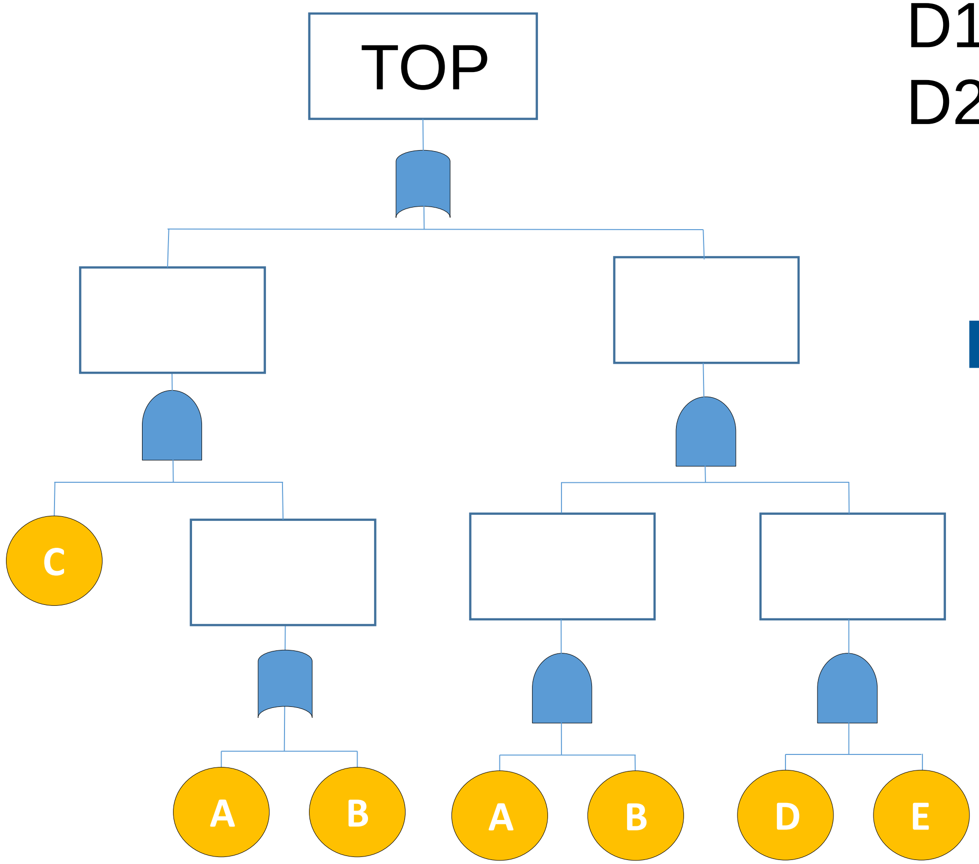
New methodology
Top event probability – dependent events

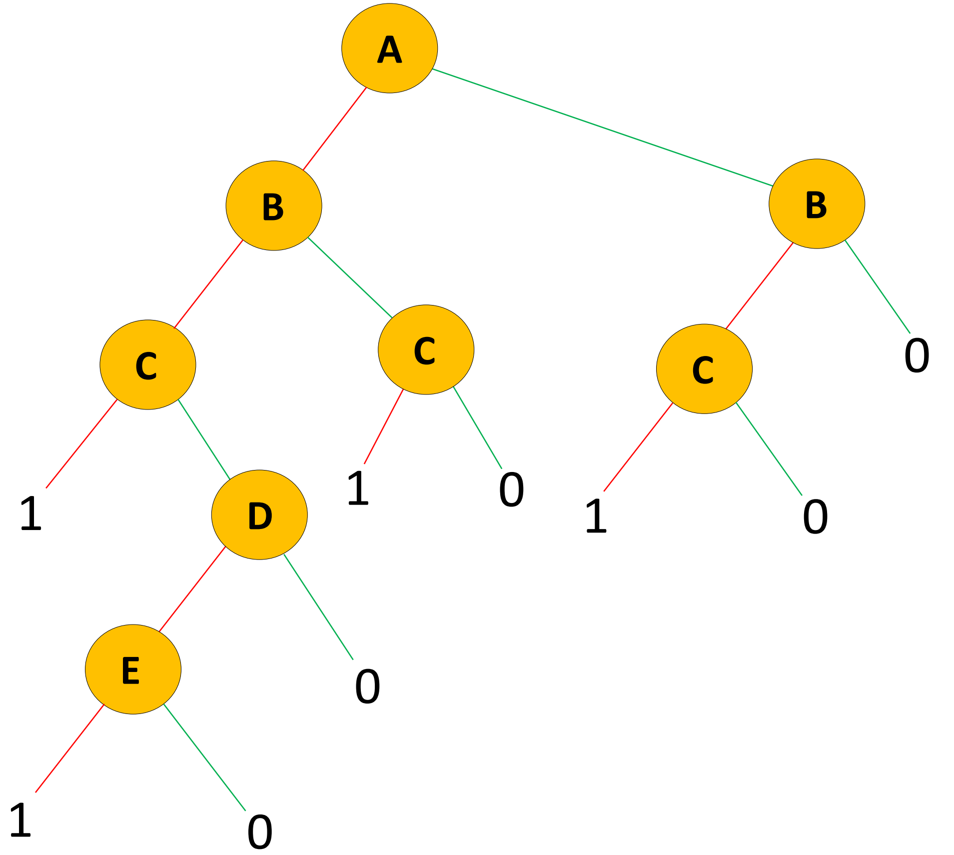
Example

Dependency groups

D1 = { B, C }

D2 = { D, E }





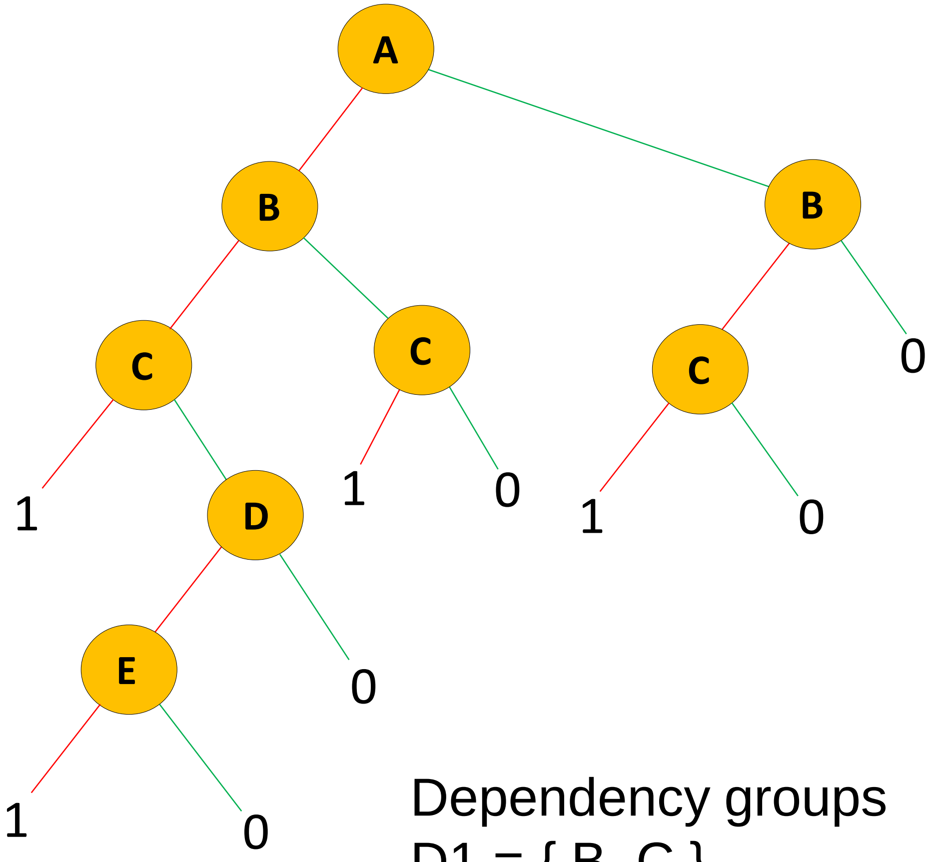
Notation

$path_j$ – jth path through the BDD to a terminal – 1

- {variables on the path identifying if they pass on the 1-branch or 0-branch}

$Ipath_j$ - {independent variables on path *j* identifying if they pass on the 1-branch or 0-branch}

$Dpath_j^k$ - {variables on the path *j* belonging to dependency group identifying if they pass on the 1-branch or 0-branch}

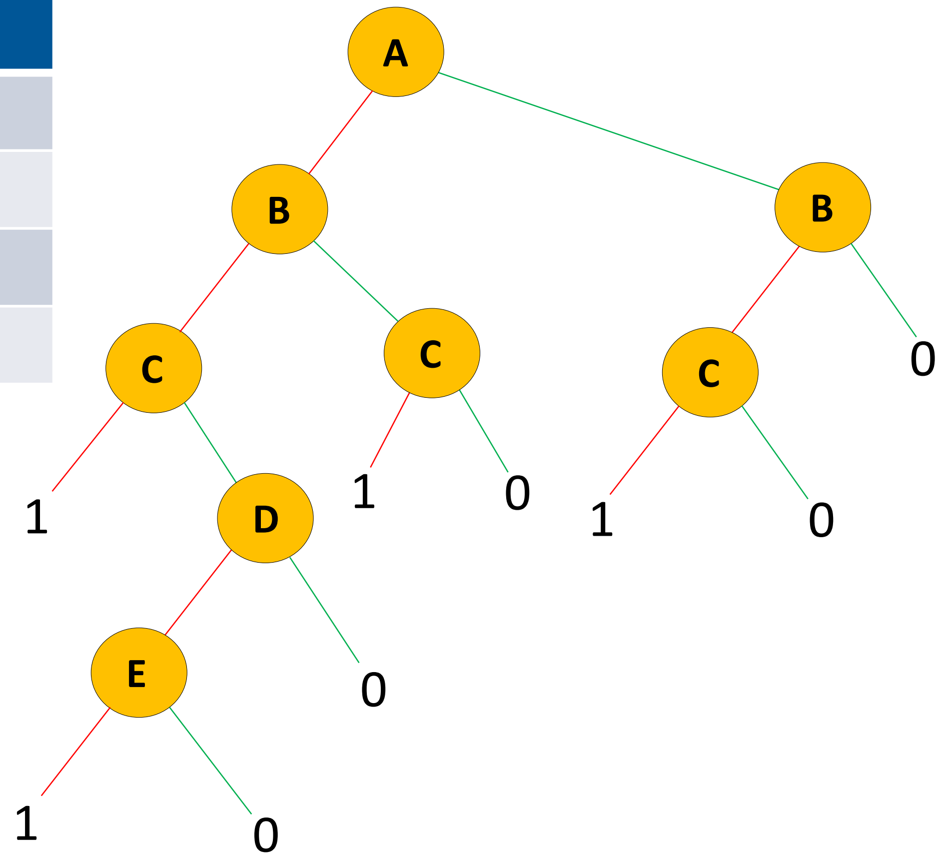


Dependency groups
D1 = { B, C }
D2 = { D, E }

j	$path_j$	$Ipath_j$	$Dpath_j^1$	$Dpath_j^2$
1	a_1, b_1, c_1	a_1	b_1, c_1	
2	a_1, b_1, c_0, d_1, e_1	a_1	b_1, c_0	d_1, e_1
3	a_1, b_0, c_1	a_1	b_0, c_1	
4	a_0, b_1, c_1	a_0	b_1, c_1	

Notation

j	$path_j$	$Ipath_j$	$Dpath_j^1$	$Dpath_j^2$
1	a_1, b_1, c_1	a_1	b_1, c_1	
2	a_1, b_1, c_0, d_1, e_1	a_1	b_1, c_0	d_1, e_1
3	a_1, b_0, c_1	a_1	b_0, c_1	
4	a_0, b_1, c_1	a_0	b_1, c_1	



$$Q_{SYS} = \sum_{j=1}^{npath} \left[P(Ipath_j) . \prod_{k=1}^{ndep} P(Dpath_j^k) \right]$$

University of Nottingham
UK | CHINA | MALAYSIA

New methodology
Top event intensity – dependent events

Criticality Function

For independent events

$$G_i(\boldsymbol{q}) = \frac{\partial Q_{SYS}}{\partial q_i} = Q_{SYS}(1_i, \boldsymbol{q}) - Q_{SYS}(0_i, \boldsymbol{q})$$

$$= \sum_{all\ x_i} \left(pr_{x_i}(\boldsymbol{q}).po^1_{x_i}(\boldsymbol{q}) \right) + Z(\boldsymbol{q}) - \left(pr_{x_i}(\boldsymbol{q}).po^0_{x_i}(\boldsymbol{q}) \right) - Z(\boldsymbol{q})$$

$$= \sum_{all\ x_i} pr_{x_i}(\boldsymbol{q}).(po^1_{x_i}(\boldsymbol{q}) - po^0_{x_i}(\boldsymbol{q}))$$

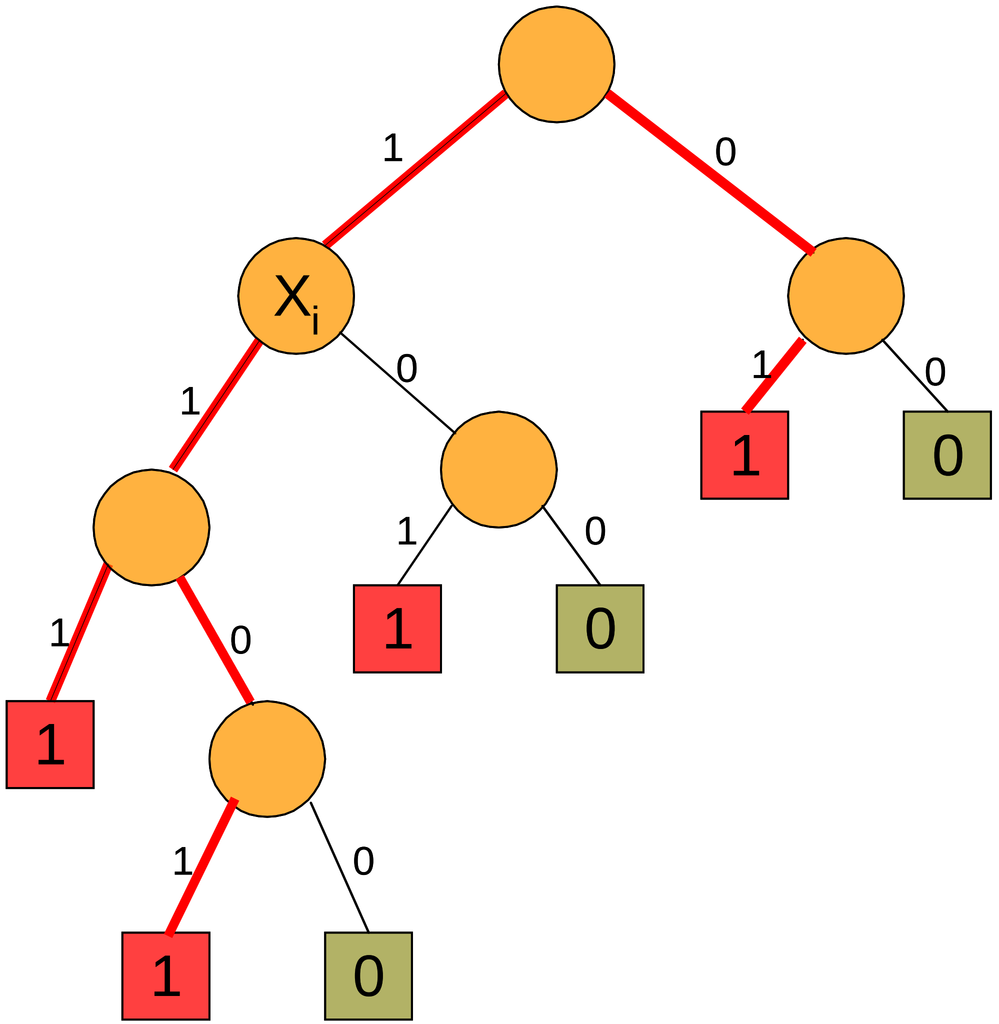
For dependent events

- Cannot use the same form of equations as for independent events:
  - The *pr(**q**)* and *po(**q**)* terms may each contain events in the same dependency group
  - The *Z(**q**)* term may also contain events in the same dependency group as X_i and so will not cancel each other

Criticality Function: Routes to a terminal-1

Criticality for X_i

$$Q_{SYS}\left(1_i,\underline{q}\right)=\sum_{x_{i_1\in path_j}} P\left(path_j - x_{i_1}\right)+\sum_{x_i\notin path_j} P(path_j|x_i=1)$$

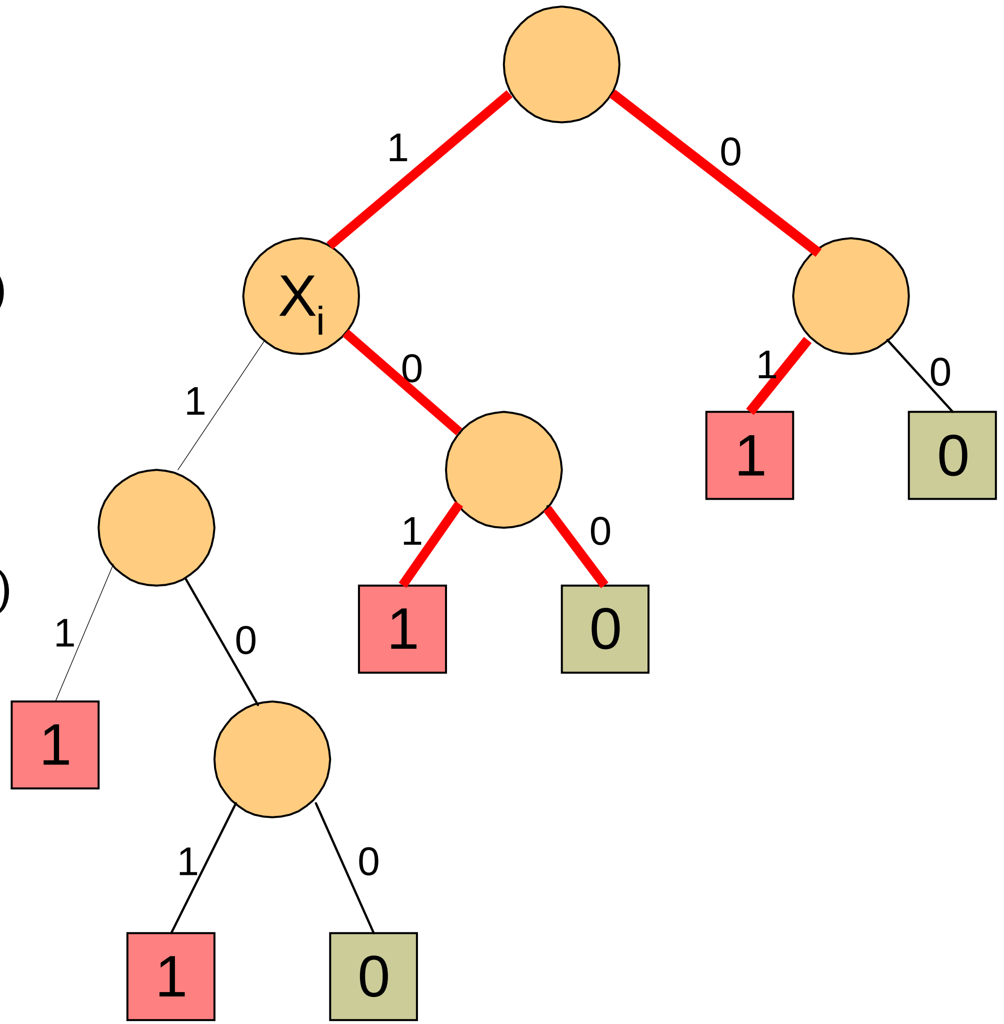


Criticality Function: Routes to a terminal-1

Criticality for X_i

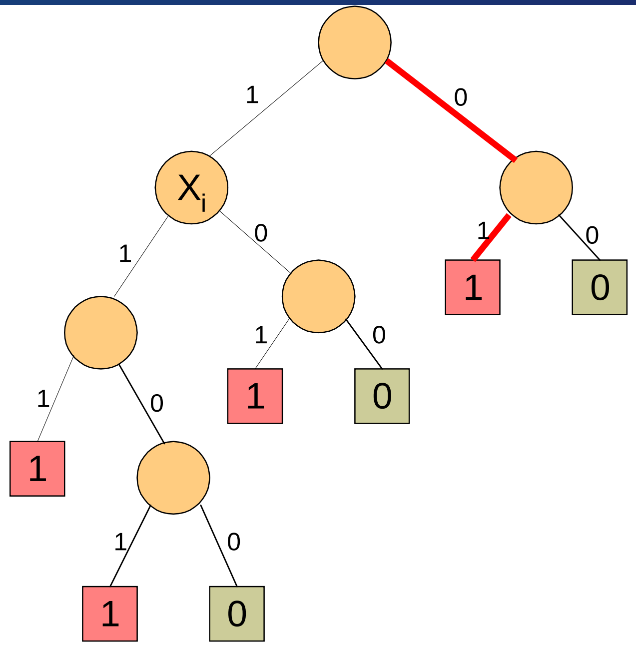
$$Q_{SYS}\left(1_i, \underline{q}\right) = \sum_{x_{i_1} \in path_j} P\left(path_j - x_{i_1}\right) + \sum_{x_i \notin path_j} P(path_j | x_i = 1)$$

$$Q_{SYS}\left(0_i, \underline{q}\right) = \sum_{x_{i_0} \in path_j} P\left(path_j - x_{i_0}\right) + \sum_{x_i \notin path_j} P(path_j | x_i = 0)$$



Criticality Function

Criticality for X_i (X_i in dependency group d)



$$G_i(\boldsymbol{q}) = \sum_{x_{i_1} \in path_j} P\left(path_j - x_{i_1}\right) + \sum_{x_i \notin path_j} P(path_j | x_i = 1)$$

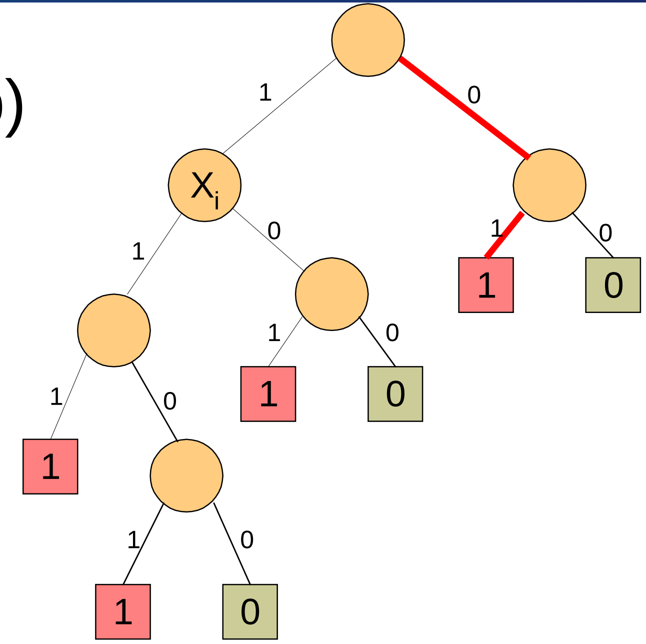
$$- \sum_{x_{i_0} \in path_j} P\left(path_j - x_{i_0}\right) - \sum_{x_i \notin path_j} P(path_j | x_i = 0)$$

$$G_i(\boldsymbol{q}) = \sum_{x_{i_1} \in path_j} \left[P\left(Ipath_j\right). \prod_{\substack{k=1 \\ k \neq d}}^{ndep} [P(Dpath_j^k)] \,.\, P(Dpath_j^d - x_{i_1} | x_i = 1) \right] + \sum_{x_i \notin path_j} \left[P\left(Ipath_j\right). \prod_{\substack{k=1 \\ k \neq d}}^{ndep} [P(Dpath_j^k)] \,.\, P(Dpath_j^d \, | x_i = 1) \right]$$

$$- \sum_{x_{i_0} \in path_j} \left[P\left(Ipath_j\right). \prod_{\substack{k=1 \\ k \neq d}}^{ndep} [P(Dpath_j^k)] \,.\, P(Dpath_j^d - x_{i_0} | x_i = 0) \right] - \sum_{x_i \notin path_j} \left[P\left(Ipath_j\right). \prod_{\substack{k=1 \\ k \neq d}}^{ndep} [P(Dpath_j^k)] \,.\, P(Dpath_j^d \, | x_i = 0) \right]$$

Criticality Function

Criticality for X_i (X_i not an element of a dependency group)



$$G_i(\boldsymbol{q}) = \sum_{x_{i_1} \in path_j} P(path_j - x_{i_1}) + \sum_{x_i \notin path_j} P(path_j)$$

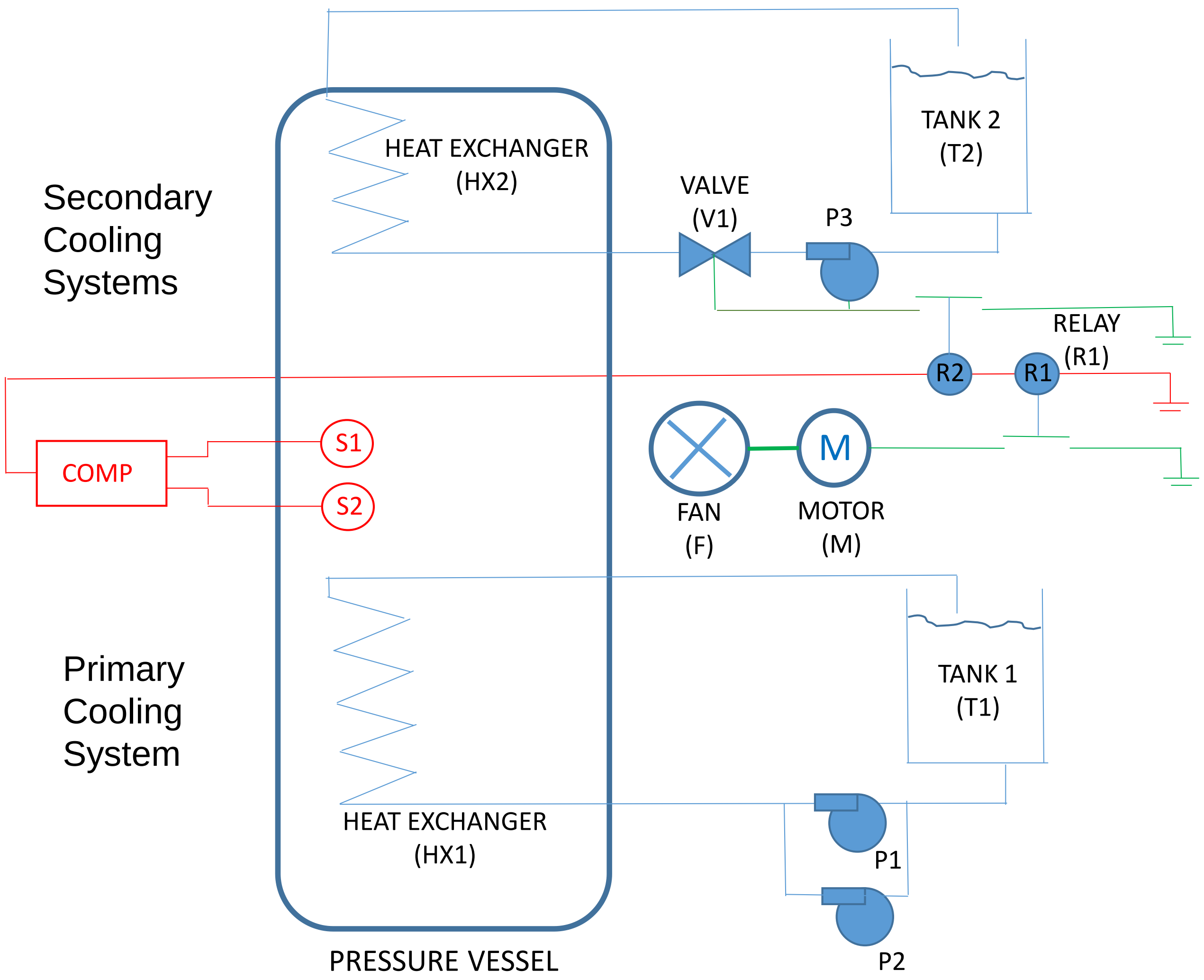
$$- \sum_{x_{i_0} \in path_j} P(path_j - x_{i_0}) - \sum_{x_i \notin path_j} P(path_j)$$

$$G_i(\boldsymbol{q}) = \sum_{x_{i_1} \in path_j} \left[P(Ipath_j - x_{i_1}) . \prod_{k=1}^{ndep} [P(Dpath_j^k)] \right] - \sum_{x_{i_0} \in path_j} \left[P(Ipath_j - x_{i_0}) . \prod_{k=1}^{ndep} [P(Dpath_j^k)] \right]$$

University of Nottingham
UK | CHINA | MALAYSIA

Case Study

Plant Cooling System and Features

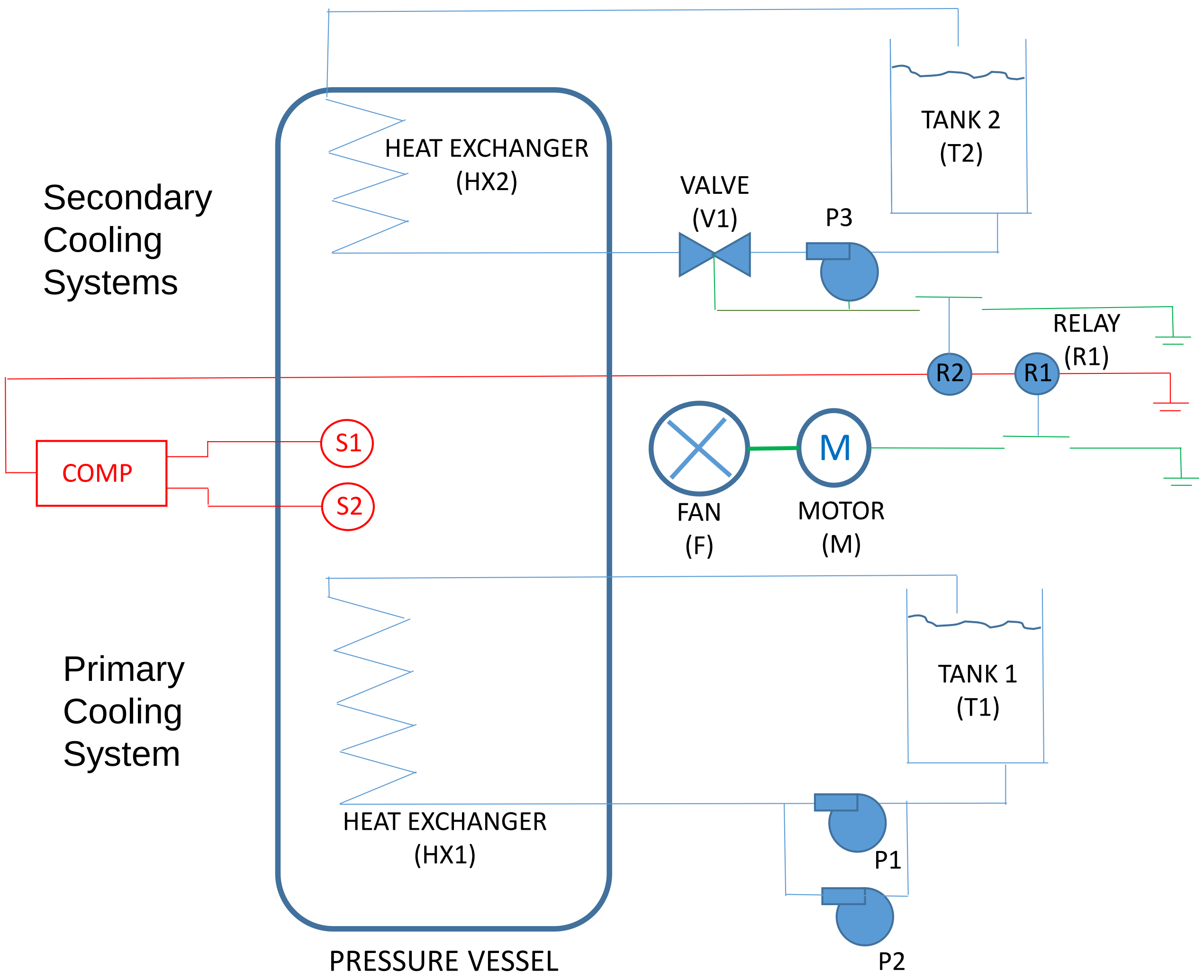


Power supply to all pumps and the valve – B1

Sub-Systems

- Primary Cooling Water System
  - Tank (T1), Pumps (P1,P2), Heat Exchanger (Hx1), Power Supply (B1)
- Detection System
  - Sensors (S1,S2), Computer (Comp)
- Secondary Cooling Water System
  - Tank(T2), Pump (P3), Heat Exchanger (Hx2), Valve (V1), Relay (R2), Power Supply (B1)
- Secondary Cooling Fan System
  - Fan (F), Motor (M), Relay (R1)

Plant Cooling System and Features

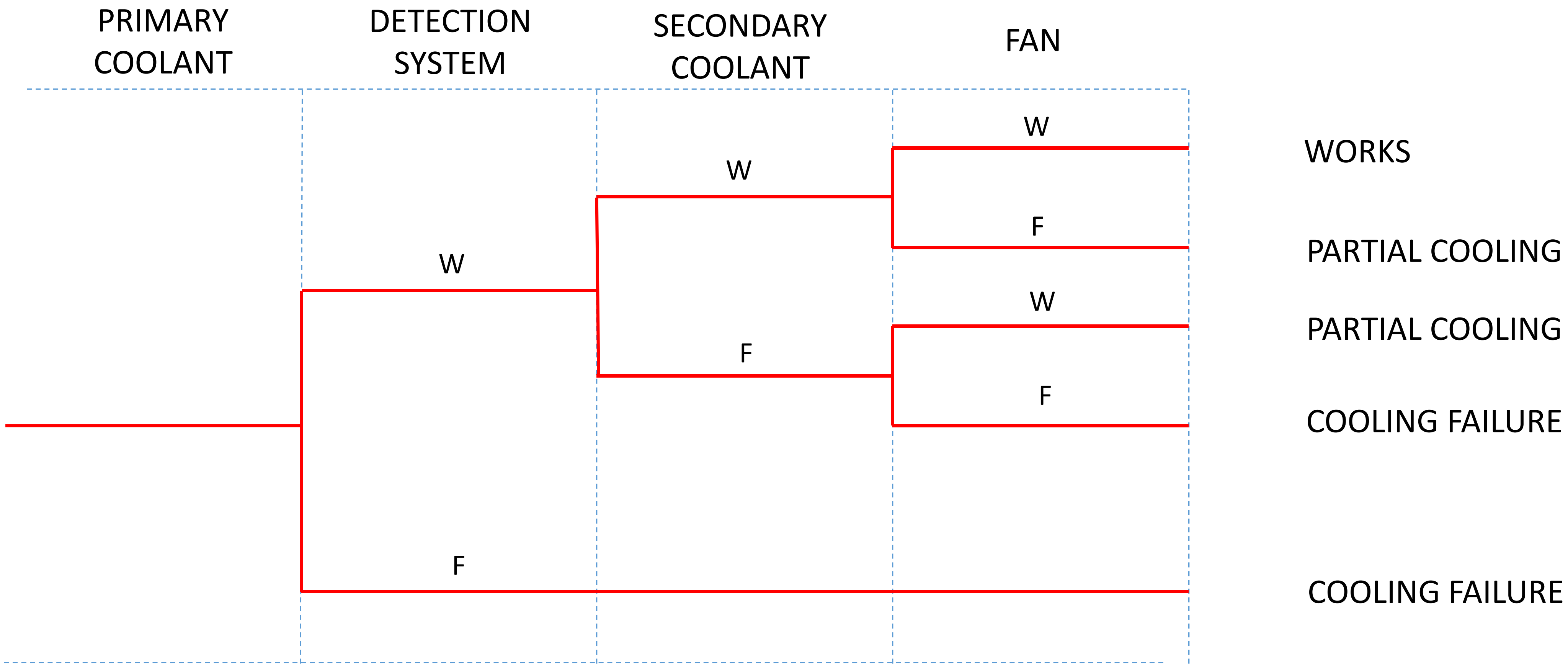


Power supply to all pumps and the valve – B1

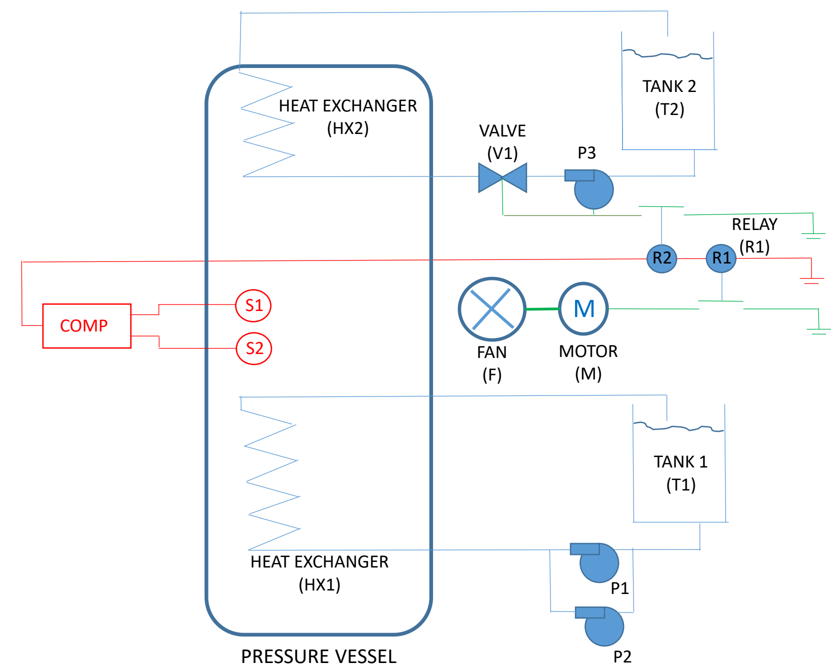
Complex Features

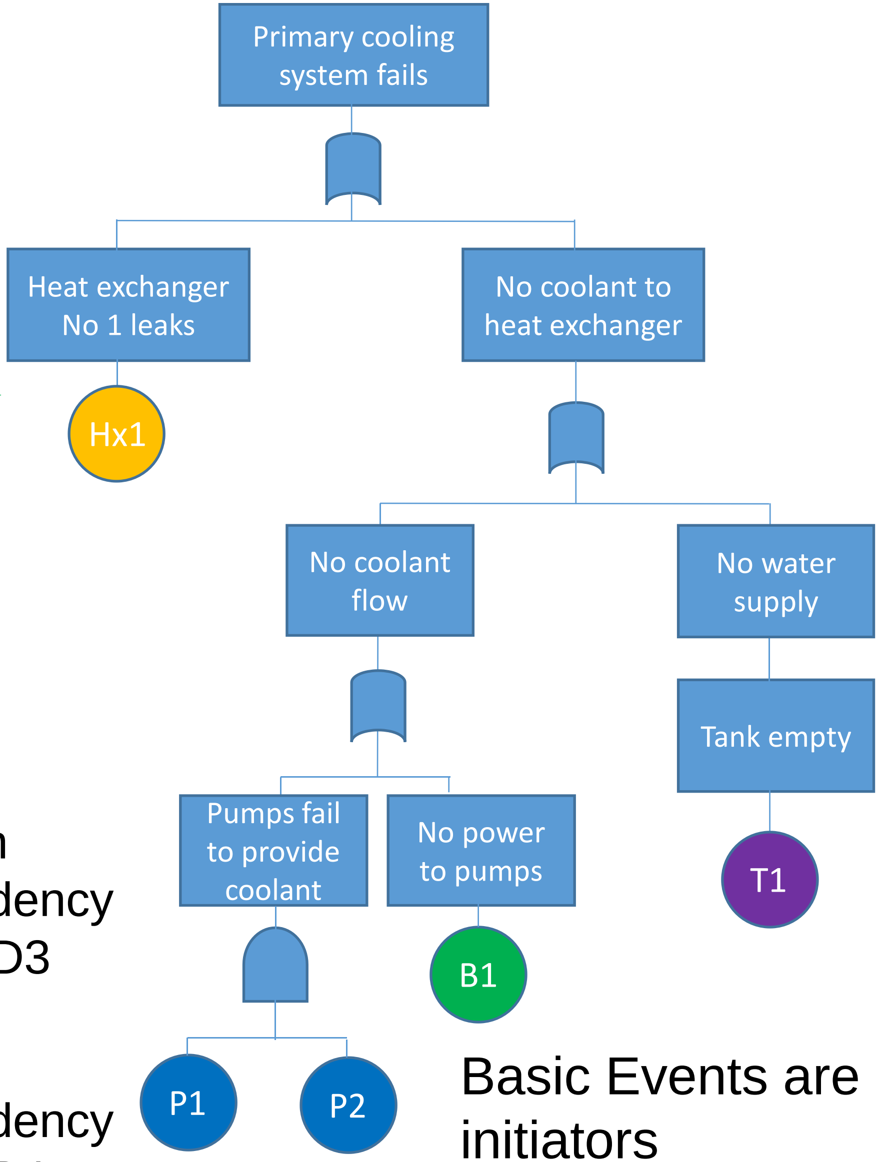
- Non-constant failure / repair rates
  - Relays R1 & R2 have a Weibull failure time distribution and a lognormal repair time distribution
- Dependencies
  - Pumps P1 & P2 – if one fails it puts increased load (and increases the failure rate) of the other
  - Sensors, S1 and S2 have a common cause calibration failure
  - Tanks T1 and T2, when one fails both are replaced
- Maintenance process
  - The motor, M, has a condition monitoring system with different maintenance actions depending on the condition state.

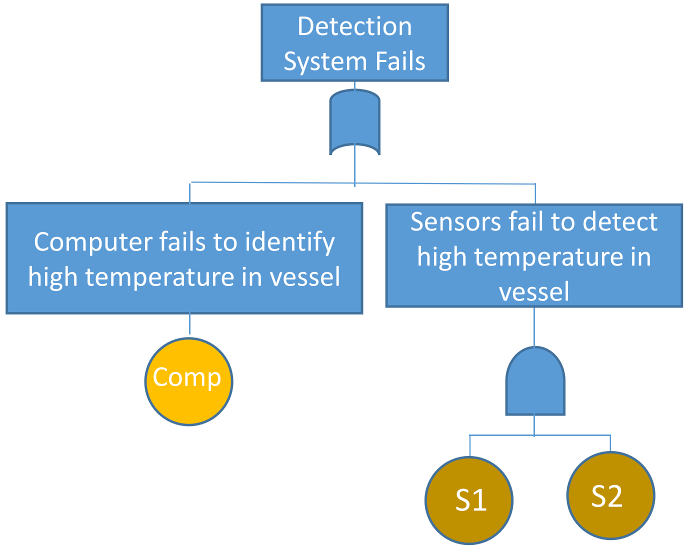
Event Tree Analysis



Fault Tree – Primary Cooling Water System







T1 is in dependency group D3

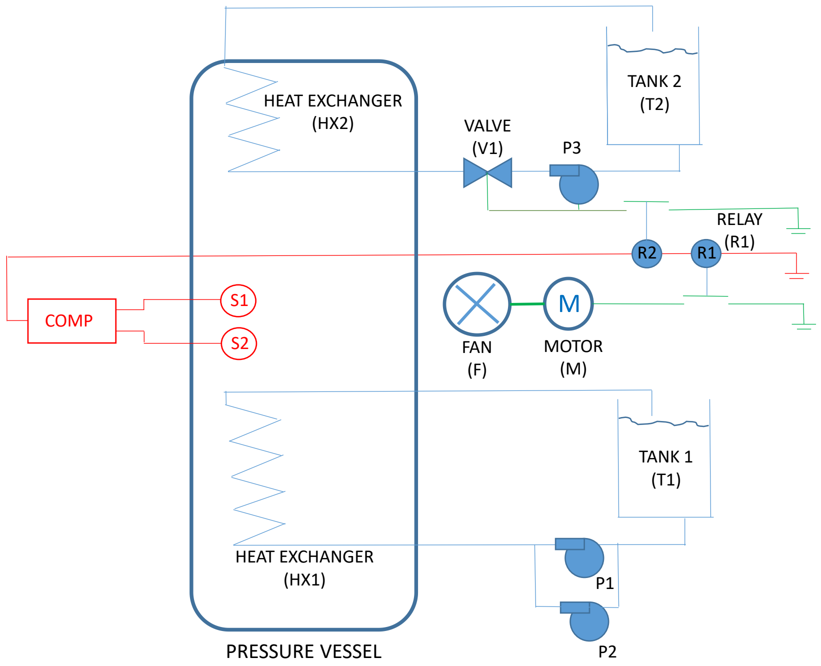
P1, P2 dependency group D1

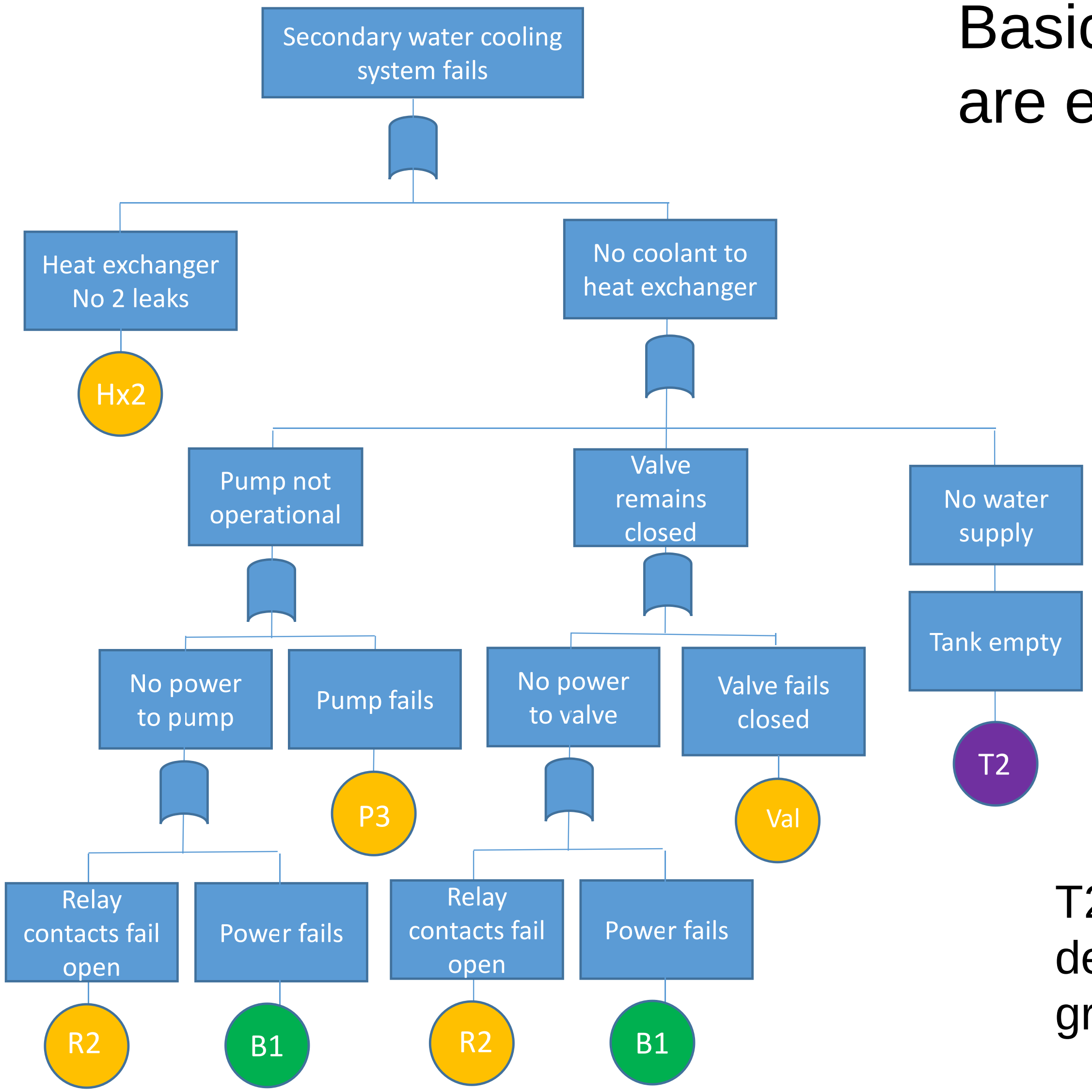
Basic Events are initiators

Basic Events are enablers

S1, S2 dependency group D2

Fault Tree – Secondary Cooling Water System

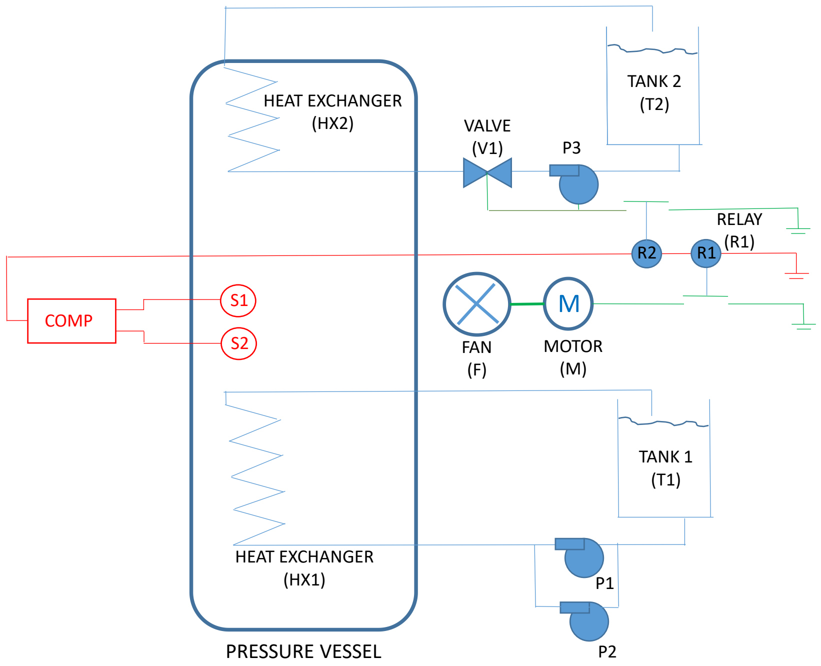


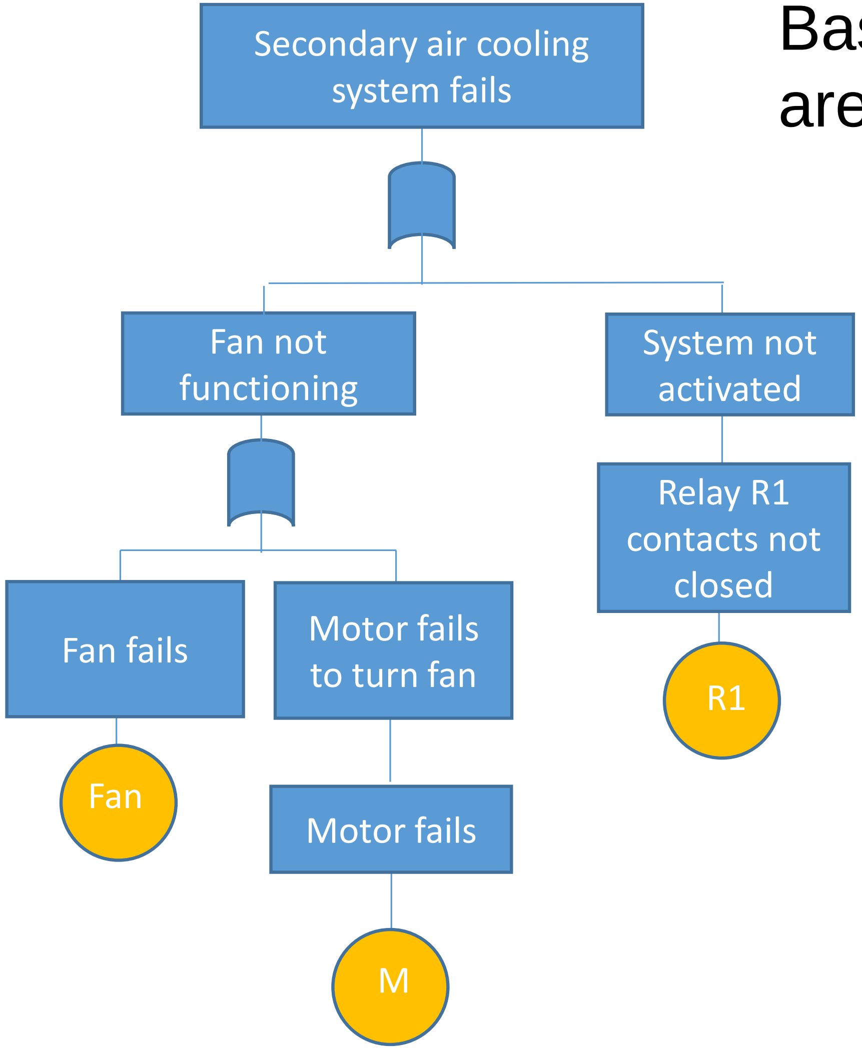


Basic Events are enablers

T2 is in dependency group D3

Fault Tree – Fan Cooling System





Basic Events are enablers

University of Nottingham
UK | CHINA | MALAYSIA

Step 1

Calculate simple component failure models

Simple Component Failure Models

Revealed Failures - initiators

Component	Code	Failure rate (λ) Per year	Mean time to repair (τ) years	Failure Probability $q=\frac{\lambda}{\lambda+\nu}$	Failure Intensity w=λ(1-q)
Heat Exchanger	HX1	0.125	5.5×10^{-3}	6.8703×10^{-4}	0.1249
Power Supply	B1	0.5	2.5×10^{-3}	1.248×10^{-3}	0.4994

Unrevealed Failures - enablers

Component	Code	Failure rate (λ) Per year	Mean time to repair (τ) years	Inspection int (θ) years	q=λ(θ/2+τ)
Heat Exchanger	HX2	0.125	5.5×10^{-3}	1	0.06319
Computer	Comp	0.4	5.0×10^{-3}	0.08	0.034
Pump	P3	0.05	0.08333	0.5	0.01667
Fan	Fan	0.06	5.0×10^{-3}	0.5	0.0153

University of Nottingham
UK | CHINA | MALAYSIA

Step 2

Build and analyse the complexity/dependency models

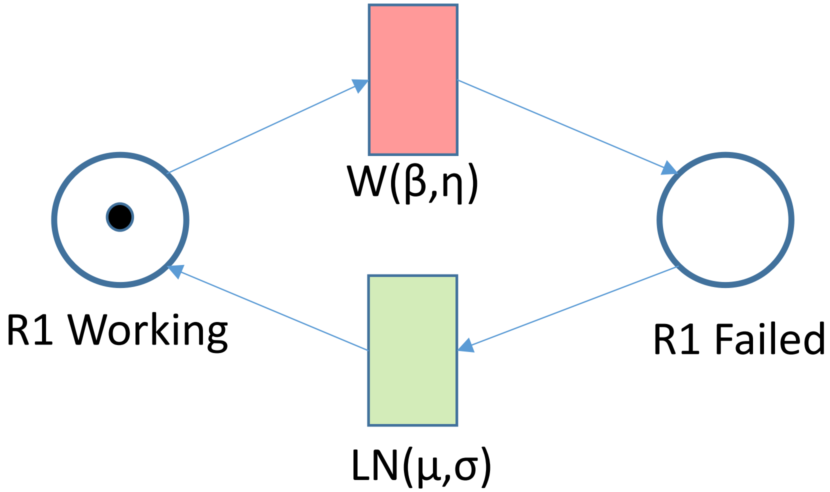
Complexity models

Relays R1 & R2

Non-constant failure / repair rates

Weibull failure time distribution

lognormal repair time distribution



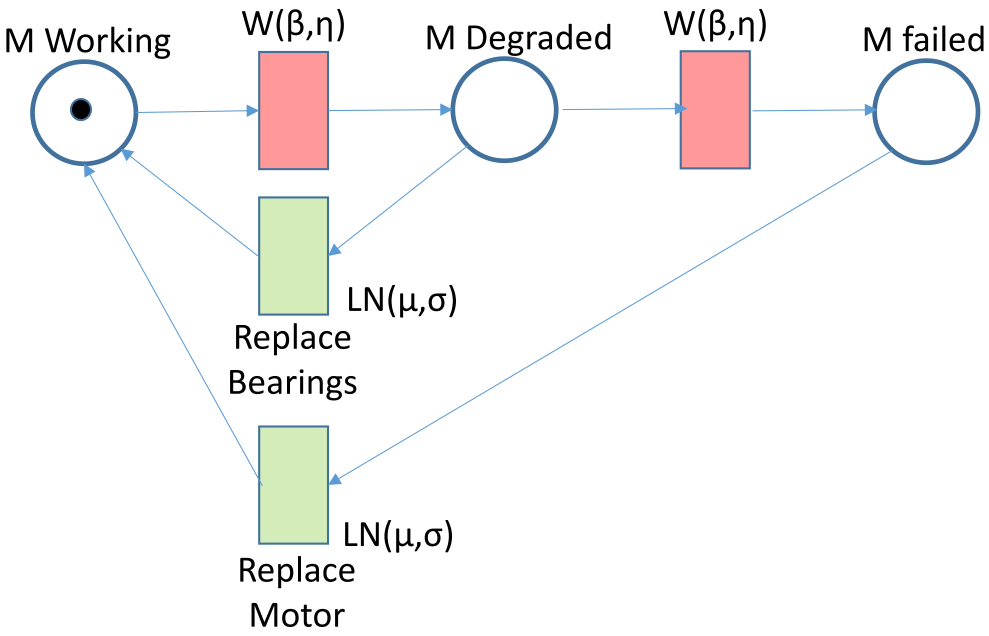
q_{R1}

q_{R2}

Motor M

Maintenance process

a condition monitoring system with different maintenance actions depending on the condition state.

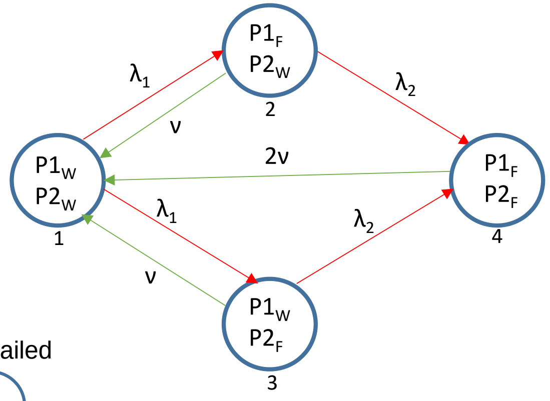


q_M

University of Nottingham
UK | CHINA | MALAYSIA

Dependency models

Pumps P1 & P2
if one fails it puts increased load on the other



P1 & P2 Initiators

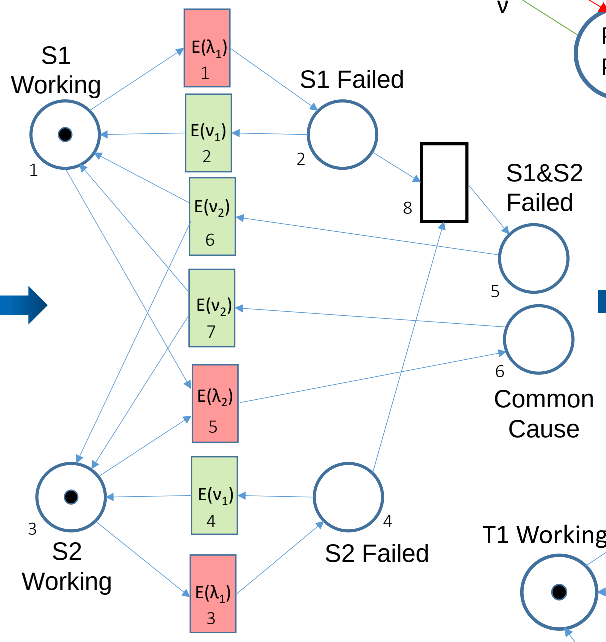
$q_{P1.P2}$ $w_{QP1.P2}$

$q_{\overline{P1}.P2}$ $w_{Q\overline{P1}.P2}$

$q_{P1.\overline{P2}}$ $w_{P1.Q\overline{P2}}$

$q_{\overline{P1}.\overline{P2}}$ $w_{P1.QP2}$

Sensors S1 & S2
common cause calibration failure



S1 & S2 Enablers

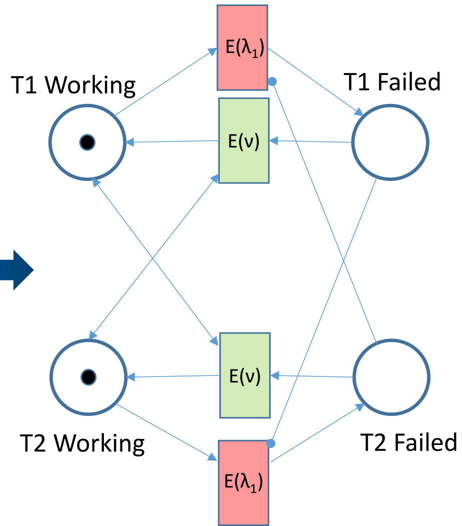
$q_{S1.S2}$

$q_{\overline{S1}.S2}$

$q_{S1.\overline{S2}}$

$q_{\overline{S1}.\overline{S2}}$

Tanks T1 & T2
common cause calibration failure



T1 Initiator
T2 Enablers

$q_{T1.T2}$ $w_{QT1.T2}$

$q_{\overline{T1}.T2}$ $w_{Q\overline{T1}.T2}$

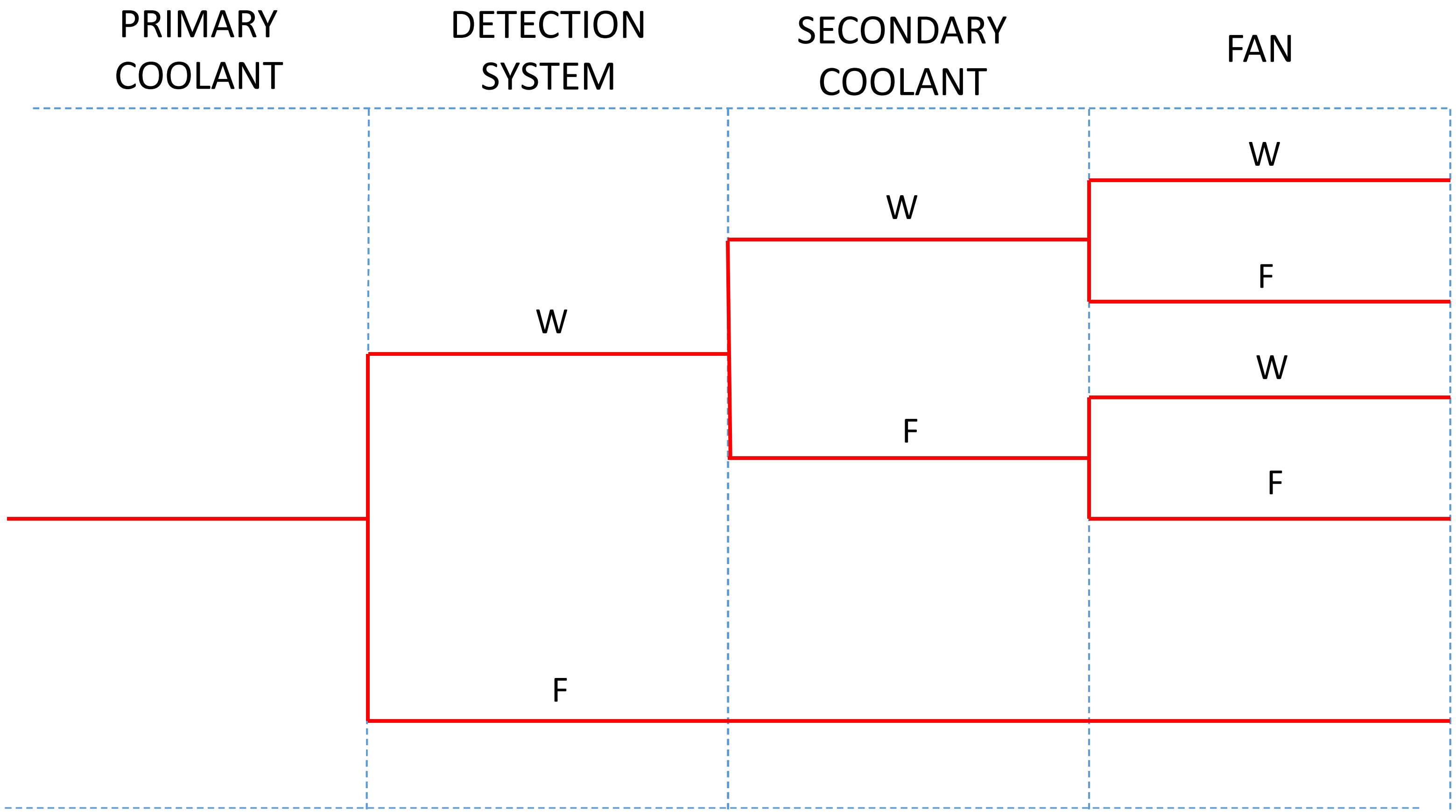
$q_{T1.\overline{T2}}$ $w_{T1.Q\overline{T2}}$

$q_{\overline{T1}.\overline{T2}}$ $w_{T1.QT2}$

Step 3

Construct and Analyse the BDDs required to give each Event Tree outcome

Event Tree Analysis



5 $P(DET_w)P(FAN_w)w(SEC_w.PC_f)$

4 $P(DET_w)P(FAN_f)w(SEC_w.PC_f)$

3 $P(DET_w)P(FAN_w)w(SEC_f.PC_f)$

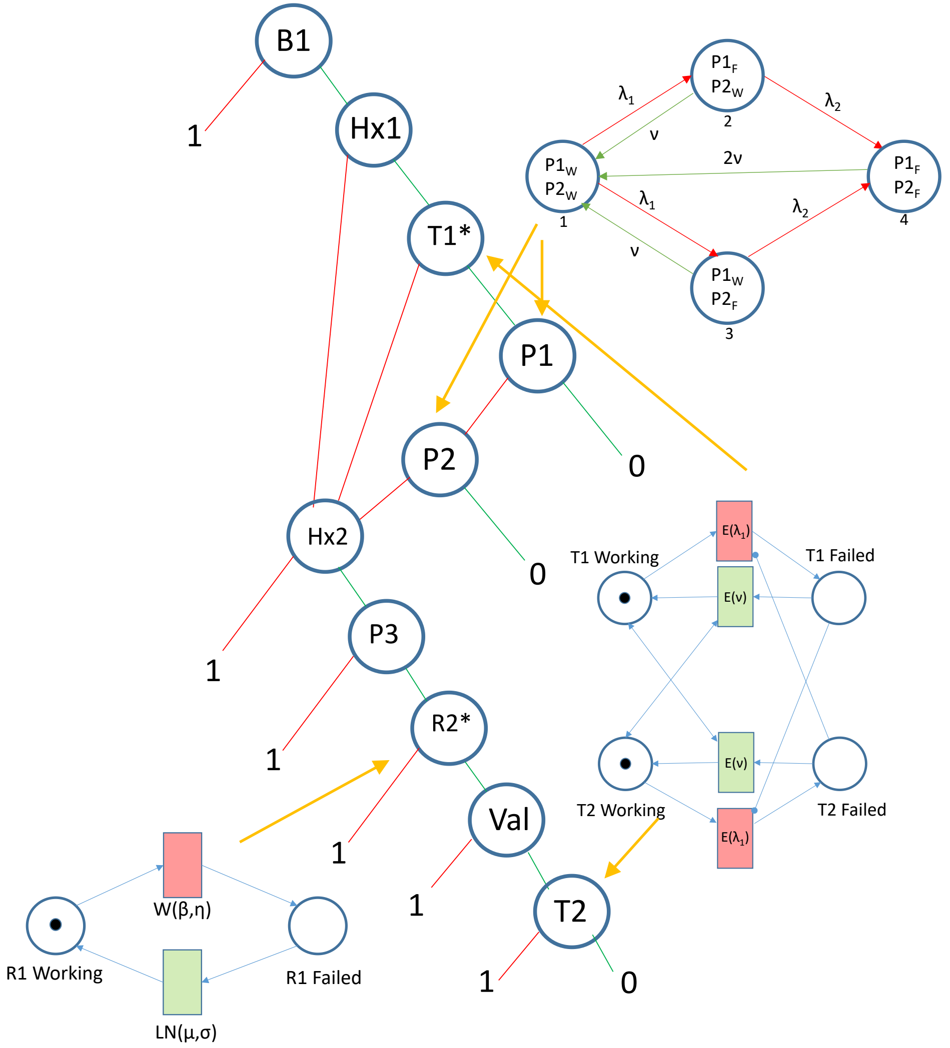
2 $P(DET_w)P(FAN_f)w(SEC_f.PC_f)$

1 $P(DET_f).w(PC_f)$

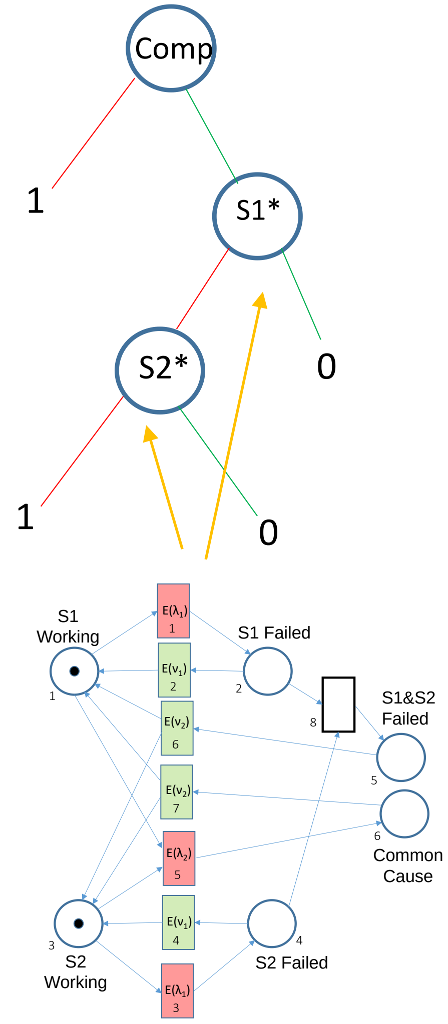
Basic event B1 and dependency group with T1 & T2 in common

BDD Independent Modules

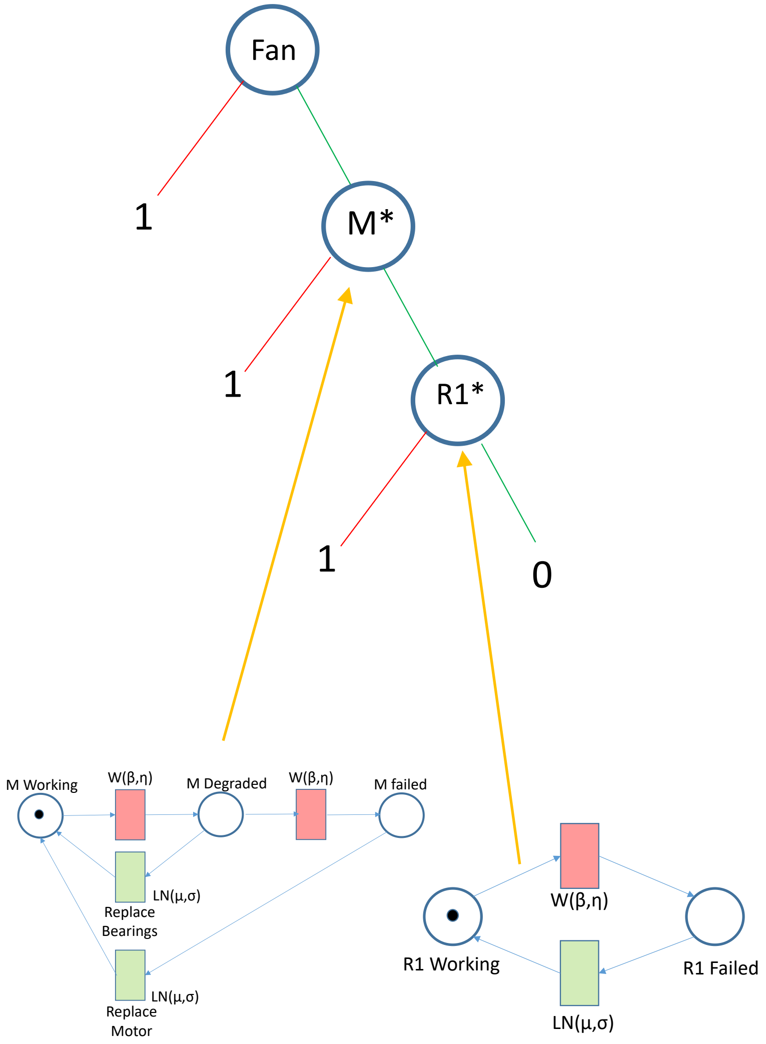
Failure of the Primary and Secondary Cooling Systems



Failure of the Detection System



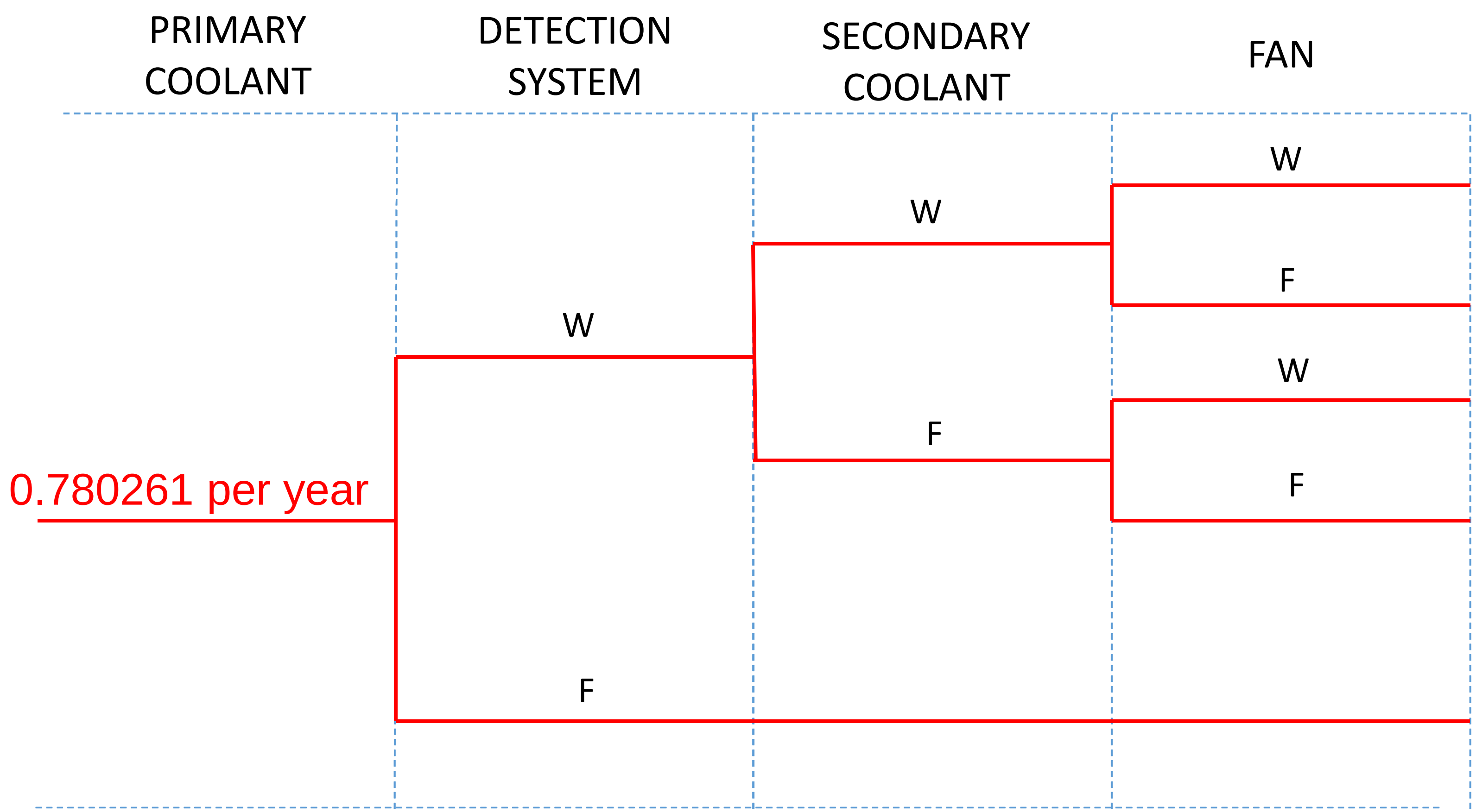
Failure of the Fan Cooling System



Step 4

Quantify each Event Tree outcome

Repeating this process for all other events



5 $P(DET_w)P(FAN_w)w(SEC_w.PC_f)$ = 8.31126E-7 per year

4 $P(DET_w)P(FAN_f)w(SEC_w.PC_f)$ = 3.63607E-8 per year

3 $P(DET_w)P(FAN_w)w(SEC_f.PC_f)$ = 0.648495 per year

2 $P(DET_w)P(FAN_f)w(SEC_f.PC_f)$ = 0.028371 per year

1 $P(DET_f).w(PC_f)$ = 0.103395 per year

$P(DET_f) = 0.132513$

$P(FAN_f) = 0.041915$

$w(SEC_w.PC_f) = 1.0e-6$

$w(PC_f) = 0.780261$ per year

$w(SEC_f.PC_f) = 0.780260$

University of Nottingham
UK | CHINA | MALAYSIA

Summary / Conclusions

- First Phase of the NxGen project has been described
- This incorporates the following features into the modelling
  - Dependencies
  - Non-constant failure and repair rates
  - Complex maintenance strategies
- A method has been developed which enables results from the PN/Markov models to be integrated into the BDDs
- Current work:
  - Modularisation methods
  - Building dependencies into the phased mission methodology
  - Solving case studies
    - aero – engine air cooling system
    - railway – derailment
    - nuclear - LOCA

Thank you for your attention

Any Questions?