University of Nottingham
UK | CHINA | MALAYSIA

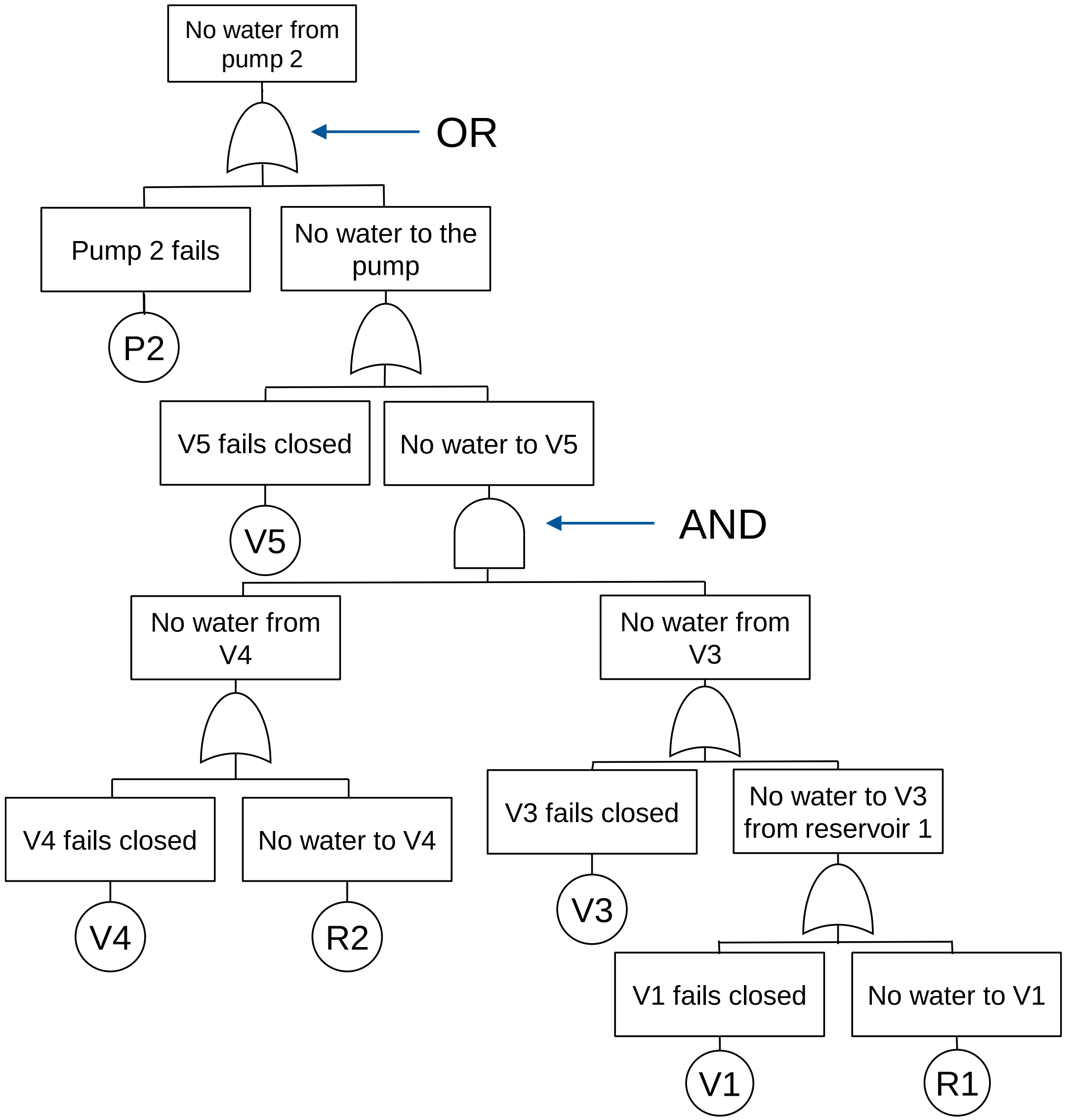
LR Foundation

The Analysis of Fault Trees with Dependent Basic Events

John Andrews

Durham University
4 June 2024

Fault Tree Analysis



Component failure models

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

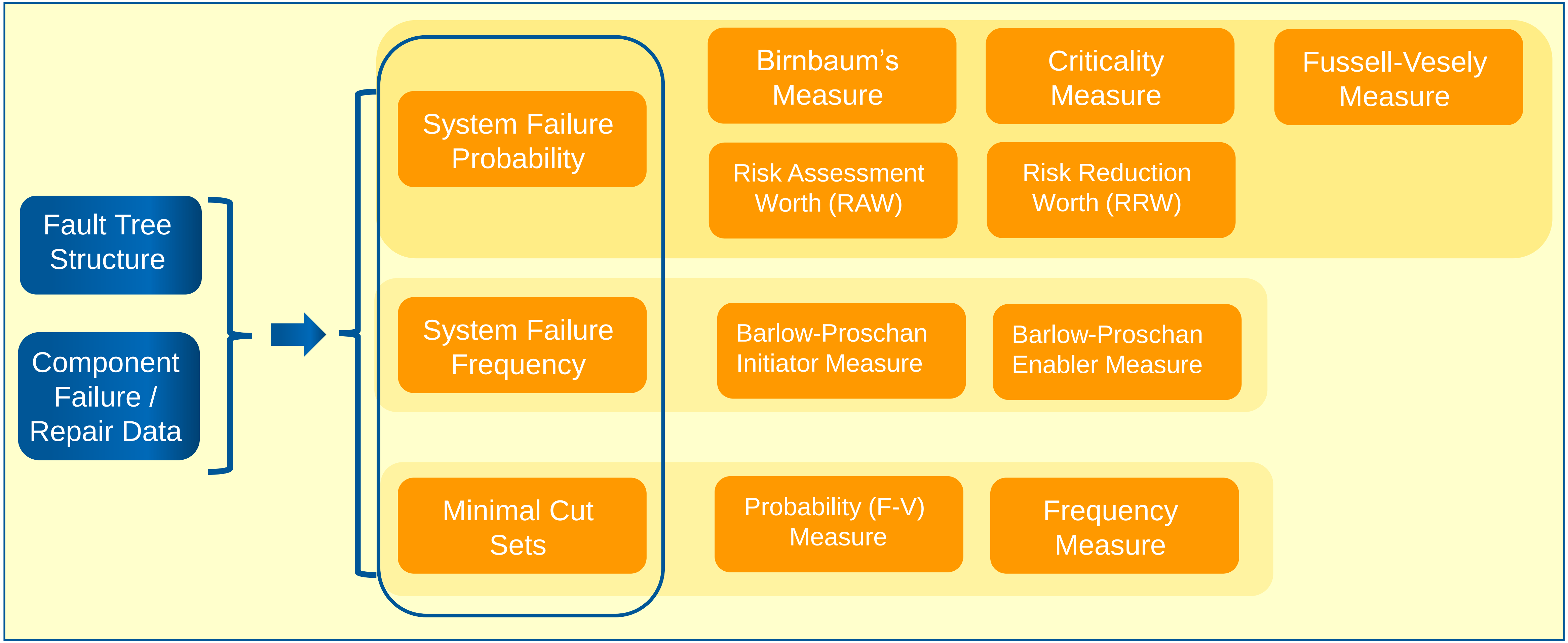
PROJECT AIMS

- Incorporate:
  - non-constant failure and repair rates
  - dependent events
  - highly complex maintenance strategies
  - dynamic features

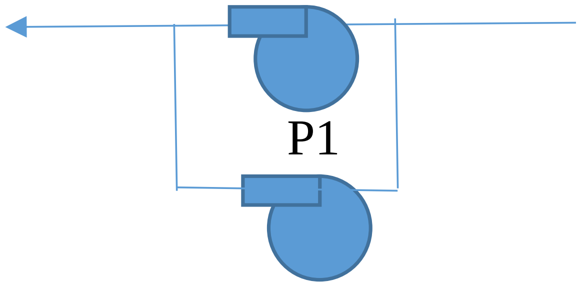
Fault Tree Analysis

System Failure Mode Analysis

Importance Measures



Safety System Analysis - Standby Systems



Standby System

- Pump P1 operational.
- When P1 fails P2 takes over the duty

Hot Standby	Warm Standby	Cold Standby
Both pumps are operational but the fluid is just driven by P1. On failure of P1, the fluid now passes through P2	Pump P2 is not operational in standby. It becomes operational when P1 fails. It can fail in standby but with a lower rate than when operational.	Pump P2 is not operational in standby. It becomes operational when P1 fails. It cannot fail in standby.
P1 & P2 Independent	P1 & P2 Dependent	P1 & P2 Dependent

Dependency Examples

Type	Description
Secondary Failure	When one component fails it increases the load on a second component which then experiences an increased failure rate
Opportunistic Maintenance	A component fails which causes a system shutdown or requires specialist equipment for the repair. The opportunity is taken to do work on a second component which has not failed but is in a degraded state
Common Cause	When one characteristic (eg materials, manufacturing, location, operation, installation maintenance) causes the degraded performance in several components
Queueing	Failed components all needing the same maintenance resource are queued. Then repaired in priority order

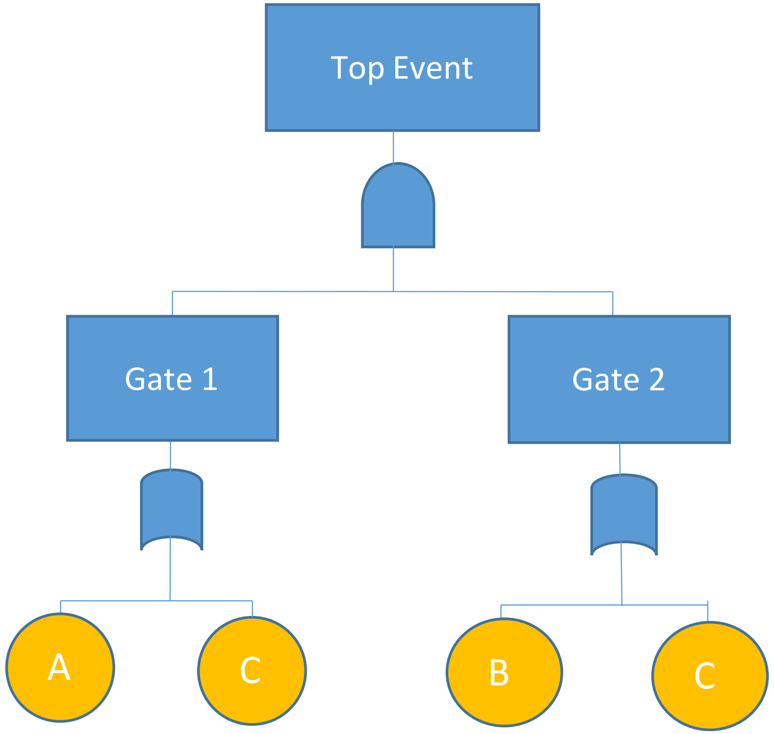
Integration of Fundamental Quantification Methodologies

Fault Tree Analysis => Binary Decision Diagrams (BDD)

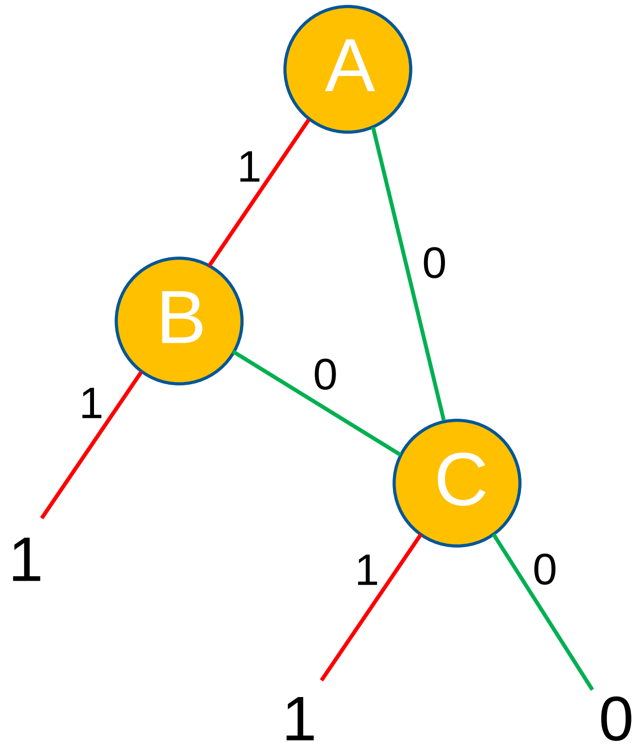
Petri Nets

Markov Methods

Binary Decision Diagrams



ORDERING A < B < C



- Exact
- Fast
- Efficient

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

+ OR
. AND

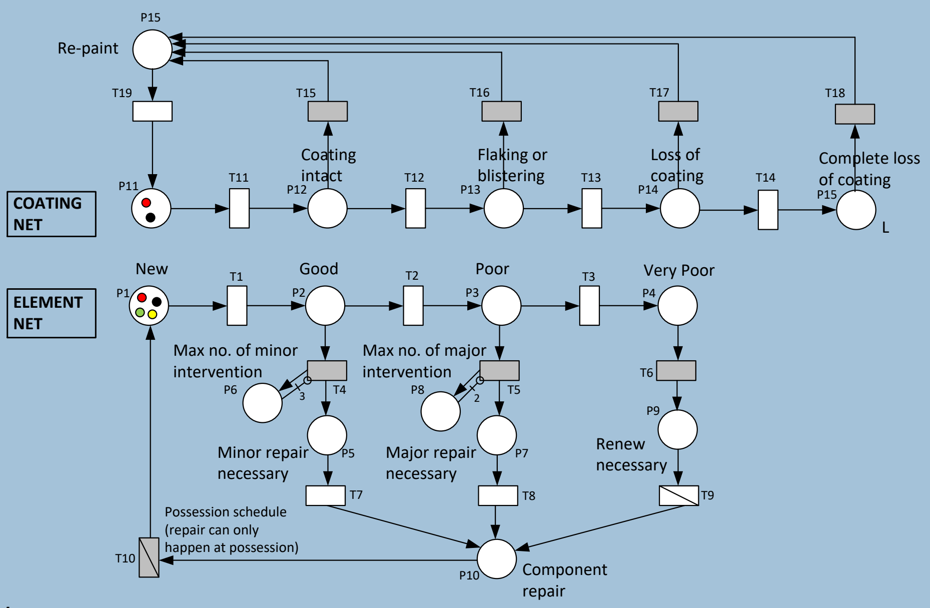
$$TOP = A.B + C$$

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

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

Modelling Methodology

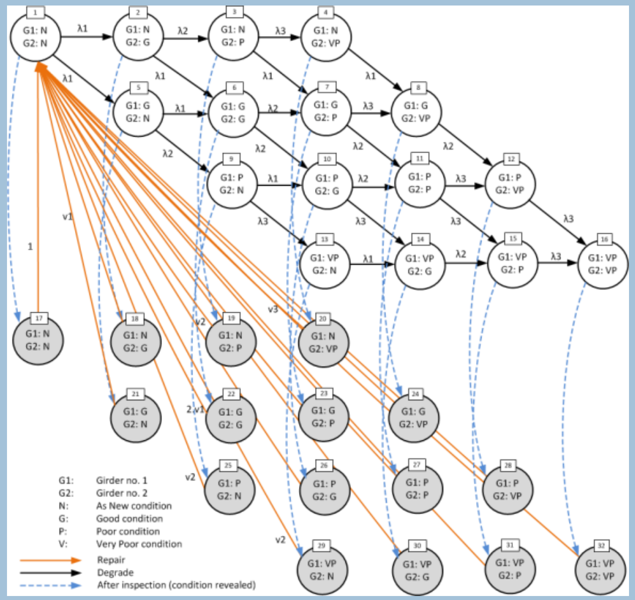
Petri-Net model (1939)



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

Markov model (1906)



Assumes:

- The future condition depends only on the current condition and not the history

Features

- Constant rates of transition
- State-space explosion

University of Nottingham
UK | CHINA | MALAYSIA

Dynamic & Dependent Tree Theory (D^2T^2)

A Fault Tree Analysis Framework

Approach

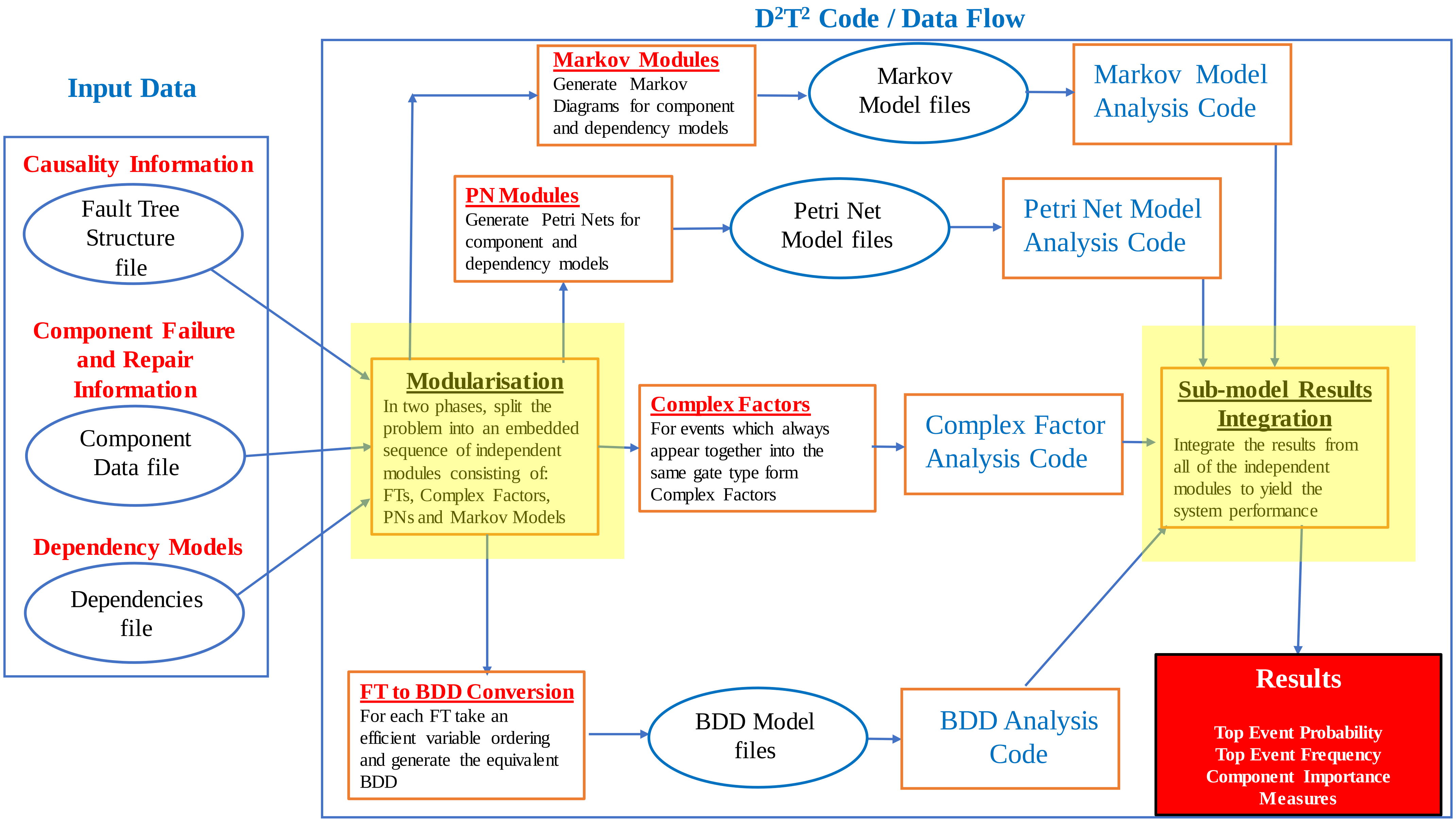
Dependencies

- Model the dependencies and complexities using Petri Nets or Markov models
  - Always use the *simplest dependency model*

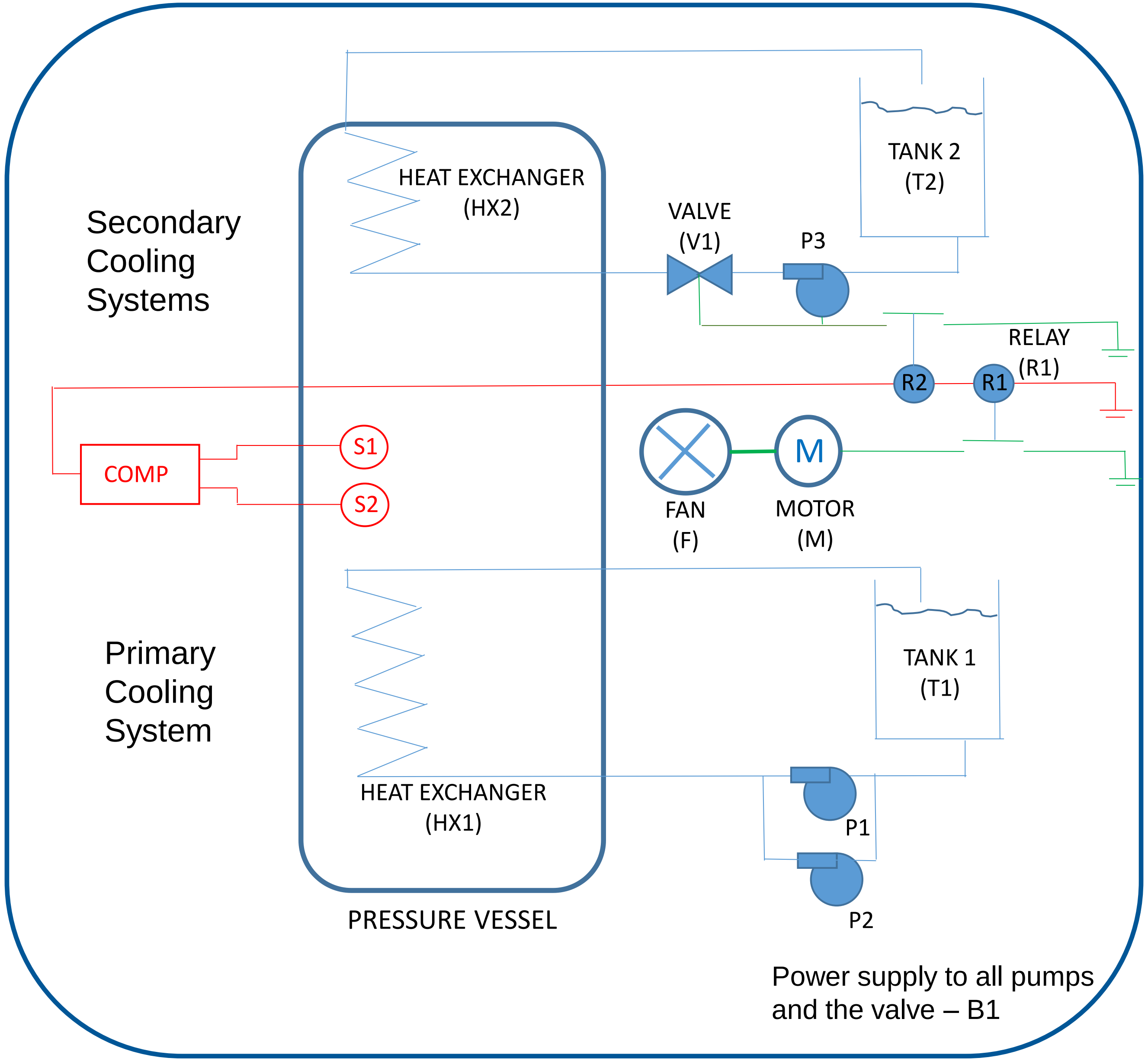
Binary Decision Diagrams

- Dependencies are just required to be considered on each path
- Path numbers can be very high so every effort needs to be made to *minimise the size of the BDD*
  - minimise the fault tree size using an effective modularisation
  - effective variable ordering

Basic Structure of the Code



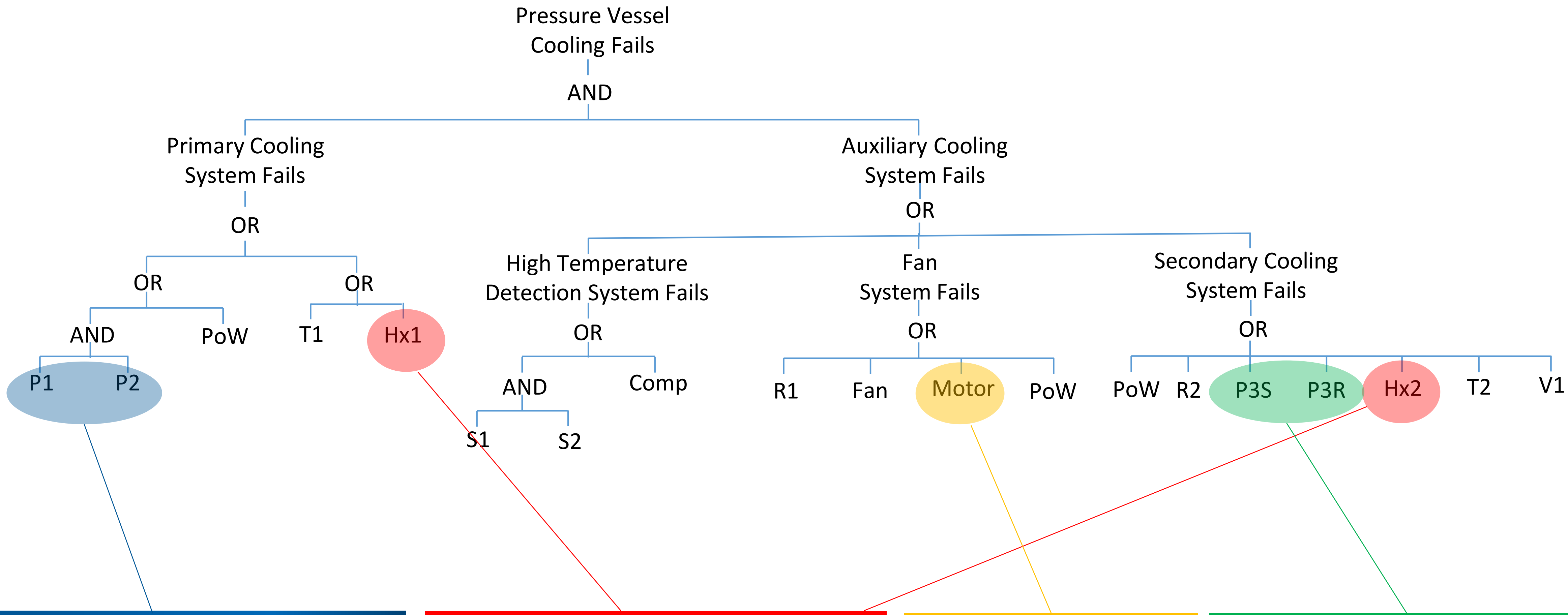
Plant Cooling System and Features



Complex Features

- Non-constant failure / repair rates
  - Motor M - 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
  - Heat Exchangers Hx1 & Hx2 - when one needs replacement – needs specialist equipment and both are replaced
  - Pump P3 - two events P3S and P3R are clearly dependent

Fault Tree Structure and Dependent Events



Pumps P1 & P2 – if one fails it puts increased load (and increases the failure rate) of the other

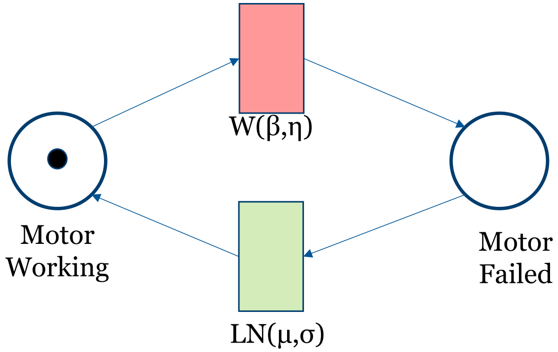
Heat Exchangers Hx1 & Hx2 - when one needs replacement – needs specialist equipment and both are replaced

Non-constant failure / repair rates

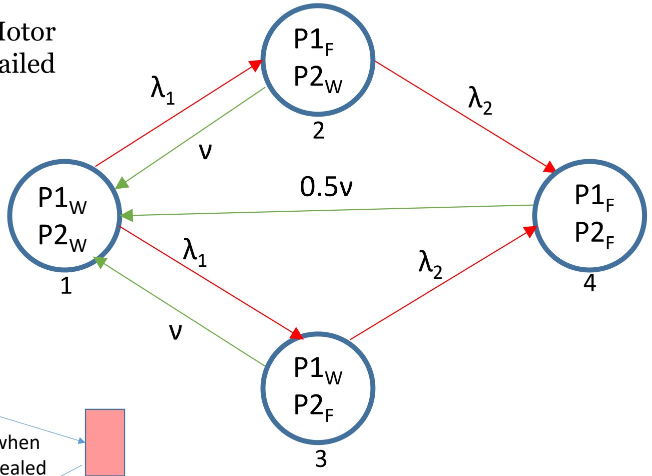
Pump P3 - two events P3S and P3R are clearly dependent

Complexity and Dependency Models

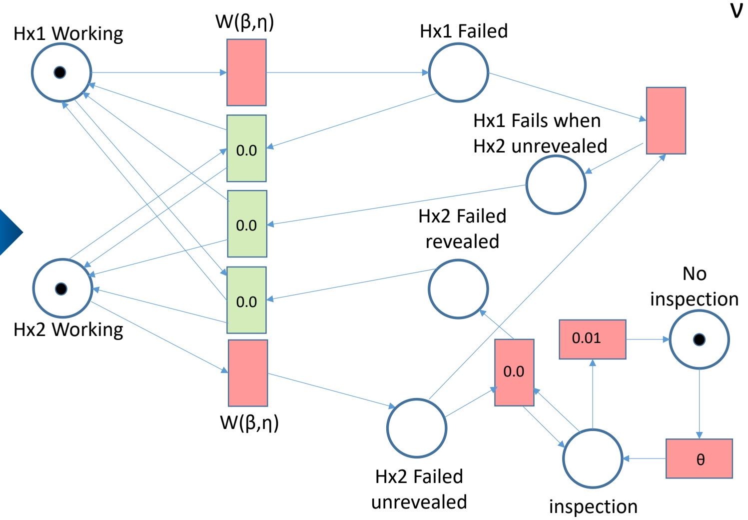
- Non-constant failure / repair rates
  - Motor M - Weibull failure time distribution and a lognormal repair time distribution



- Dependencies
  - Pumps P1 & P2 – if one fails it puts increased load (and increases the failure rate) of the other



- Heat Exchangers Hx1 & Hx2 - when one needs replacement – needs specialist equipment and both are replaced



- Pump P3 - two events P3S and P3R are clearly dependent

$$q_{P3} = q_{P3S} + (1.0 - q_{P3S})\lambda_{P3R}t_{period}$$
$$= 0.05 + 0.095 \times 10^{-4} \times 30$$
$$= 0.05285$$

Modularisation

- Factorisation Method
- Linear-time Algorithm

Modularisation – Reduction Algorithm

- **Contraction**

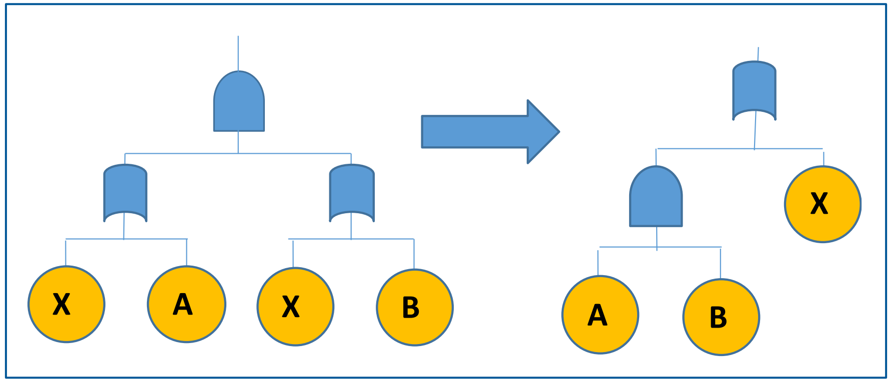
  Subsequent gates of the same type are contracted into a single gate

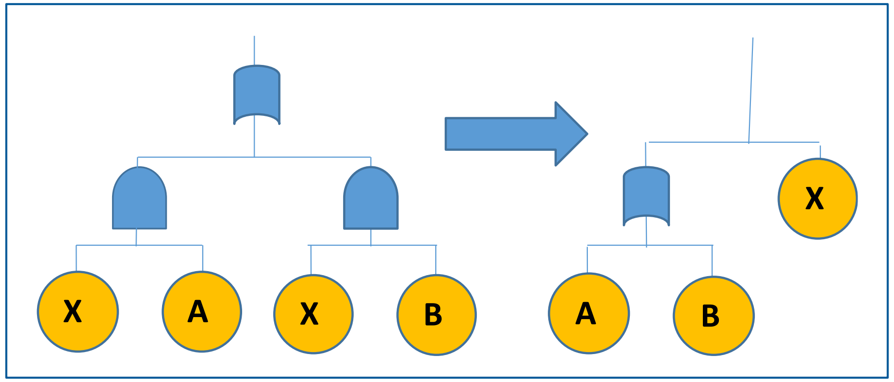
- **Factorisation**

  Identifies factors of groups of events that always occur together in the same gate type.
  The factors can be any number of events if they are all:
  - independent and initiators
  - independent and enablers.
  - a complete dependency group.

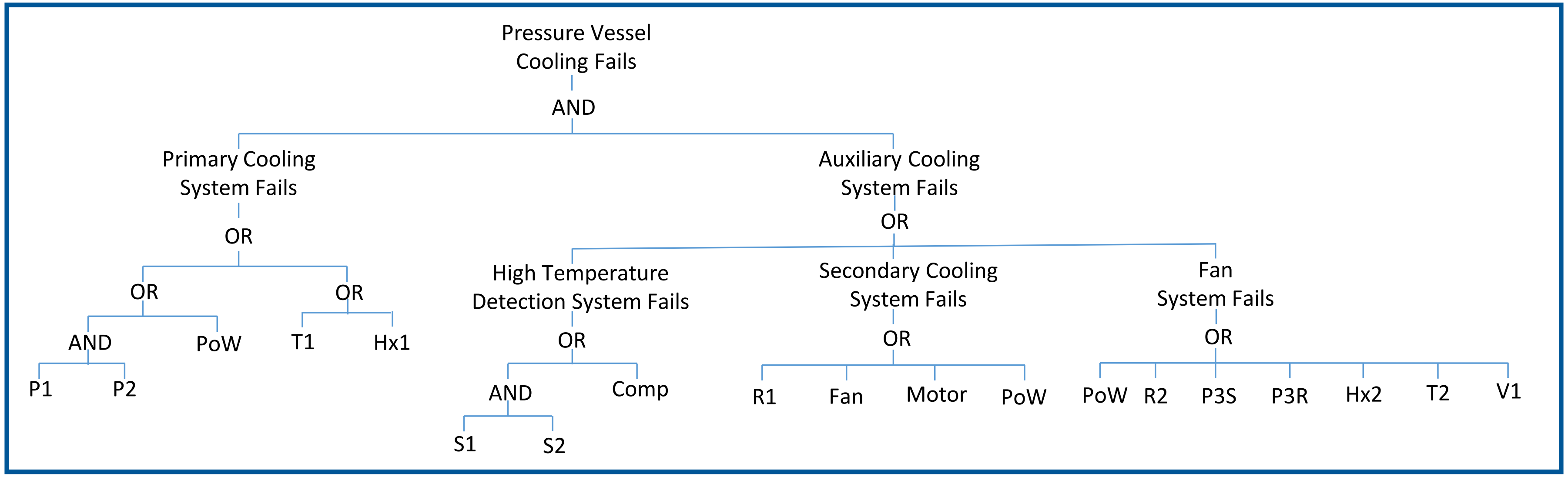
- **Extraction**

  Restructure:

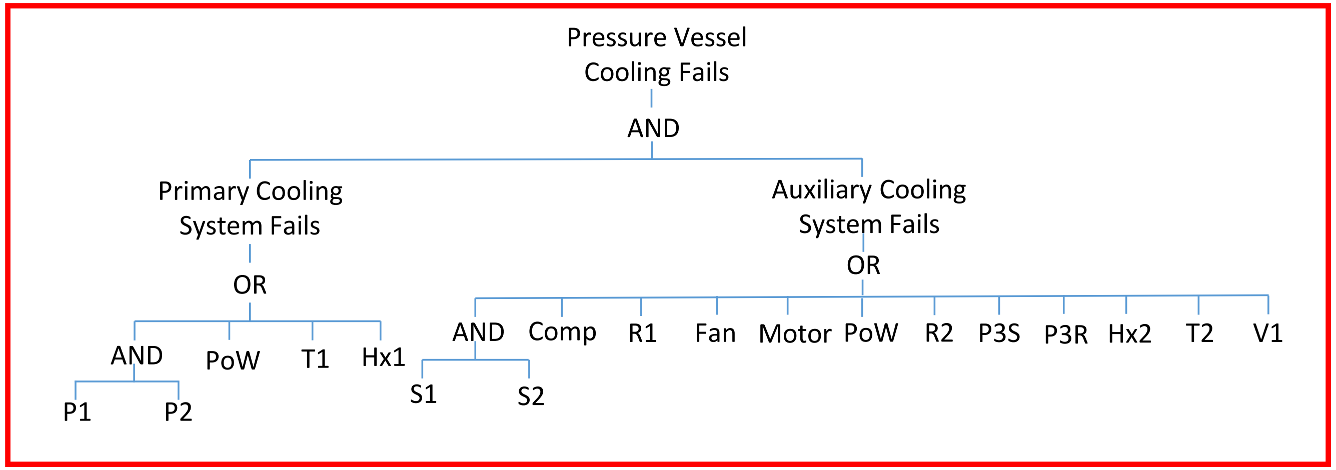




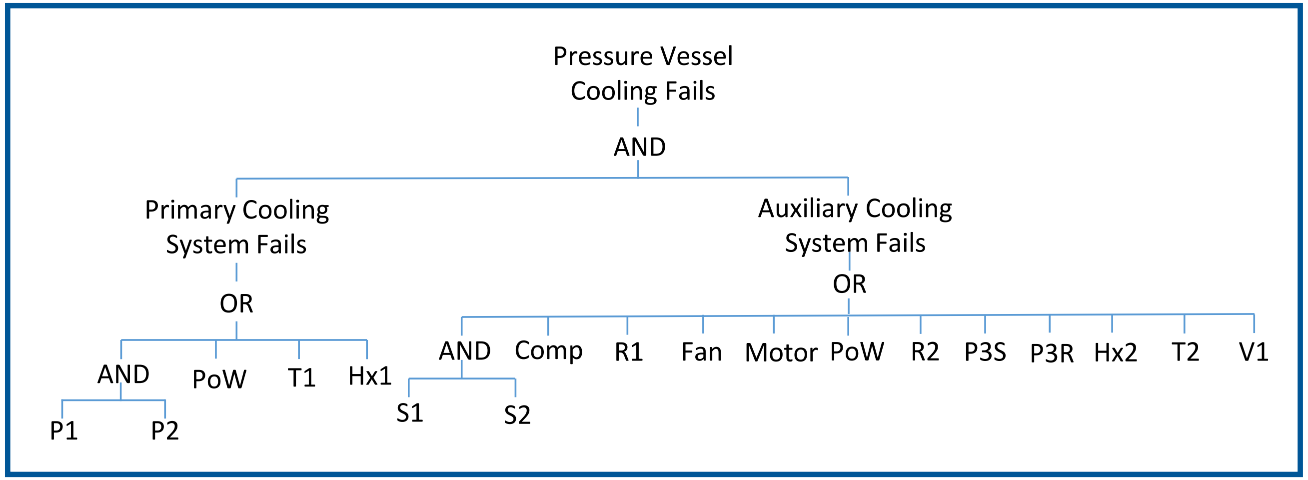
Modularisation (1)



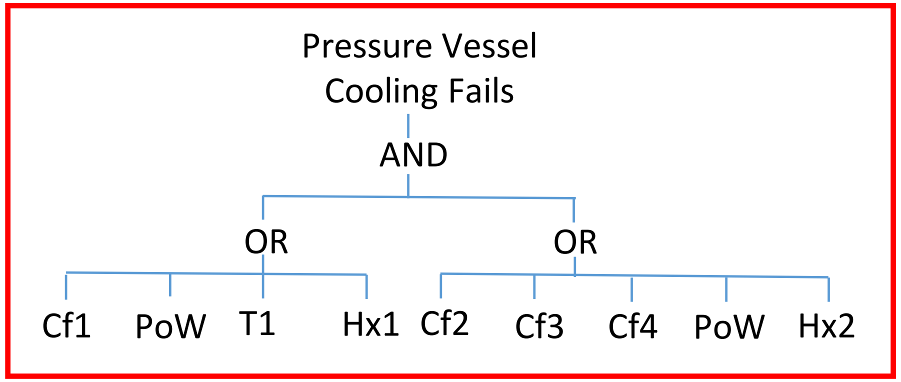
Contraction 1



Modularisation (2)



Factorise 1



$Cf_1 = P1.P2$

(dependency group D1 – initiators)

$Cf_2 = S1.S2$

(independent enablers)

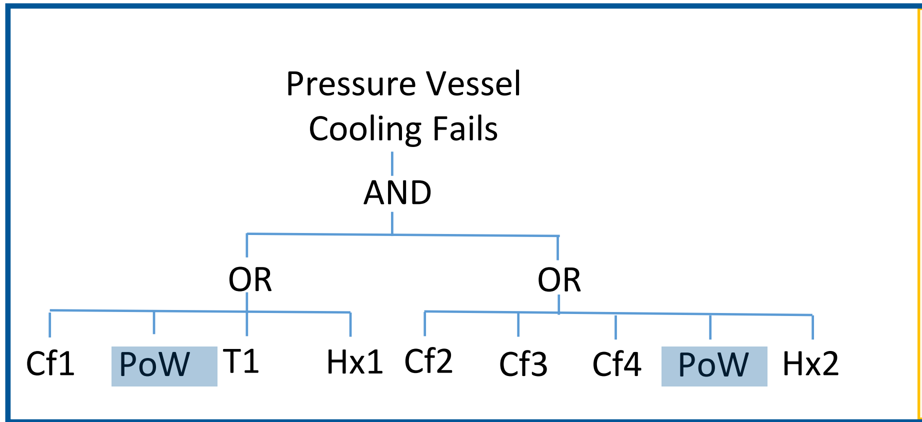
$Cf_3 = Comp + R1 + Fan + Motor + R2 + T2 + V1$

(independent enablers)

$Cf_4 = P3S + P3R$

(dependency group D3 – enablers)

Modularisation (3)



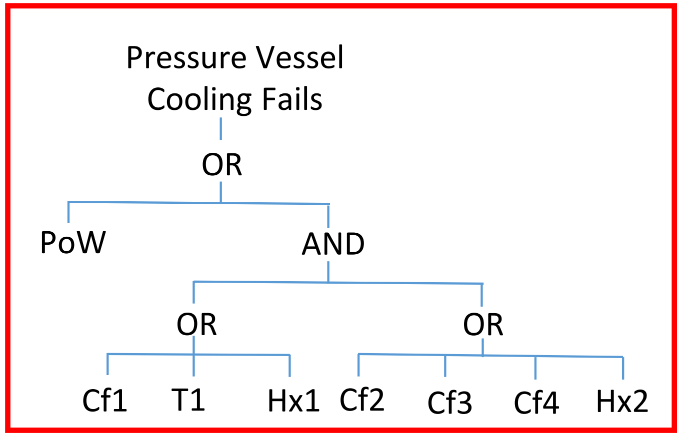
$$Cf_1 = P1.P2$$

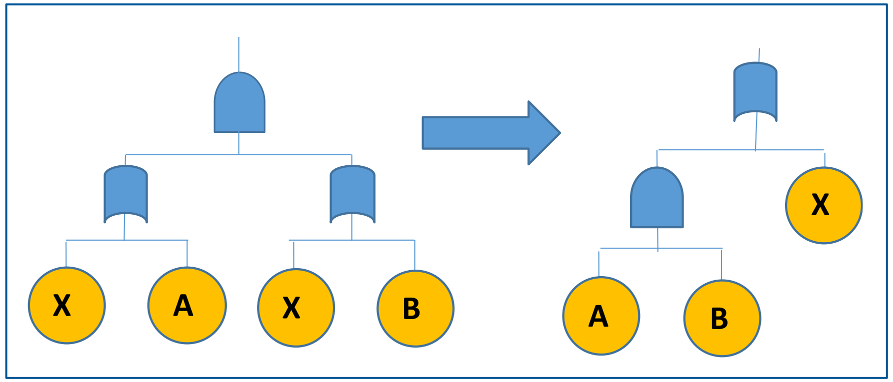
$$Cf_2 = S1.S2$$

$$Cf_3 = Comp + R1 + Fan + Motor + R2 + T2 + V1$$

$$Cf_4 = P3S + P3R$$

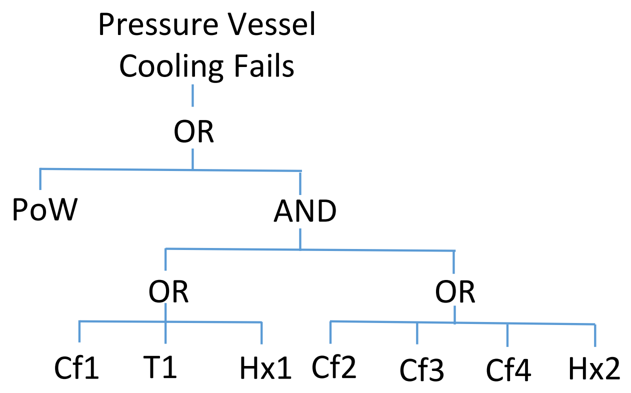
Extract 1





Contraction 2 -- No change

Modularisation (4)



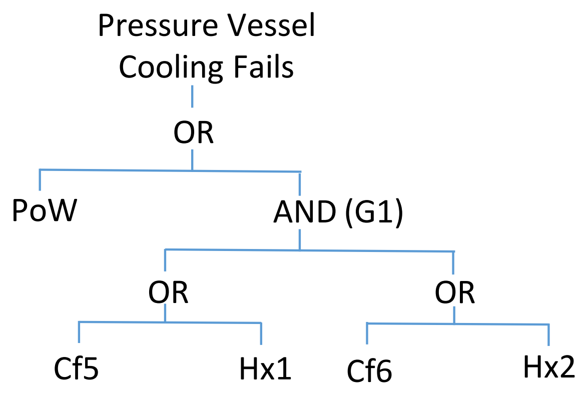
$$Cf_1 = P1.P2$$

$$Cf_2 = S1.S2$$

$$Cf_3 = Comp + R1 + Fan + Motor + R2 + T2 + V1$$

$$Cf_4 = P3S + P3R$$

Factorise 2

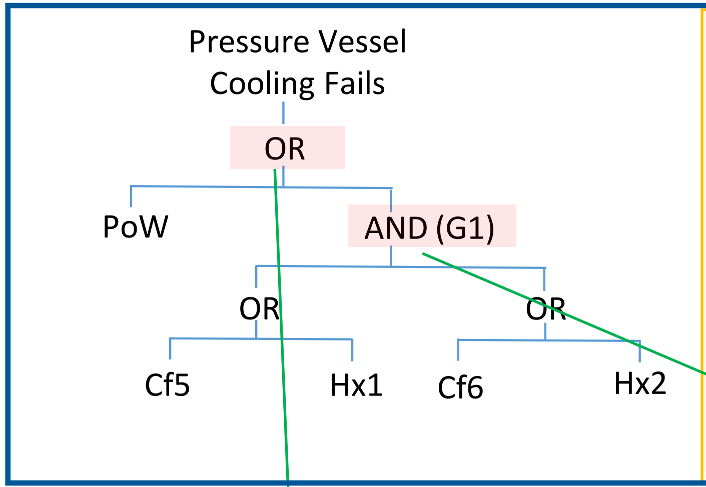


$$Cf_5 = Cf_1 + T1$$

$$Cf_6 = Cf_2 + Cf_3 + Cf_4$$

Simplest possible Reduction representation

Modularisation (5) - Rauzy & Dutuit



$$Cf_1 = P1.P2$$

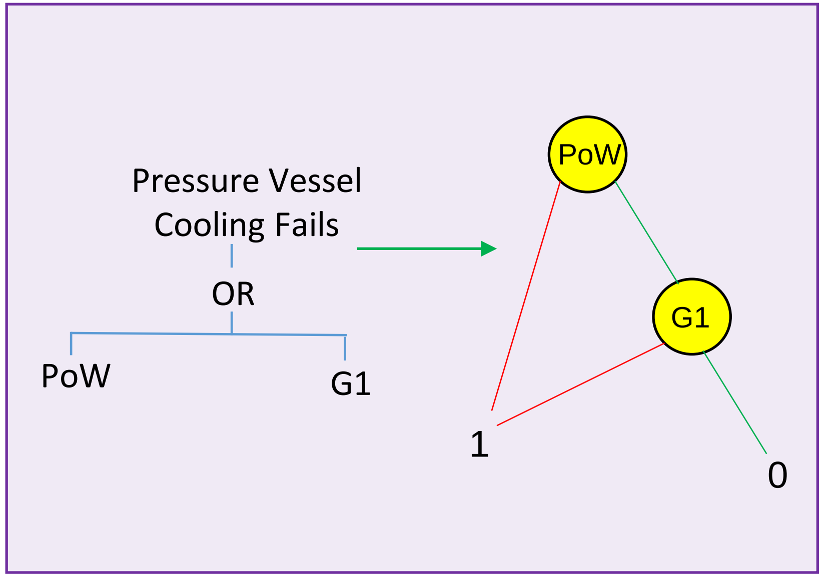
$$Cf_2 = S1.S2$$

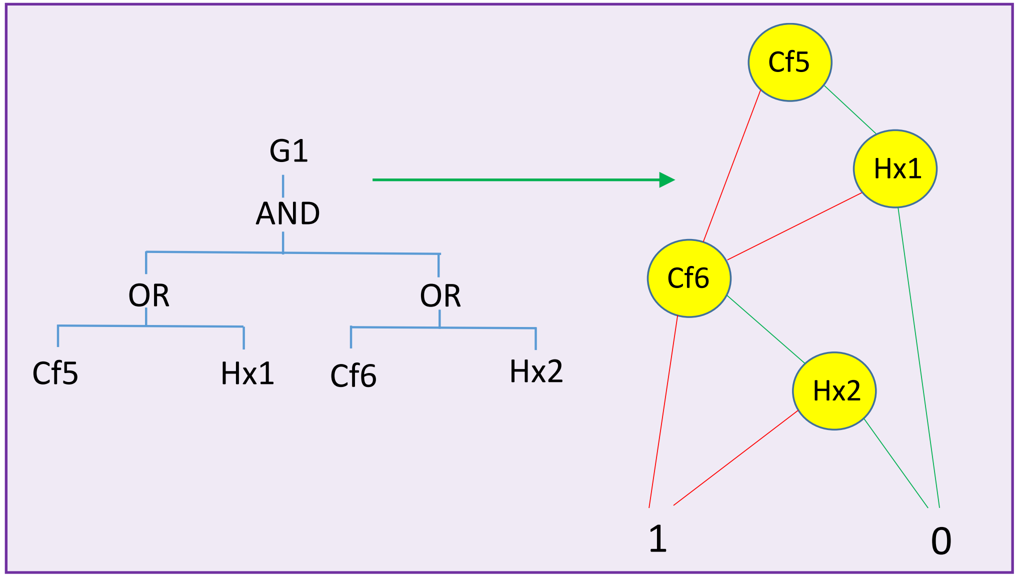
$$Cf_3 = Comp + R1 + Fan + Motor + R2 + T2 + V1$$

$$Cf_4 = P3S + P3R$$

$$Cf_5 = Cf_1 + T1$$

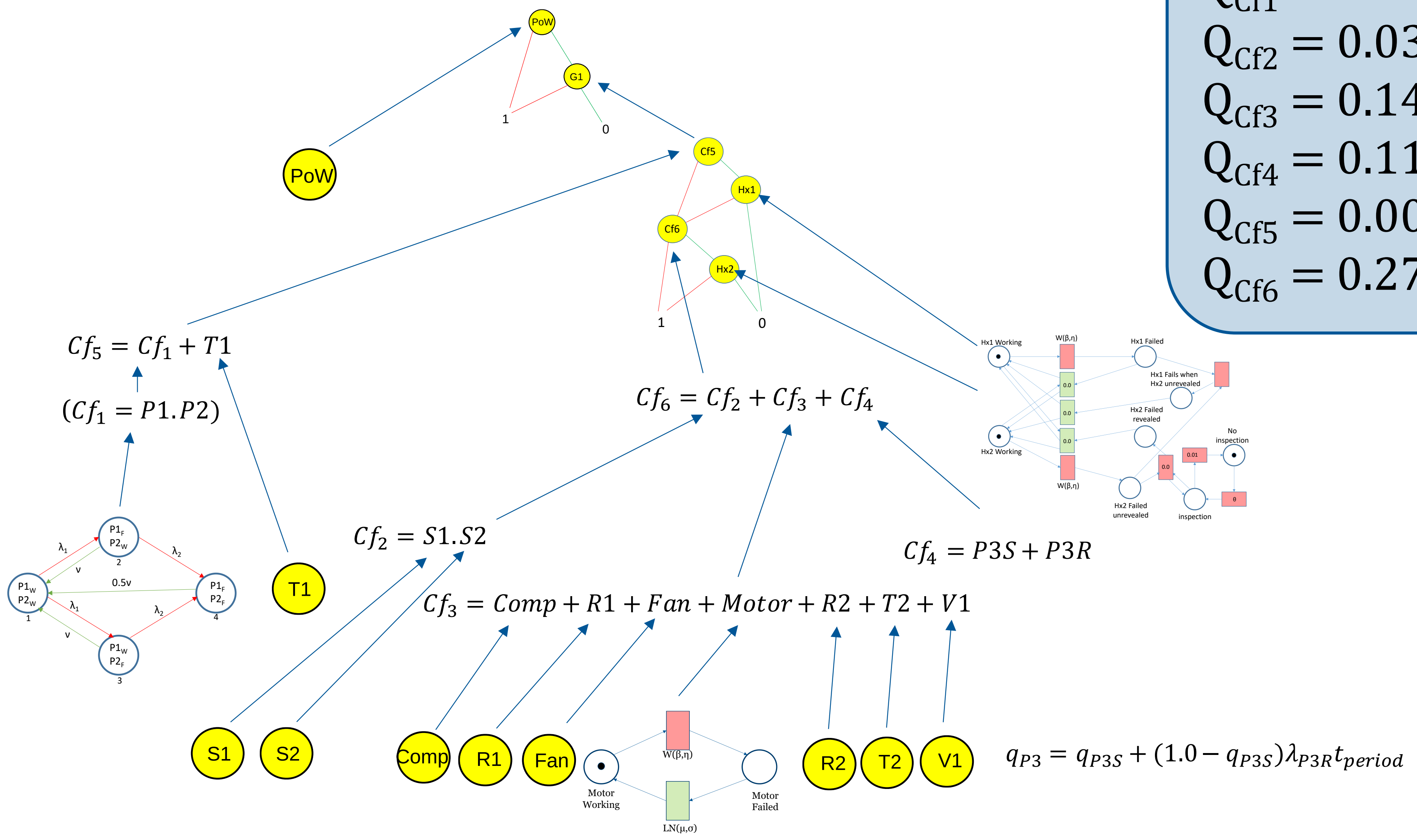
$$Cf_6 = Cf_2 + Cf_3 + Cf_4$$



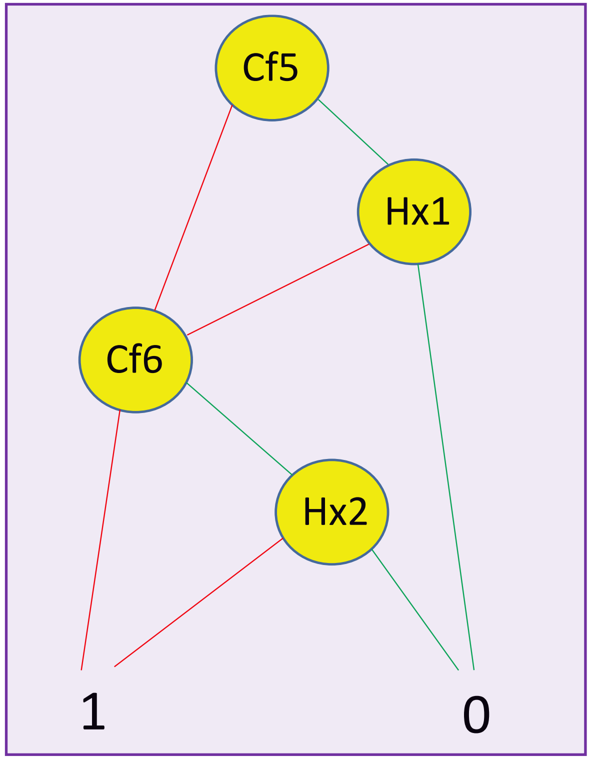


Integration

$$Q_{Cf1} = 0.00170988$$
$$Q_{Cf2} = 0.034225$$
$$Q_{Cf3} = 0.144687275700137 5$$
$$Q_{Cf4} = 0.1184$$
$$Q_{Cf5} = 0.0019494121410861265$$
$$Q_{Cf6} = 0.271763447812487 2$$



G1 Quantification



j	$path_j$	$Ipath_j$	$Dpath_j^1$
1	$Cf5_1, Cf6_1$	$Cf5_1, Cf6_1$	
2	$Cf5_1, Cf6_0, Hx2_1$	$Cf5_1, Cf6_0$	$Hx2_1$
3	$Cf5_0, Hx1_1, Cf6_1$	$Cf5_0, Cf6_1$	$Hx1_1$
4	$Cf5_0, Hx1_1, Cf6_0, Hx2_1$	$Cf5_0, Cf6_0$	$Hx1_1, Hx2_1$

$$Q_{G1} = \sum_{j=1}^{npath} \left[P(Ipath_j). \prod_{k=1}^{ndep} P(Dpath_j^k) \right]$$

$$Q_{path1} = P(Cf5_1).\,P(Cf6_1) = 0.000529778965$$

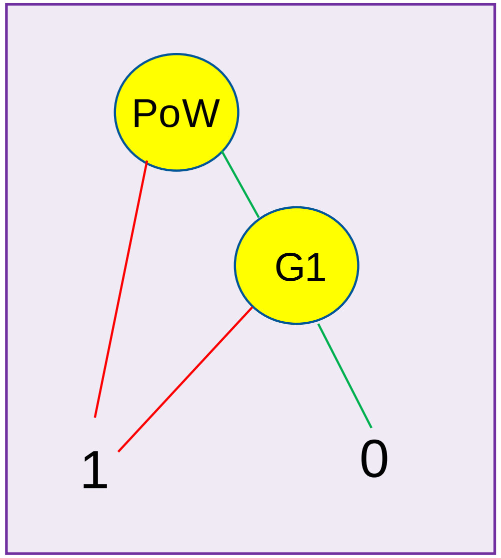
$$Q_{path2} = P(Cf5_1).\,(1 - P(Cf6_1)).\,P(Hx2_1) = 1.920777884 \times 10^{-6}$$

$$Q_{path3} = (1 - P(Cf5_1)).\,P(Cf6_1).\,P(Hx1_1) = 0.0$$

$$Q_{path4} = (1 - P(Cf5_1)).\,(1 - P(Cf6_1)).\,P(Hx1_1, Hx2_1) = 0.0$$

$$Q_{G1} = 0.00054898674$$

Top Event BDD Quantification



$$
\begin{aligned}
Q_{Cf1} &= 0.00170988\\
Q_{Cf2} &= 0.034225\\
Q_{Cf3} &= 0.144687275700137 5\\
Q_{Cf4} &= 0.1184\\
Q_{Cf5} &= 0.0019494121410861265\\
Q_{Cf6} &= 0.2717634478124872\\
Q_{G1} &= 0.0005489867435093285
\end{aligned}
$$

$$Q_{path1} = P(PoW) = 0.000999$$

$$Q_{path2} = (1.0 - P(PoW))\, P(G1) = 0.0005484383$$

$$Q_{SYS} = 0.0015474393042051 23$$

University of Nottingham
UK | CHINA | MALAYSIA

Summary

- The Dynamic and Dependent Tree Theory (D^2T^2) approach has been presented
- The framework removes the need to assume:
  - Basics events are independent
  - Component failure times and repair times are governed by the exponential distribution
  - Simplistic maintenance processes
- D^2T^2 has been formulated to produce efficiency in the quantification performed

University of Nottingham
UK | CHINA | MALAYSIA