Finite Degradation Structures

Future Tool for RAMS Analysis

Liu Yang[1] & Prof. Antoine B. Rauzy[1,2]

(1) Department of Mechanical and Industrial Engineering
Norwegian University of Science and Technology
Trondheim, Norway

(2) Chair Blériot-Fabre
CentraleSupélec / SAFRAN
Paris, France

NTNU

Agenda

Background

Finite Degradation Structures

- Why multi-states?
- Definition
- State-space

Finite Degradation Calculator

Use Cases

Future Tool for RAMS Analysis

Background

Models for RAMS analysis

Dynamic models

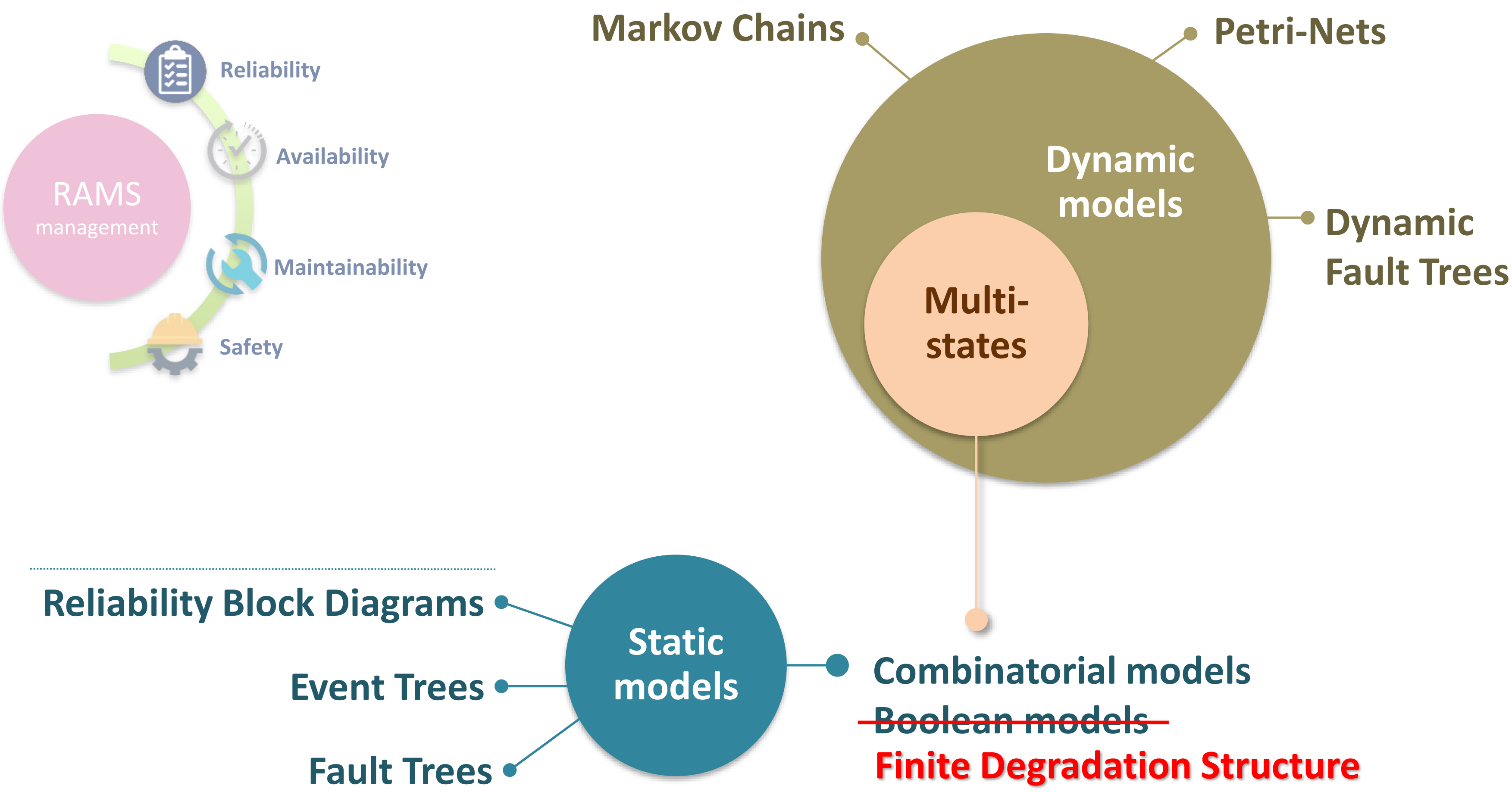
- **Markov Chains**
  - multi-states
  - hidden
- **Petri-Nets**
  - stochastic
  - colored
  - single
- **Dynamic Fault Trees**

Static models

- **Reliability Block Diagrams**
- **Event Trees**
- **Fault Trees**

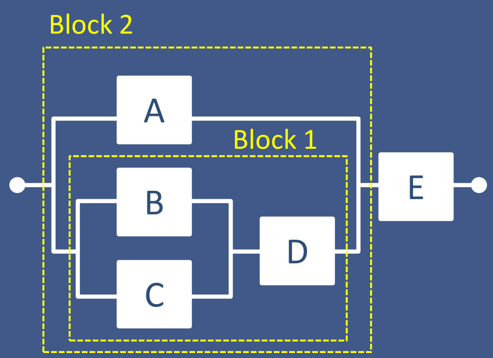
Background

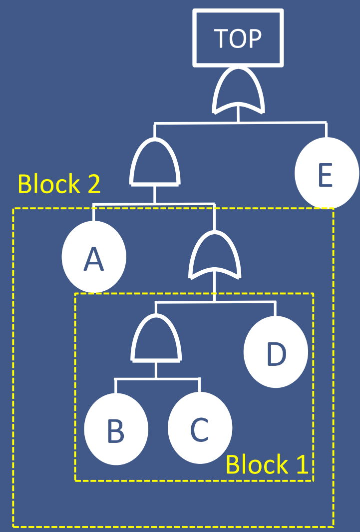
Where things start…

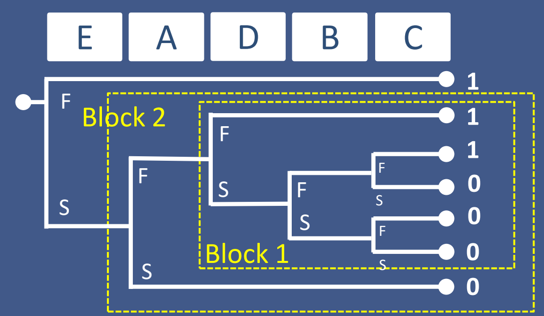


Combinatorial (static) Models

which can be composed bottom-up by sub-blocks







Reliability Block Diagram

Connections:

- Parallel
- Series

Fault Trees

Connections:

- And
- Or

Event Trees

Branches:

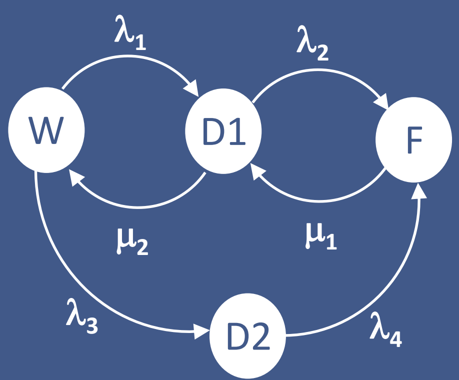
- Success
- Failure

Combinatorial and Boolean models

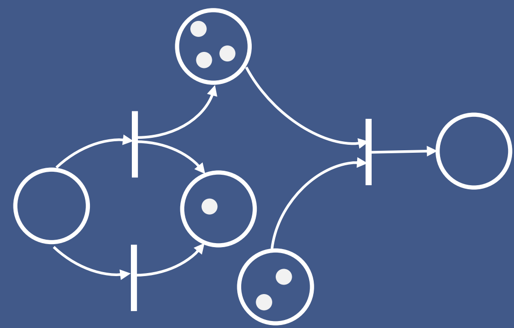
- Component states : working / failed

Multi-states (dynamic) Models

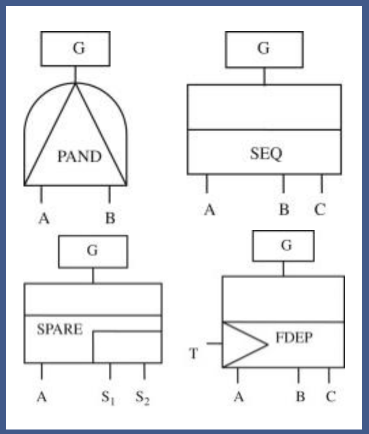
Not combinatorial



Markov Chains



Petri Nets



Dynamic Fault Trees

Multi-states models, but not combinatorial

RAMS Models

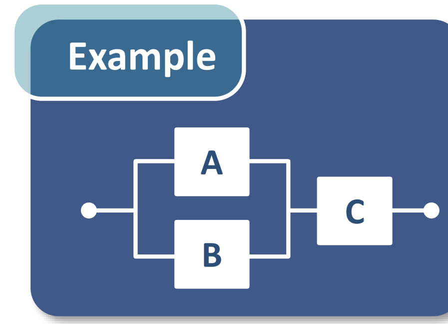
Summary

	Type	Theoretical Basis	
Static / Boolean models	Reliability Block Diagram	Boolean algebra	
	Fault Trees		
	Event Trees	(If-then-else)	
Dynamic / multi-states models	Markov Chain	Finite-state automaton	
	Petri Nets		
	Dynamic Fault Trees (Boolean)	(Priority of events) (if-then-else)	

RAMS Models

Summary

	Type	Theoretical Basis
Static / Boolean models	Reliability Block Diagram	Boolean algebra
	Fault Trees	
	Event Trees	(If-then-else)
Dynamic / multi-states models	Markov Chain	Finite-state automaton
	Petri Nets	
	Dynamic Fault Trees (Boolean)	(Priority of events) (if-then-else)



Logic relation
+
Multi-states components
+
States transition
= ?

Solution

Finite Degradation Calculator

Finite Degradation Calculator

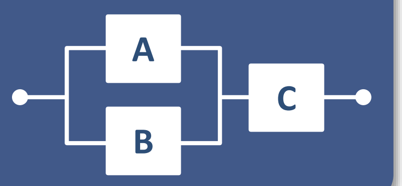
1 **Calculate the reliability function R(t) of multi-states system:**

- Allow having **multi-states components**
- Allow **parallel/series relationship** between components
- Allow **dependencies**

2 **RAMS analysis**

- Generate **minimal cut sets**
- Identify **critical states**
- Calculate **importance measures**
- …

Example



Logic relation
+
Multi-states components
+
States transition
= ?

Agenda

Background

Finite Degradation Structures

- Why multi-states?
- Definition
- State-space

Finite Degradation Calculator

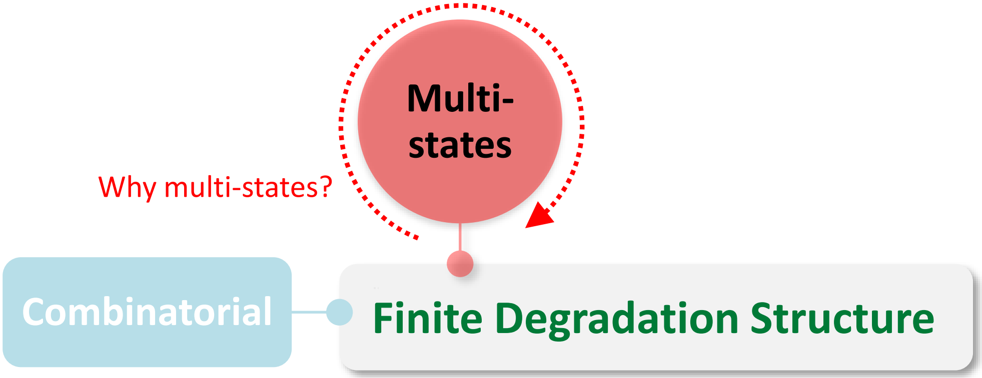
Use Cases

Future Tool for RAMS Analysis

NTNU

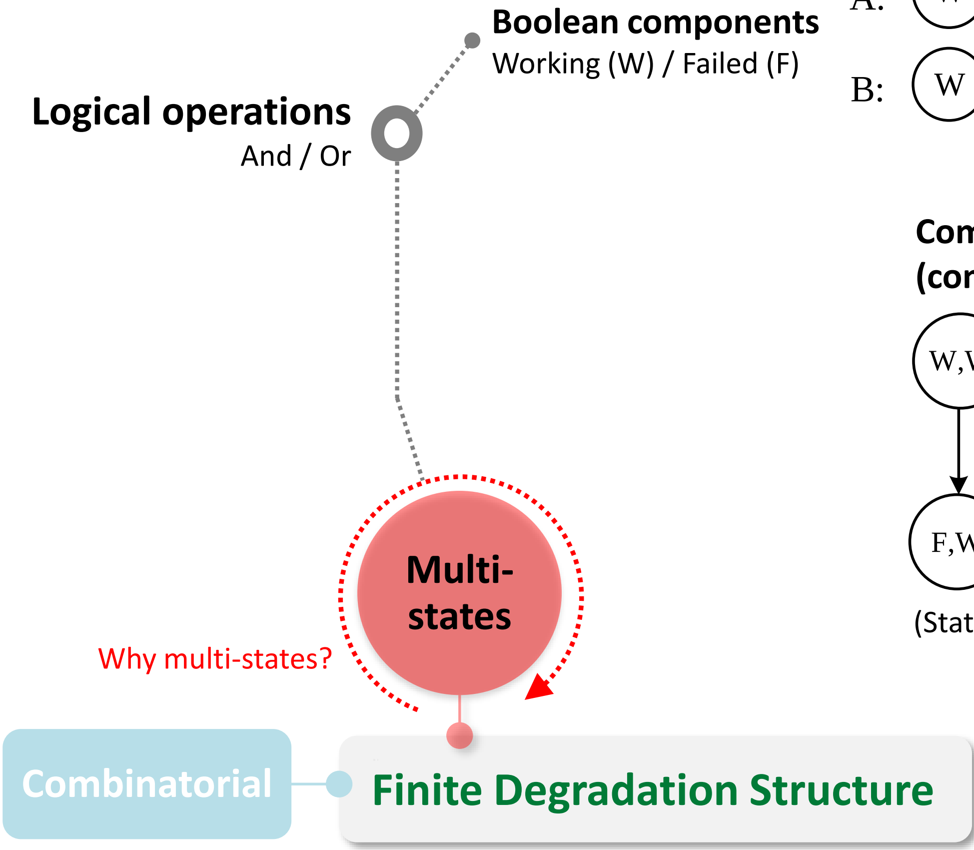
Finite Degradation Structure

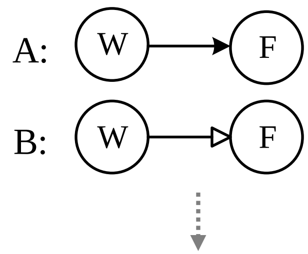
Why having multi-states?

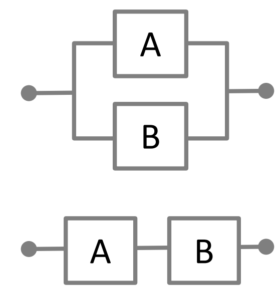


Finite Degradation Structure

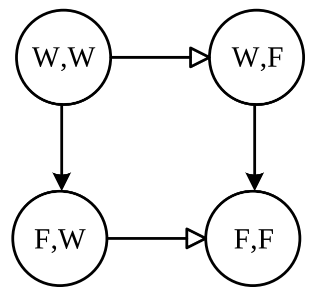
Why having multi-states?







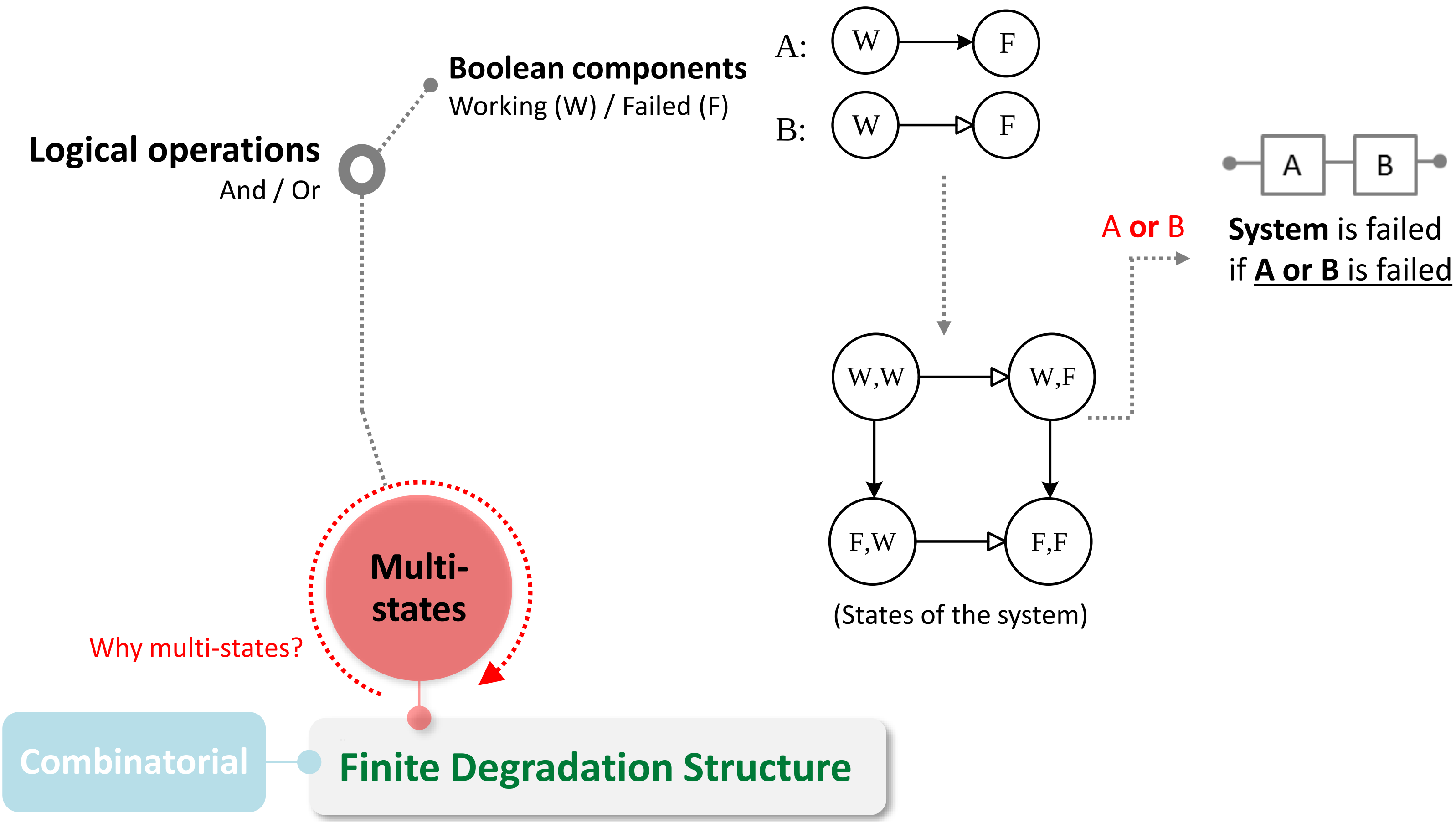
Compose A, B into a system (combinatorial)



(States of the system)

Finite Degradation Structure

Why having multi-states?



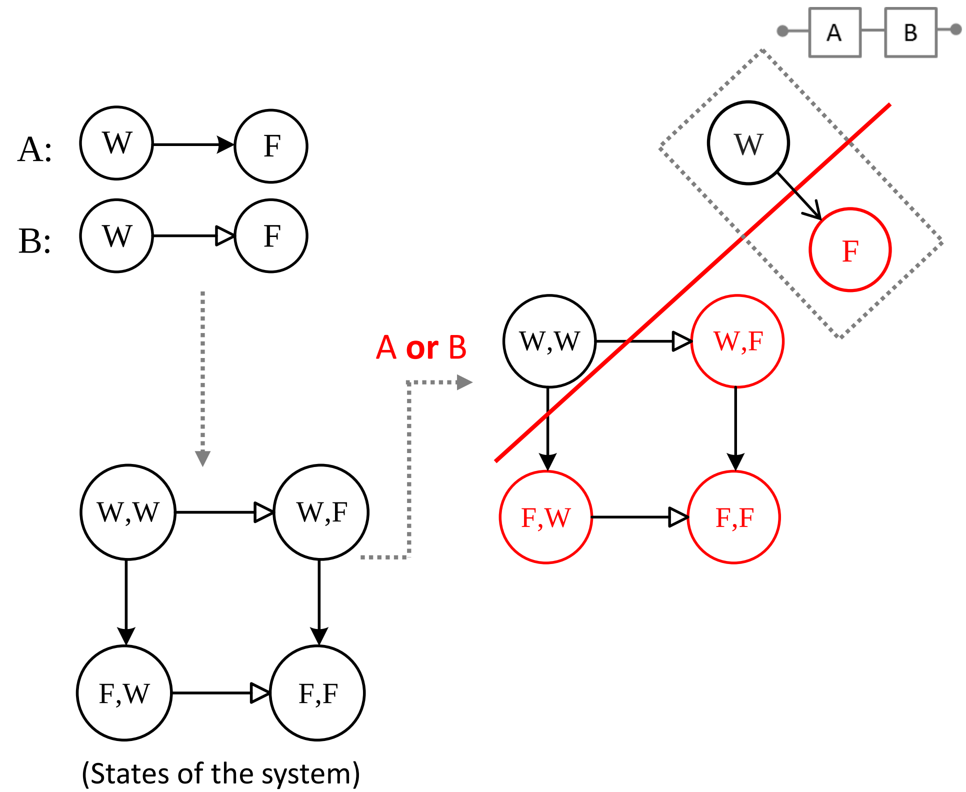
NTNU

Finite Degradation Structure

Why having multi-states?

Logical operations
And / Or

Boolean components
Working (W) / Failed (F)



Why multi-states?

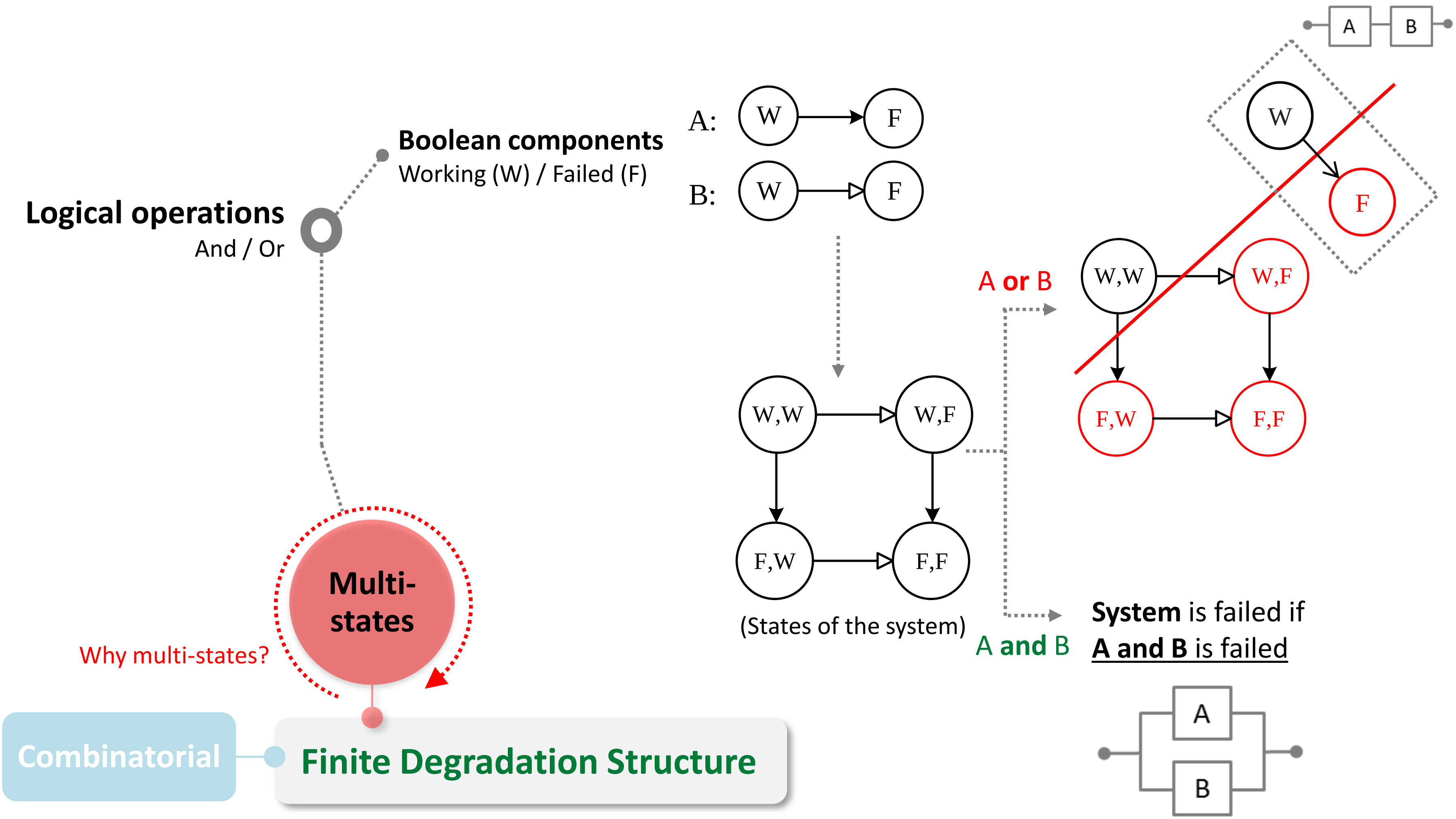
Multi-states

Combinatorial

Finite Degradation Structure

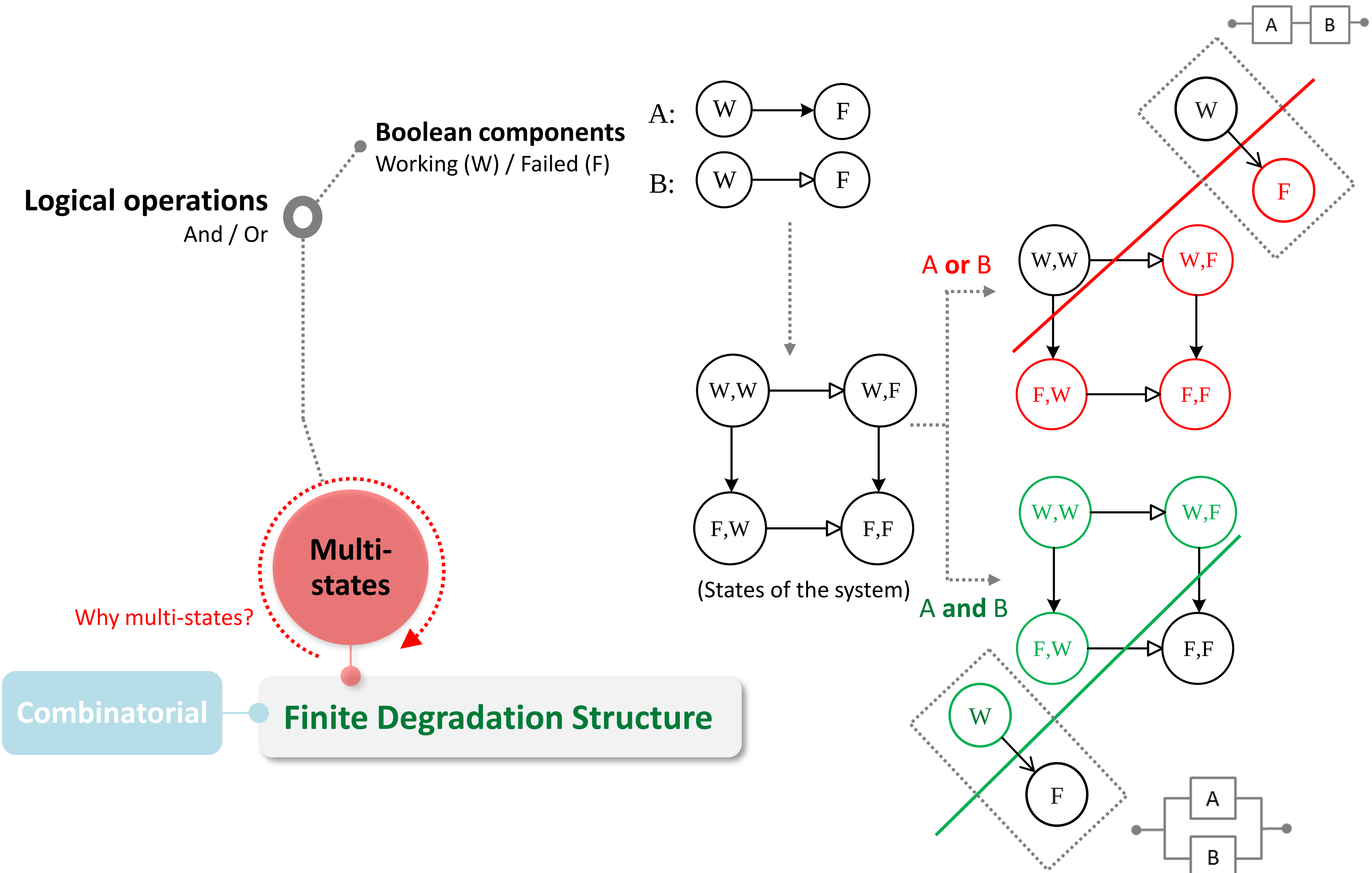
Finite Degradation Structure

Why having multi-states?



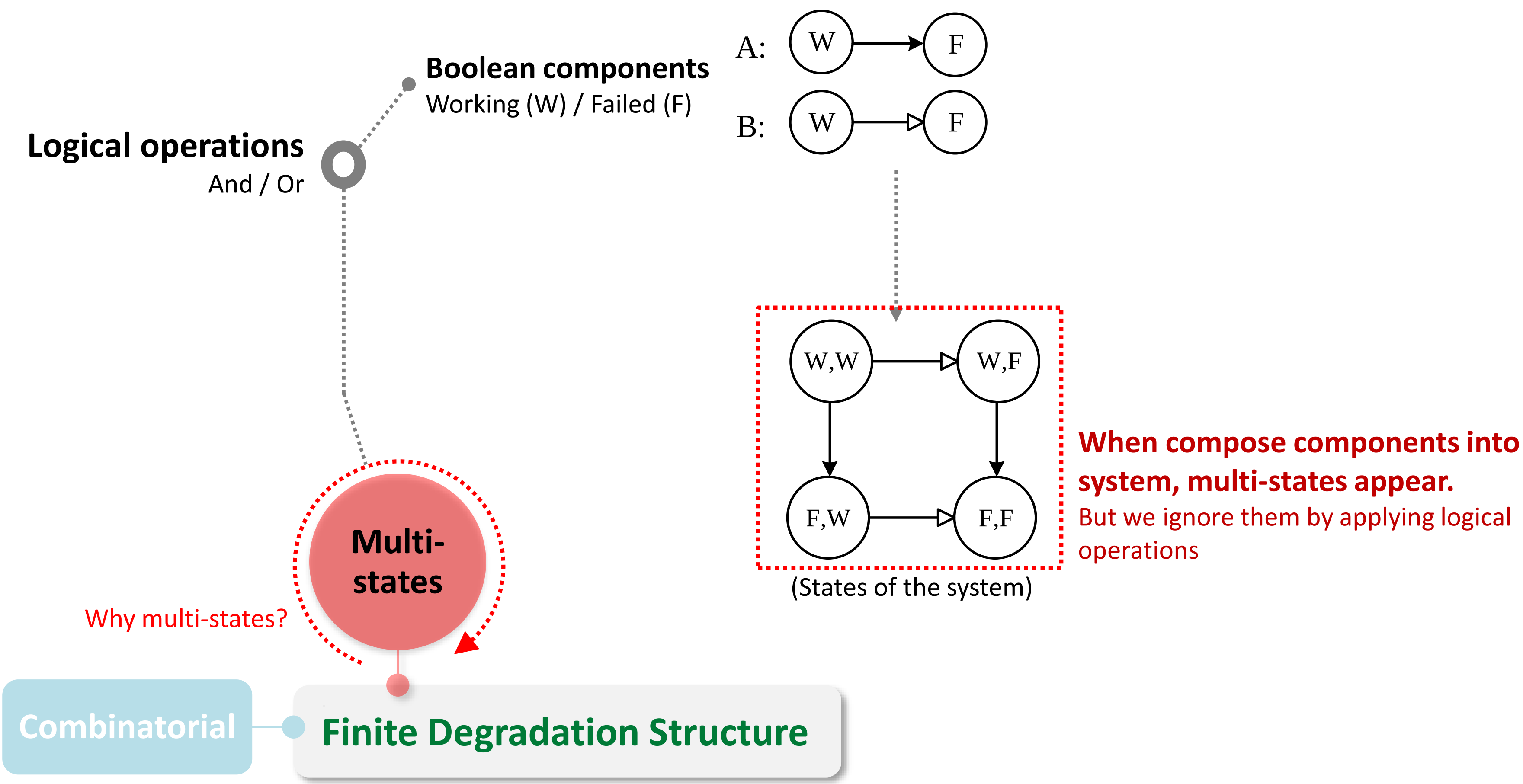
Finite Degradation Structure

Why having multi-states?



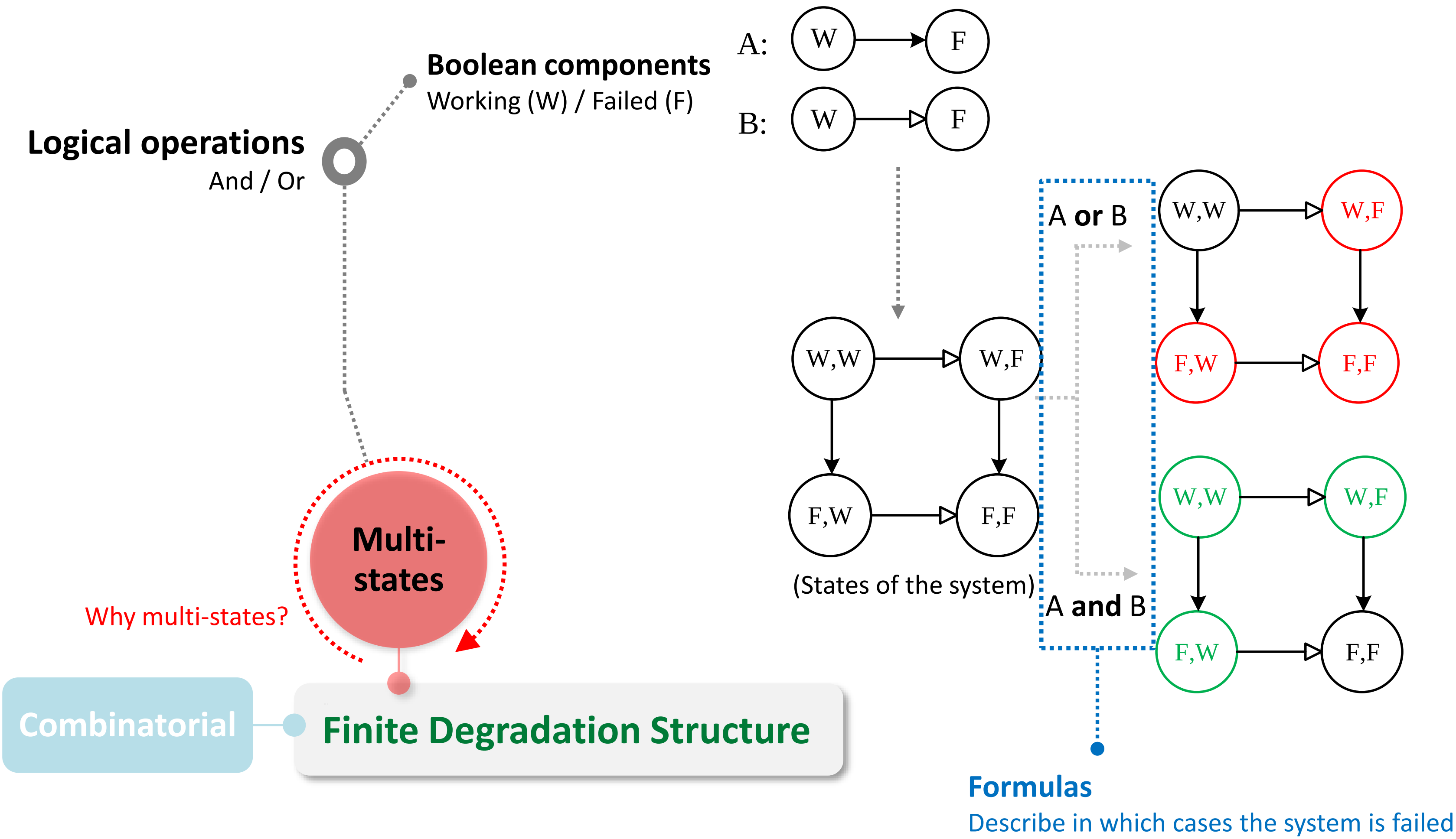
Finite Degradation Structure

Why having multi-states?



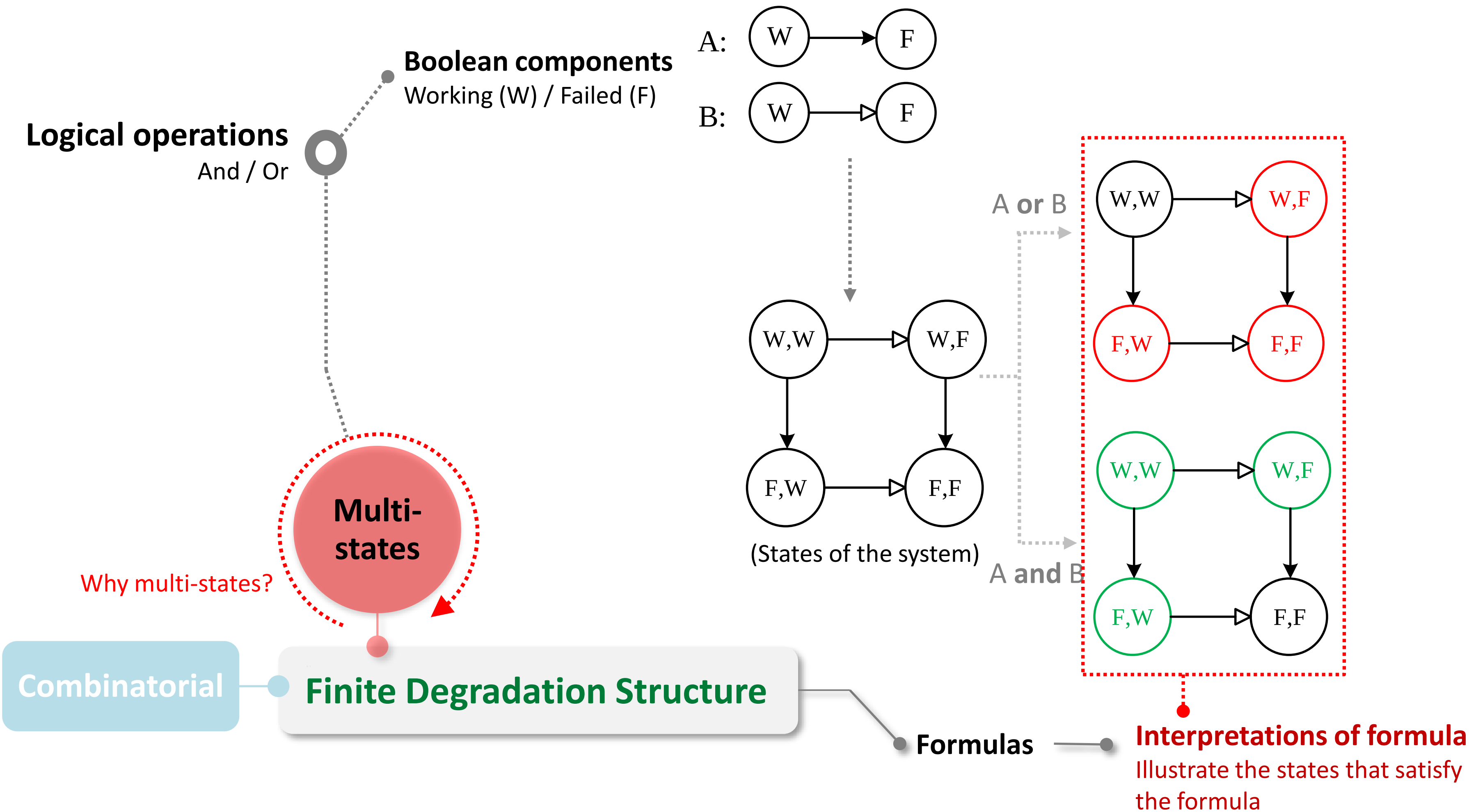
Finite Degradation Structure

Why having multi-states?



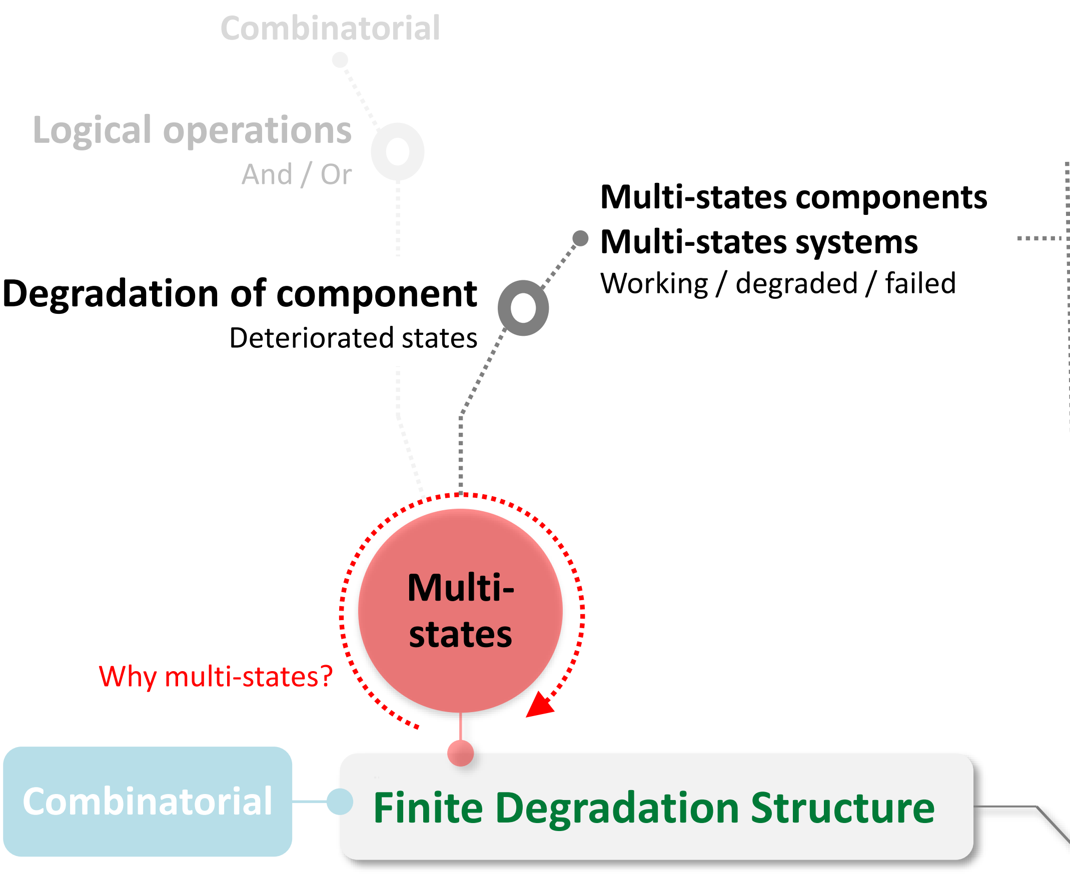
Finite Degradation Structure

Why having multi-states?



Finite Degradation Structure

Why having multi-states?



- Some systems can perform their tasks with various distinctive levels of efficiency usually referred to as ***performance rates***.
- A system that can have a **finite** number of **performance rates** is called a **multi-state system (MSS)**. [1]

[1] Lisnianski, A., Frenkel, I., & Ding, Y. (2010). *Multi-state system reliability analysis and optimization for engineers and industrial managers*. Springer Science & Business Media.

Finite Degradation Structure

Why having multi-states?

Combinatorial

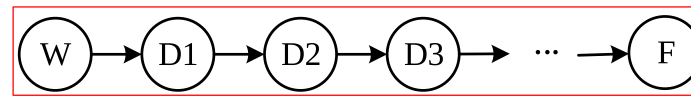
Logical operations
And / Or

Degradation of component
Deteriorated states

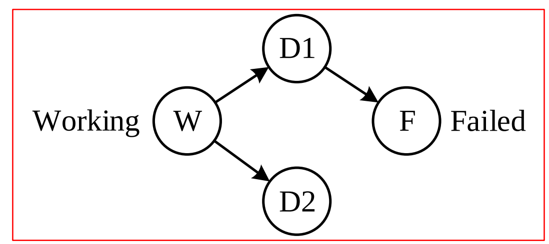
Multi-states components
Multi-states systems
Working / degraded / failed

Performance rate
range from perfect functioning up to complete failure

- Proportional to the number of available units. (k-out of-n).



- vary as a result of their deterioration (fatigue, partial failures) or because of variable ambient conditions.



Multi-states

Why multi-states?

Combinatorial

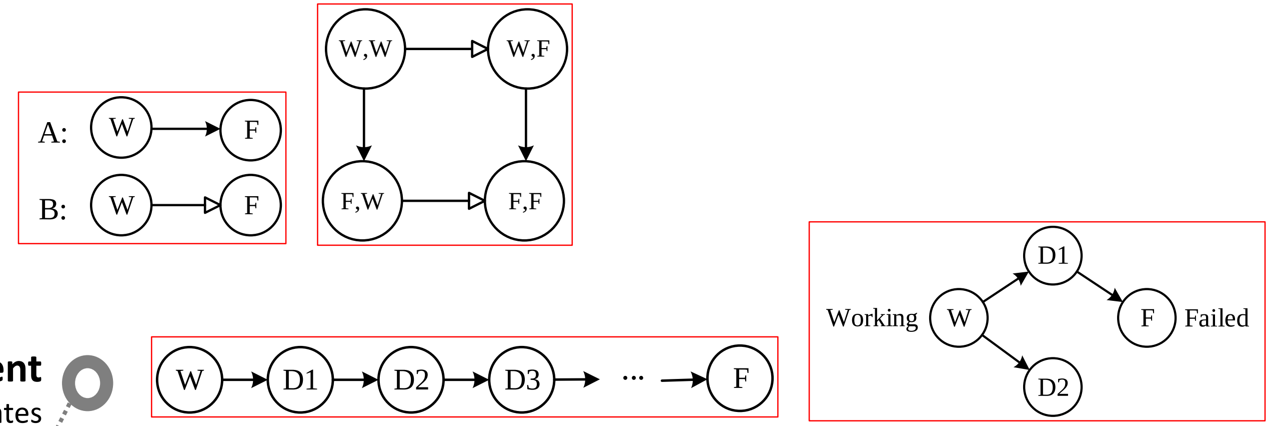
Finite Degradation Structure

Formulas

Interpretations

Finite Degradation Structure

Why having multi-states?



Logical operations
And / Or

Degradation of component
Deteriorated states

Finite degradation structures
Describe the degradation paths of a component or a system

Why multi-states?

Multi-states

Combinatorial

Finite Degradation Structure

Formulas

Interpretations

Agenda

Background

Finite Degradation Structures

- Why multi-states?
- Definition
- State-space

Finite Degradation Calculator

Use Cases

Future Tool for RAMS Analysis

NTNU

Finite Degradation Structure

Definition by graphs

Logical operations

And / Or

Degradation of component

Deteriorated states

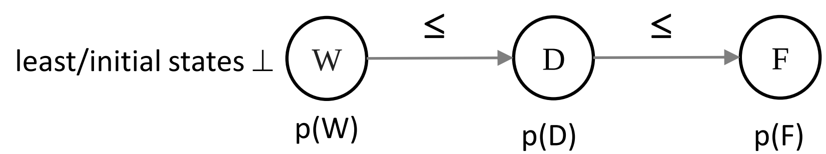
Multi-states

Combinatorial

Finite Degradation Structure

Definition (informal)

A **finite degradation structure** is a **directed acyclic graph**



$$\sum_{s \in \{W,D,F\}} p(s) = 1$$

Formulas

Interpretations

NTNU

Finite Degradation Structure

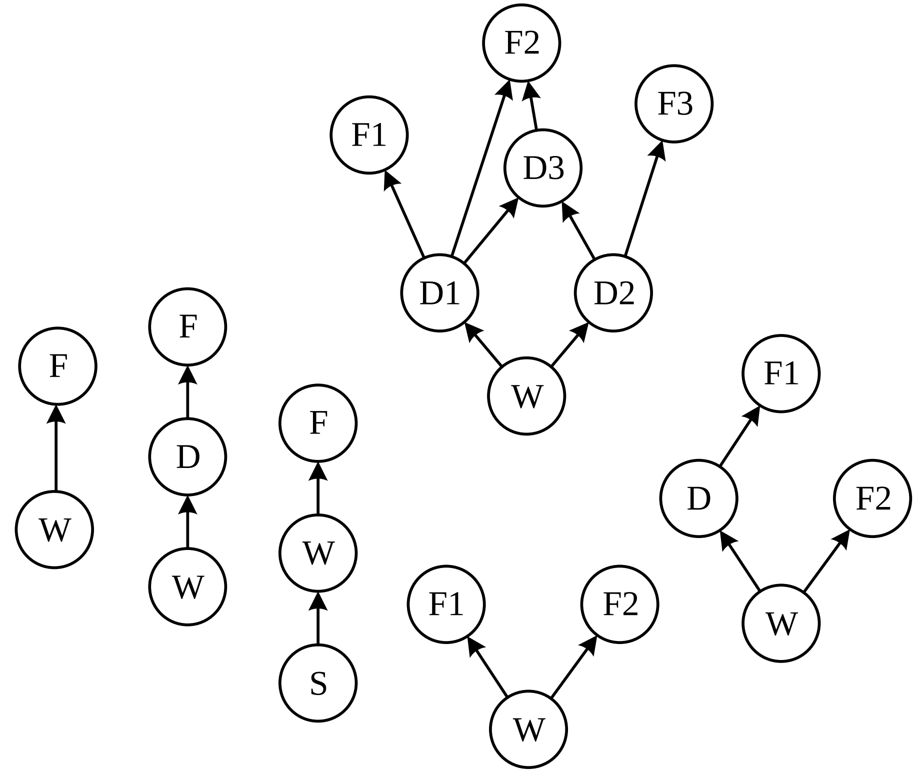
Definition by graphs

Logical operations
And / Or

Degradation of component
Deteriorated states

Multi-states

Examples



Definition (informal)

Combinatorial

Finite Degradation Structure

Formulas

Interpretations

Agenda

Background

Finite Degradation Structures

- Why multi-states?
- Definition
- State-space

Finite Degradation Calculator

Use Cases

Future Tool for RAMS Analysis

Finite Degradation Structure

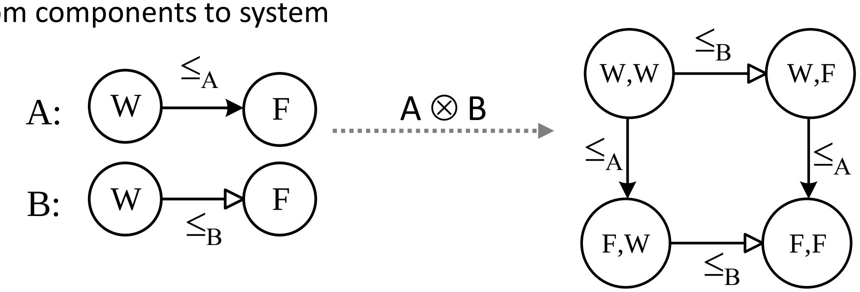
Combinatorial while having multi-states

Logical operations
And / Or

Degradation of component
Deteriorated states

Multi-states

Generate the state-space of the system
From components to system



Combinatorial
Tensor product ⊗

Combinatorial

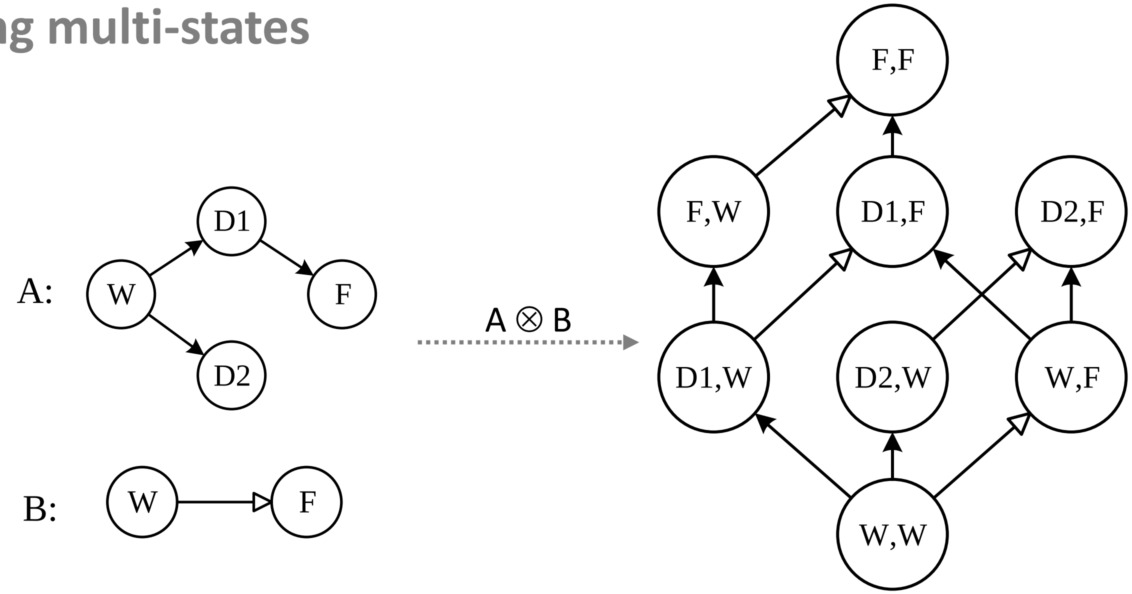
Finite Degradation Structure

Formulas

Interpretations

Finite Degradation Structure

Combinatorial while having multi-states



Logical operations
And / Or

Degradation of component
Deteriorated states

Multi-states

Generate the state-space of the system
From components to system

Combinatorial
Tensor product ⊗

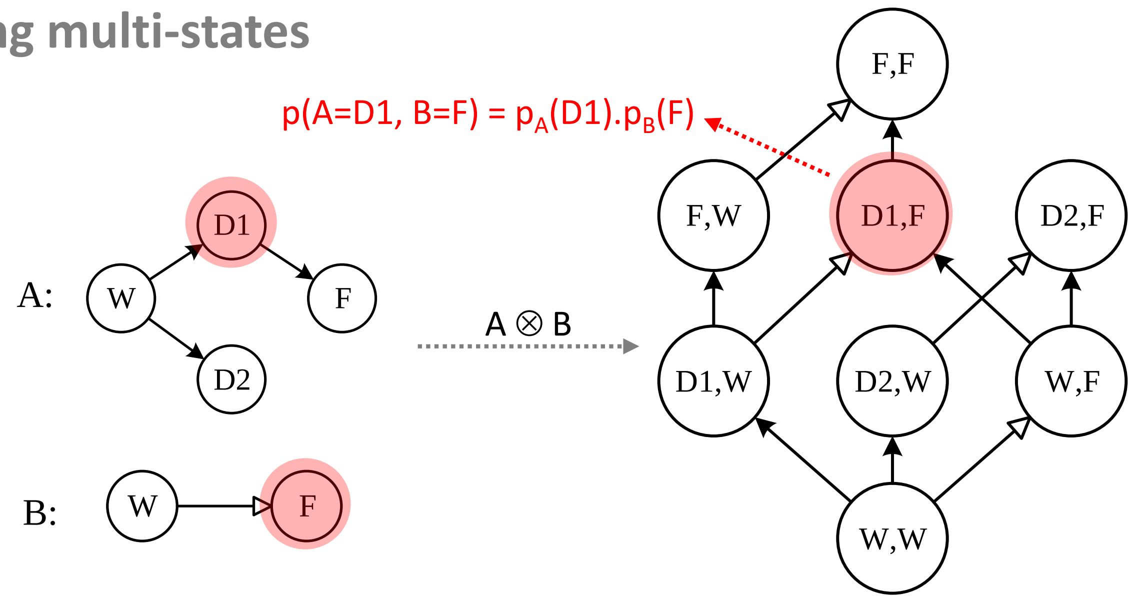
Combinatorial

Finite Degradation Structure

Formulas — **Interpretations**

Finite Degradation Structure

Combinatorial while having multi-states



Logical operations
And / Or

Degradation of component
Deteriorated states

Multi-states

Calculating the probability after A ⊗ B
The probability that the system is in a certain state

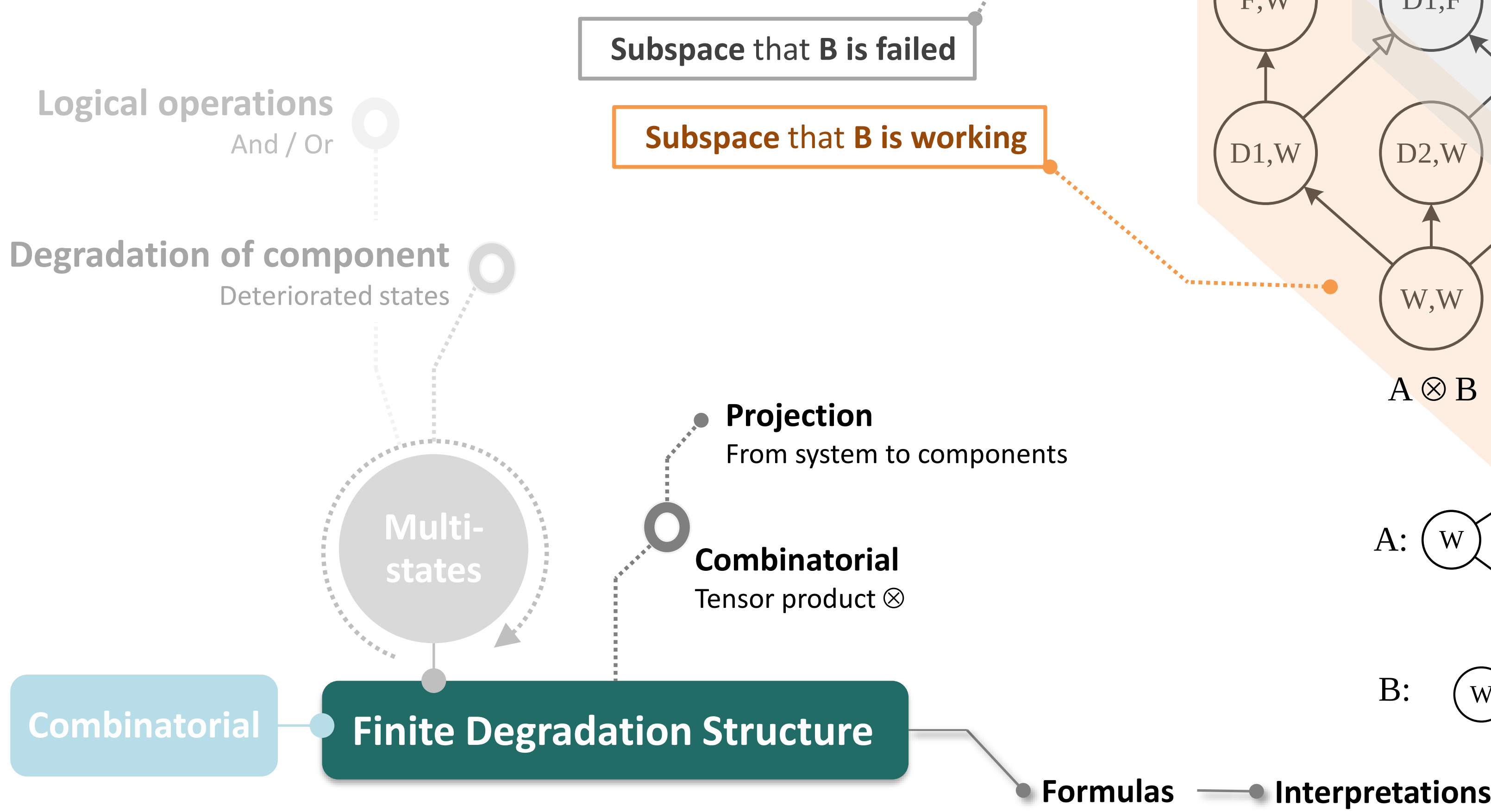
Combinatorial
Tensor product ⊗

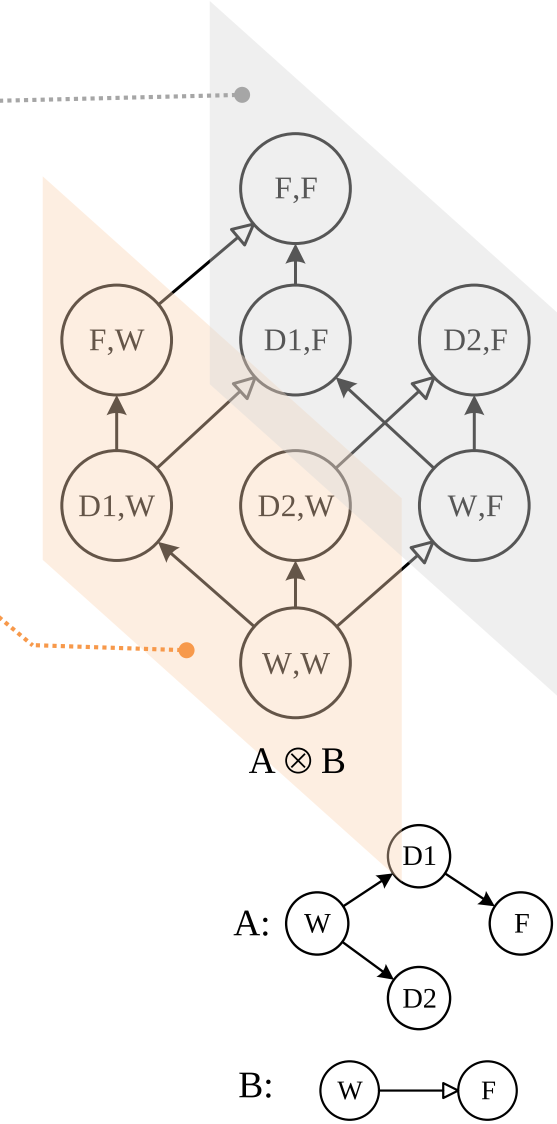
Combinatorial — **Finite Degradation Structure** — **Formulas** — **Interpretations**

NTNU

Finite Degradation Structure

Combinatorial while having multi-states





Finite Degradation Structure

Combinatorial while having multi-states

Logical operations
And / Or

Degradation of component
Deteriorated states

Multi-states

Combinatorial

Finite Degradation Structure

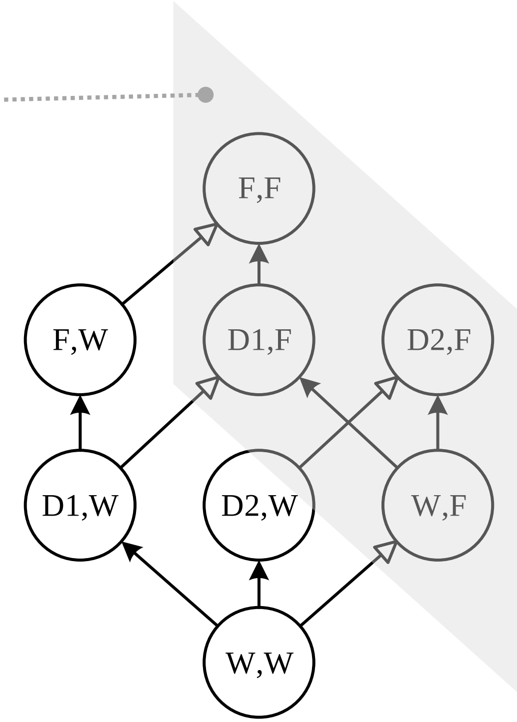
Combinatorial
Tensor product ⊗

Calculating probability after projection
Conditional probability

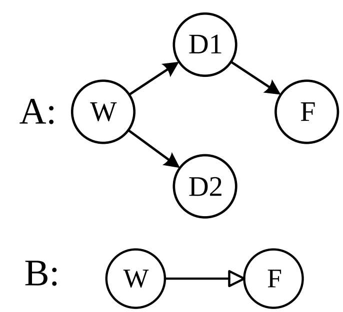
Formulas — **Interpretations**

Subspace that B is failed

$$
\begin{aligned}
p(A=*, B=F) &= p_B(F).[p_A(W)+p_A(D1)+p_A(D2)+p_A(F)] \\
&= p_B(F).1 \\
&= p_B(F)
\end{aligned}
$$



A ⊗ B



Finite Degradation Structure

Combinatorial while having multi-states

Logical operations

And / Or

Degradation of component

Deteriorated states

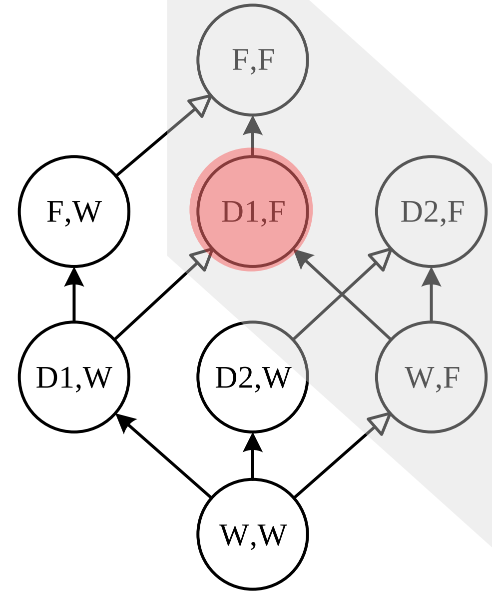
Multi-states

Combinatorial

Finite Degradation Structure

Subspace that B is failed

$p(A=D1,B=F) = p(A=D1|B=F) = p_B(F).p_A(D1)$



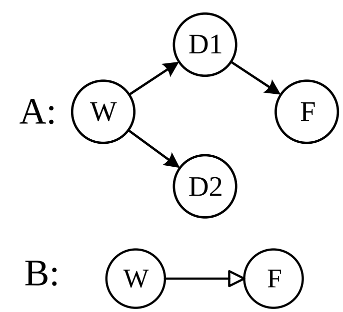
A ⊗ B

Calculating probability after projection

Conditional probability

Combinatorial

Tensor product ⊗



Formulas — **Interpretations**

NTNU

Agenda

Background

Finite Degradation Structures

- Why multi-states?
- Definition
- State-space

Finite Degradation Calculator

Use Cases

Future Tool for RAMS Analysis

New Tool for RAMS analysis

Finite Degradation Calculator

Finite Degradation Calculator

(1) **Define** the finite degradation structure of each component

(2) **Assign** the probability distribution of each state of each component

- Manual calculation
- Bayesian networks
- Markov chains
- …

(3) **Mapping** the set of states from component-level to system-level

Calculate the reliability function R(t) of a system composed by multi-states components

NTNU

Agenda

Background

Finite Degradation Structures

- Why multi-states?
- Definition
- State-space

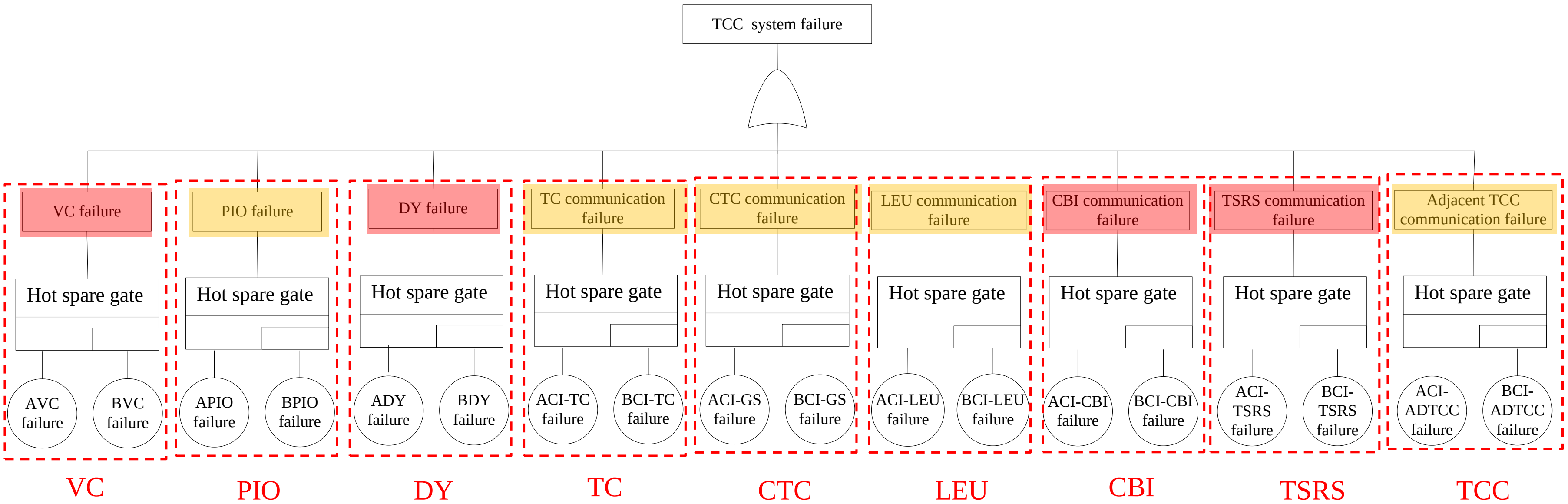
Finite Degradation Calculator

Use Cases

Future Tool for RAMS Analysis

Use Case: TCC System Failure (Railway)

Finite Degradation Calculator



	$\lambda_{A/B}$	$\mu_{A/B}$
VC	1.26E-04	0.002
PIO	2.28E-04	0.002
DY	1.59E-04	0.002
CI-TC	1.20E-04	0.002
CI-GS	3.10E-05	0.002

	$\lambda_{A/B}$	$\mu_{A/B}$
CI-LEU	9.20E-05	0.002
CI-CBI	2.10E-05	0.002
CI-TSRS	2.10E-05	0.002
CI-ADTCC	2.10E-05	0.002

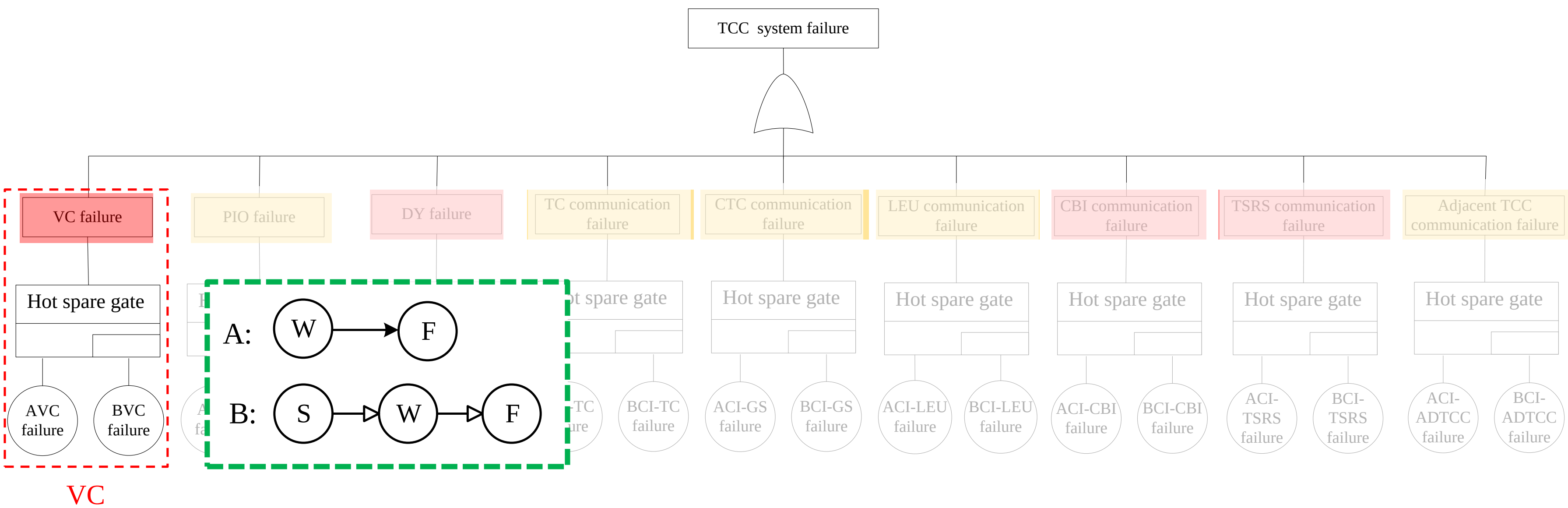
TCC system:

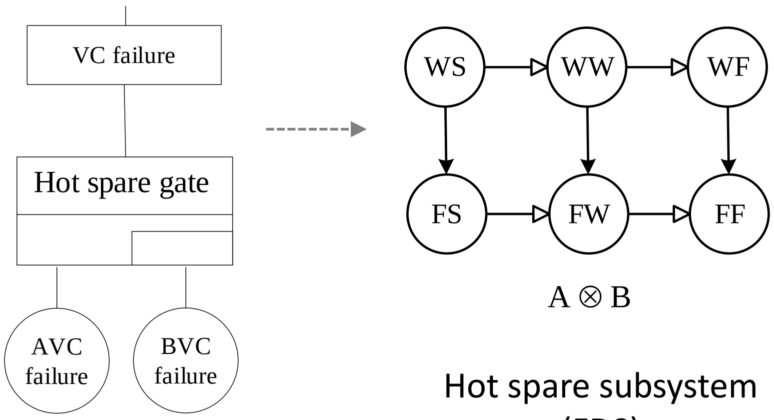
- If at least one of the "red" failures happens, the system is failed.
- If none of the "red" failures happen and at least one of the "yellow" failures happens, the system is degraded.
- Else, the system is working.

Reliability function R(t)?

Use Case: TCC System Failure (Railway)

Finite Degradation Calculator

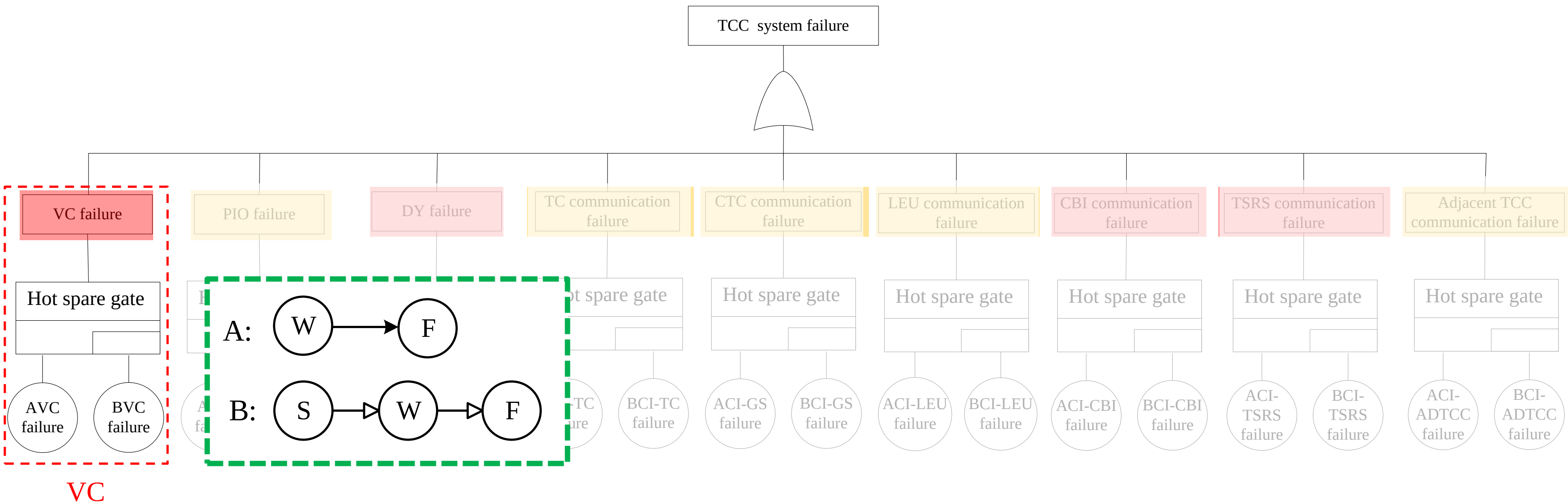


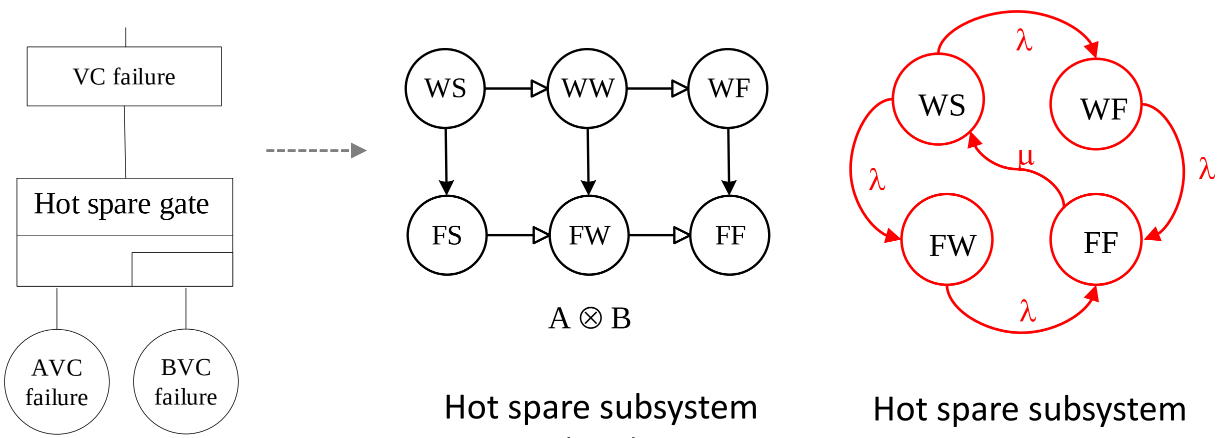


Hot spare subsystem
(FDS)

Use Case: TCC System Failure (Railway)

Finite Degradation Calculator



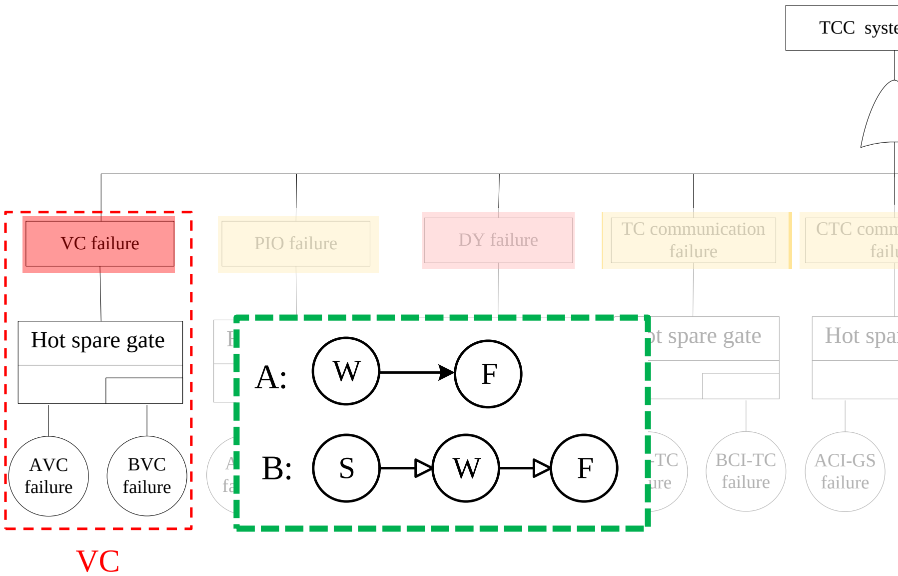


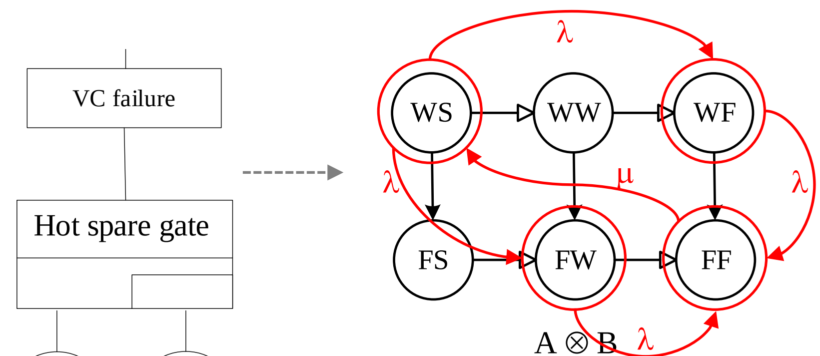
Hot spare subsystem (FDS)

Hot spare subsystem (Markov Chain model)

Use Case: TCC System Failure (Railway)

Finite Degradation Calculator





Hot spare subsystem (FDS)

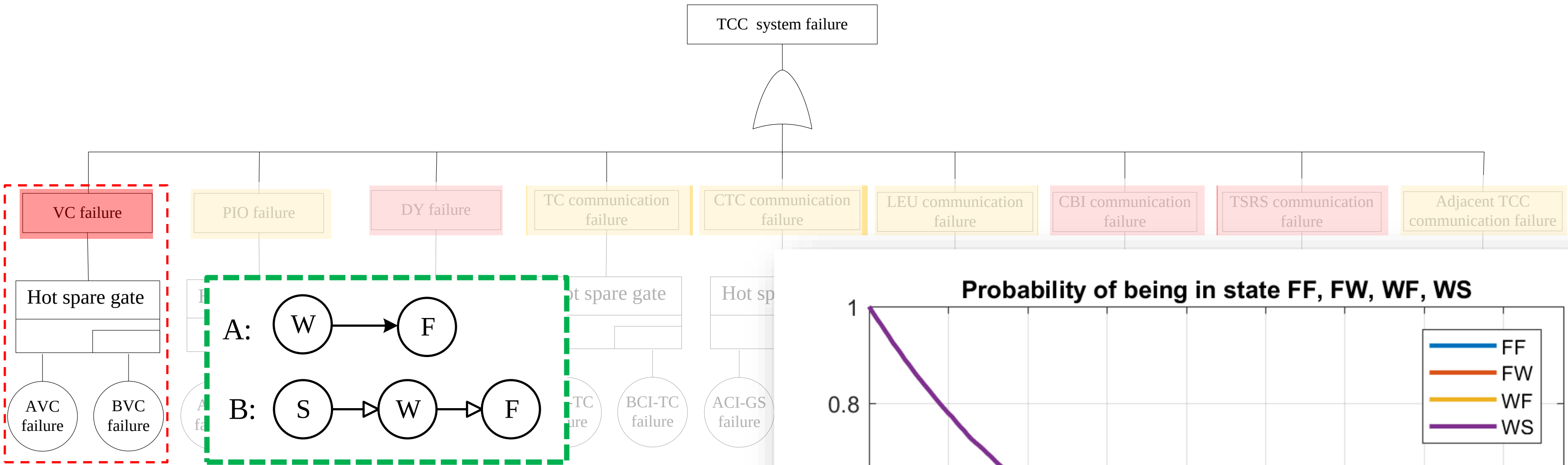
```
domain sysState {WS,WF,FW,FF}

block system
        sysState s (init = WS);
        parameter Real lambda = 1.26E-4;
        parameter Real miu = 0.002;
        event e12 (delay = exponential (lambda));
        event e24 (delay = exponential (lambda));
        event e13 (delay = exponential (lambda));
        event e34 (delay = exponential (lambda));
        event e41 (delay = exponential (miu));
        transition
                e12: s==WS -> s:=WF;
                e24: s==WF -> s:=FF;
                e13: s==WS -> s:=FW;
                e34: s==FW -> s:=FF;
                e41: s==FF -> s:=WS;
        observer Boolean s_WS = s==WS;
        observer Boolean s_WF = s==WF;
        observer Boolean s_FW = s==FW;
        observer Boolean s_FF = s==FF;
end
```

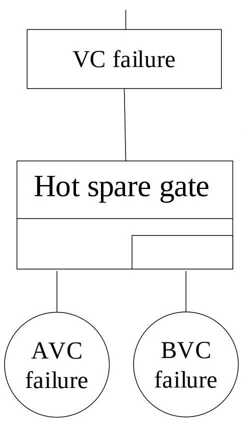
AltaRica Code

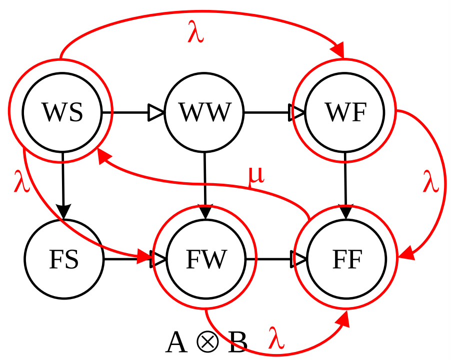
Use Case: TCC System Failure (Railway)

Finite Degradation Calculator

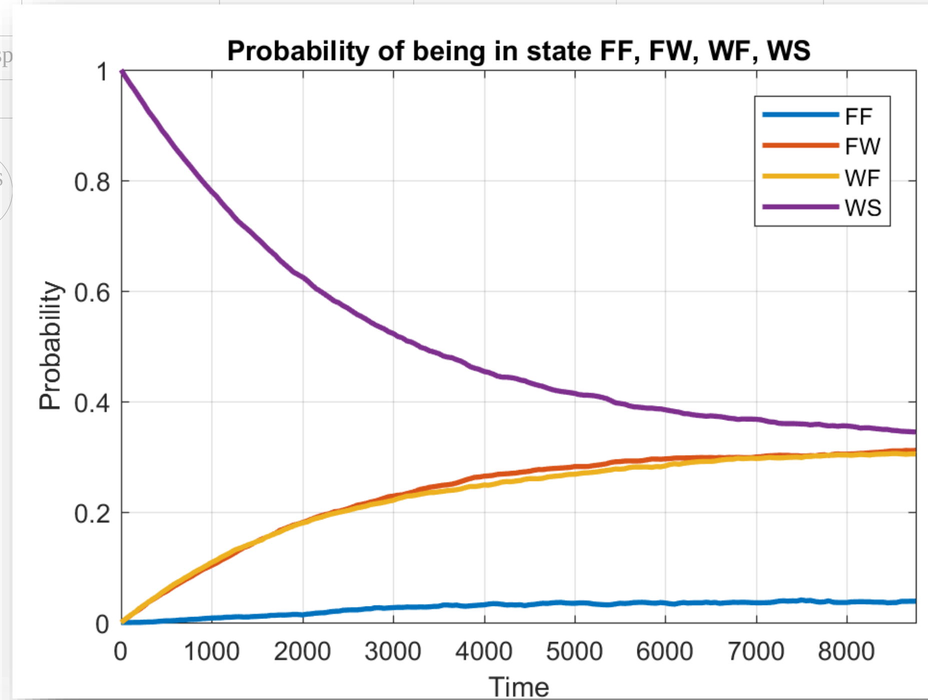


VC





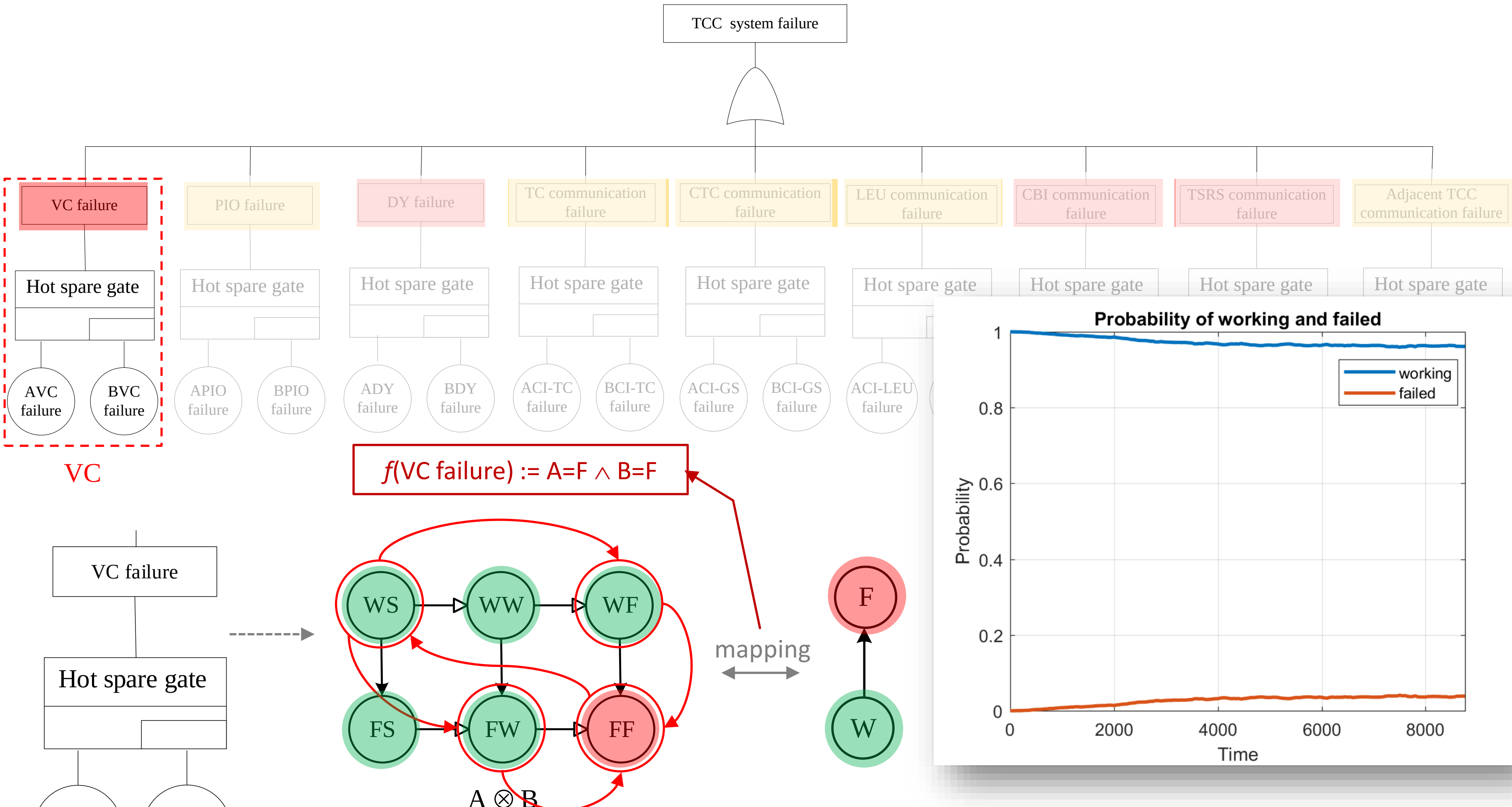
Hot spare subsystem (FDS)



AltaRica Results

Use Case: TCC System Failure (Railway)

Finite Degradation Calculator



Hot spare subsystem (FDS)

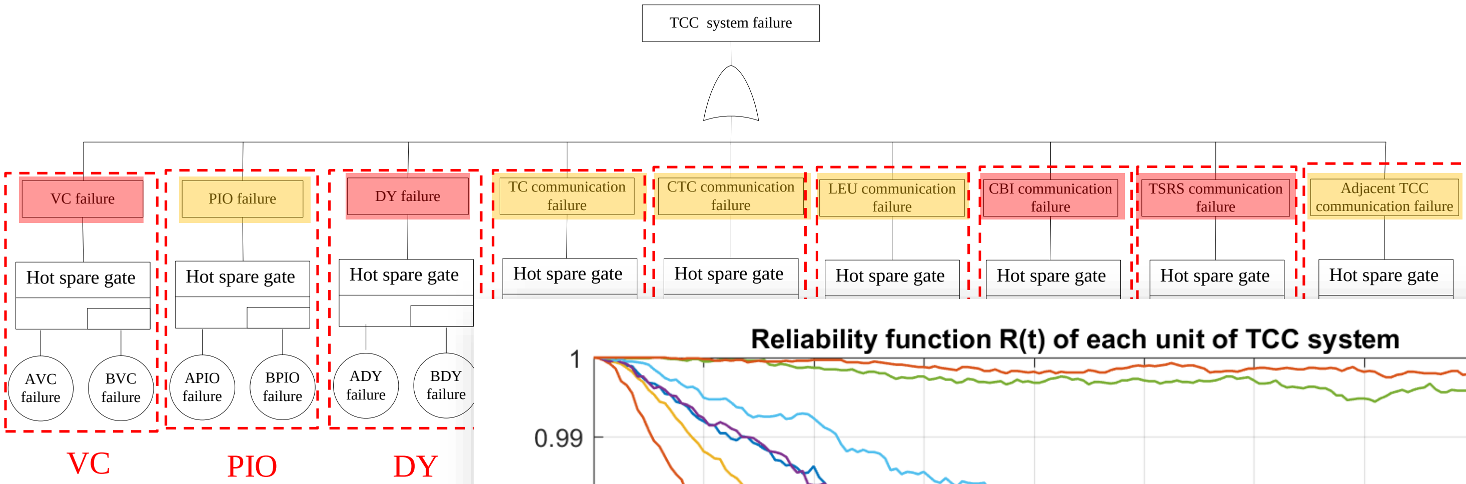
$R_{VC}(t) = Pr(WS, t) + Pr(FS, t) + Pr(WW, t) + Pr(WF, t) + Pr(FW, t)$

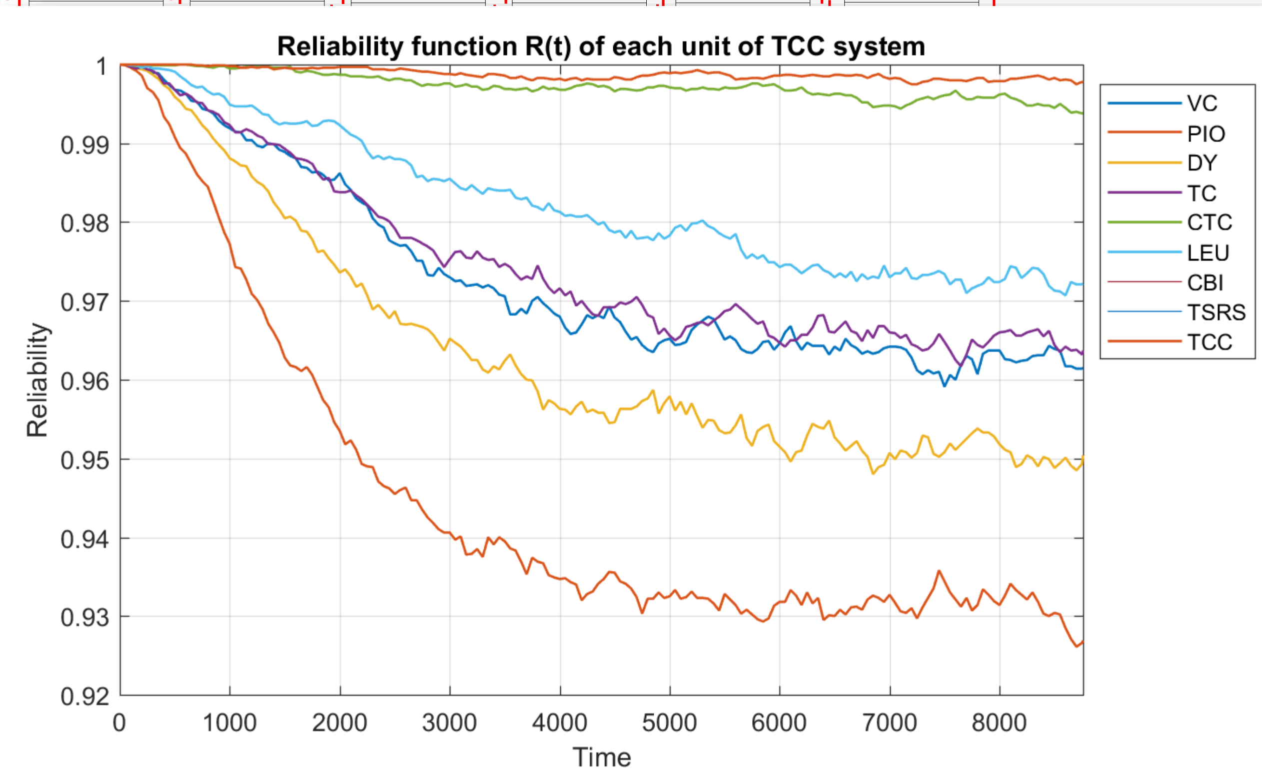
$= Pr(WS, t) + Pr(WF, t) + Pr(FW, t)$

NTNU

Use Case: TCC System Failure (Railway)

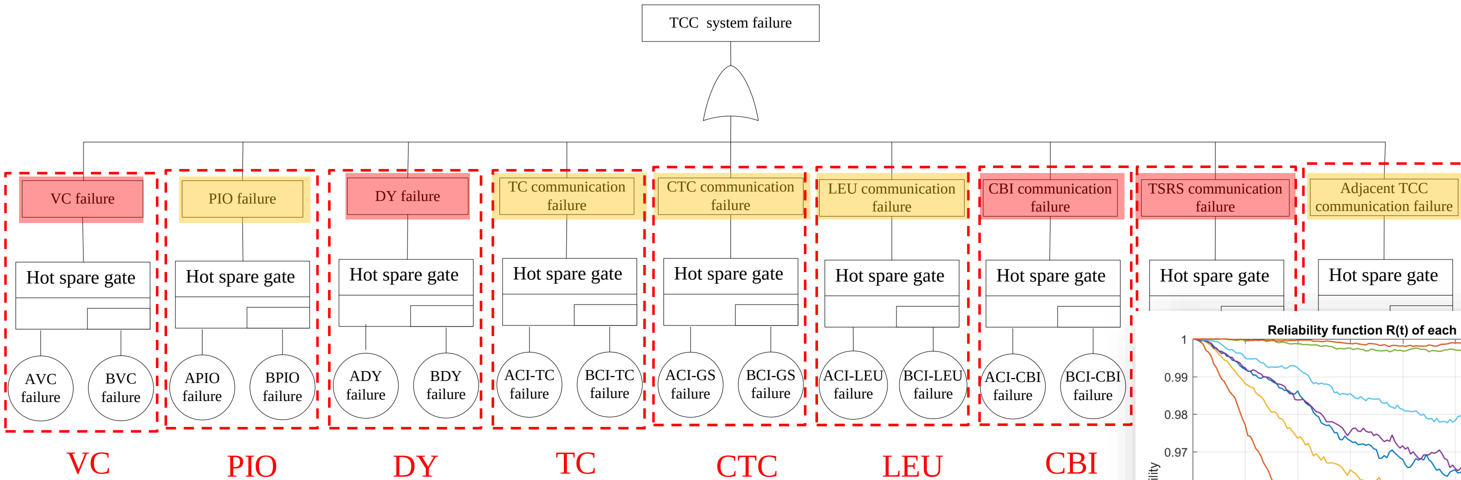
Finite Degradation Calculator

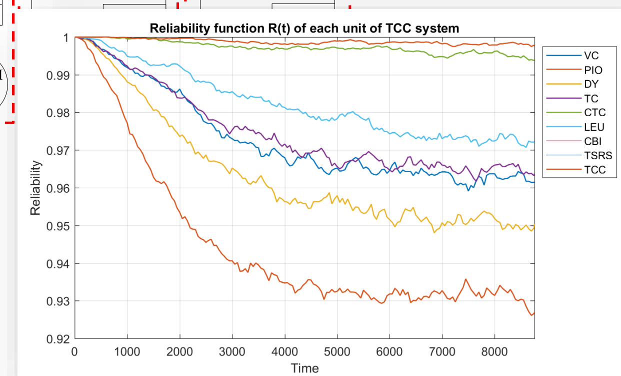


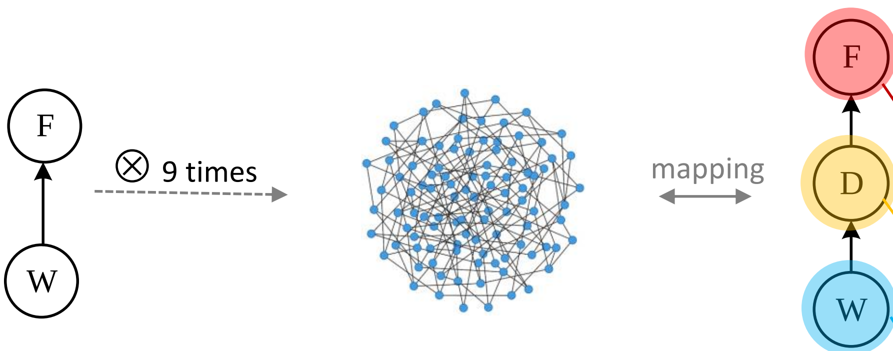


Use Case: TCC System Failure (Railway)

Finite Degradation Calculator







Hot spare FDS

Complete state-space
(2^9 = 512 states)

TCC system FDS

f(system = F) := VC=F ∨ DY=F ∨ CBI=F ∨ TSRS=F

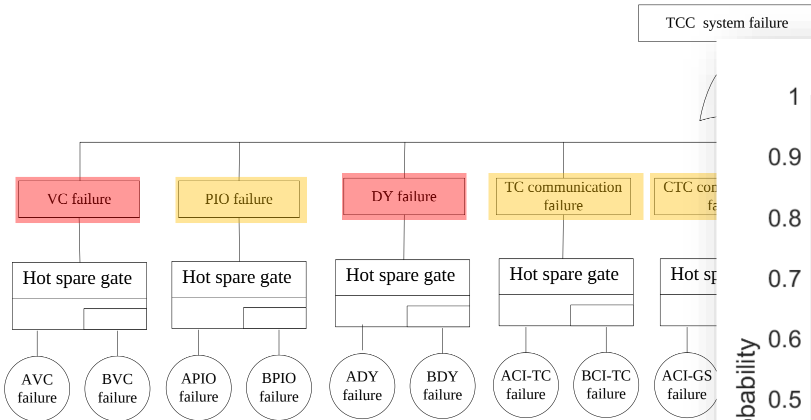
f(system = D) := …

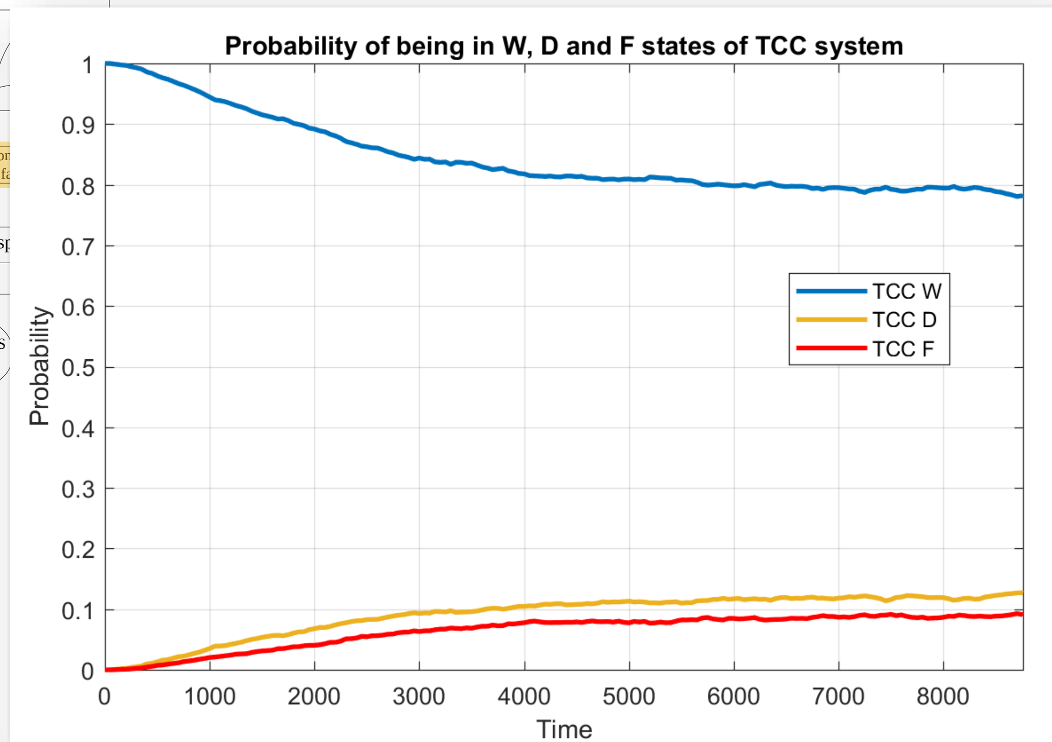
f(system = W) := VC=W ∧ … ∧ TCC=W

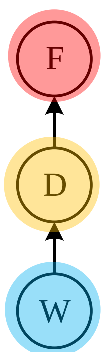
NTNU

Use Case: TCC System Failure (Railway)

Finite Degradation Calculator







TCC system FDS

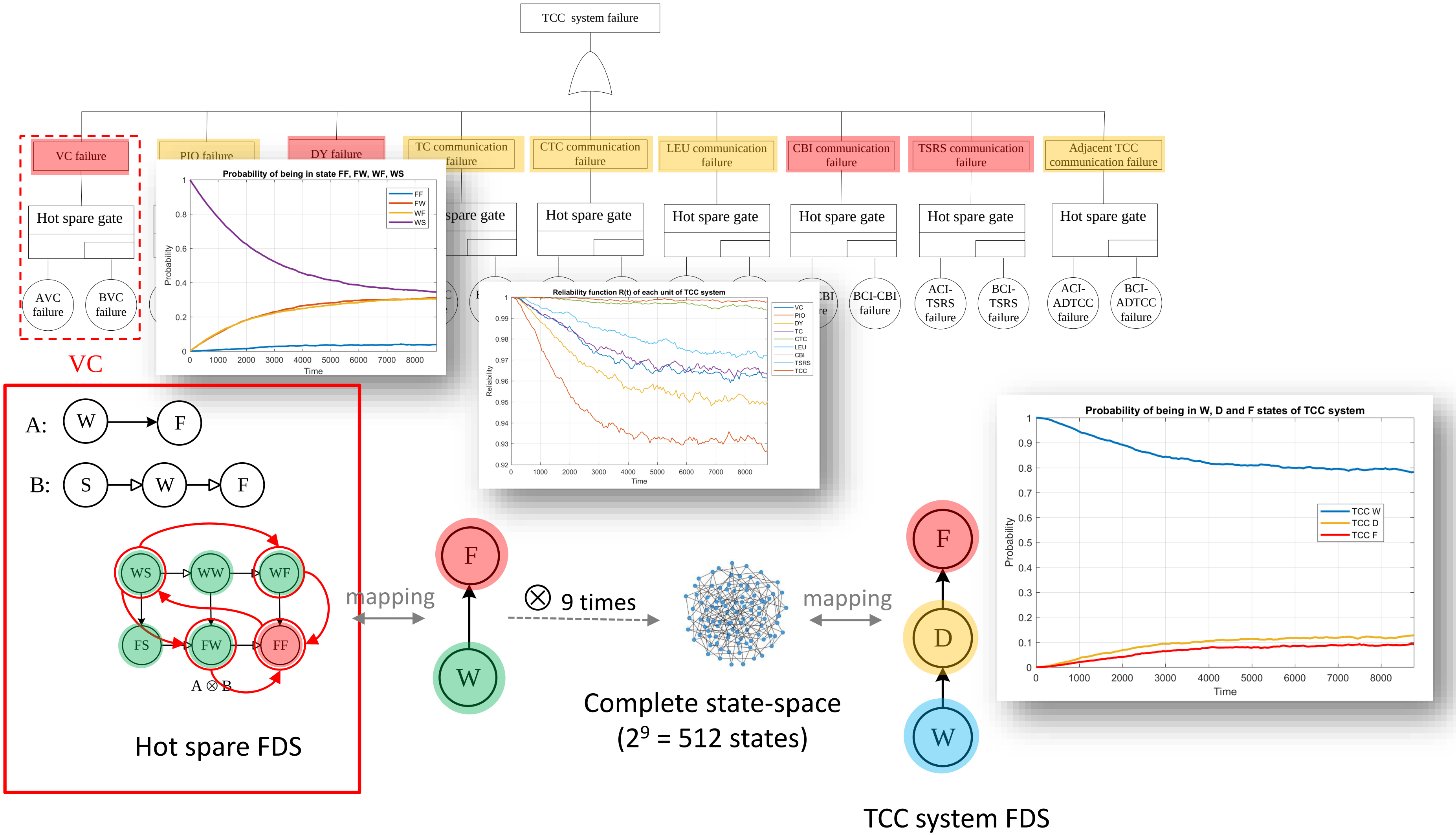
$Pr(W,t) = R_{VC}(t) * R_{PIO}(t) * R_{DY}(t) * R_{TC}(t) * R_{CTC}(t) * R_{LEU}(t) * R_{CBI}(t) * R_{TSRS}(t) * R_{TCC}(t)$

$Pr(F,t) = 1 - R_{VC}(t) * R_{DY}(t) * R_{CBI}(t) * R_{TSRS}(t)$

$Pr(D,t) = 1 - Pr(F,t) - Pr(W,t)$

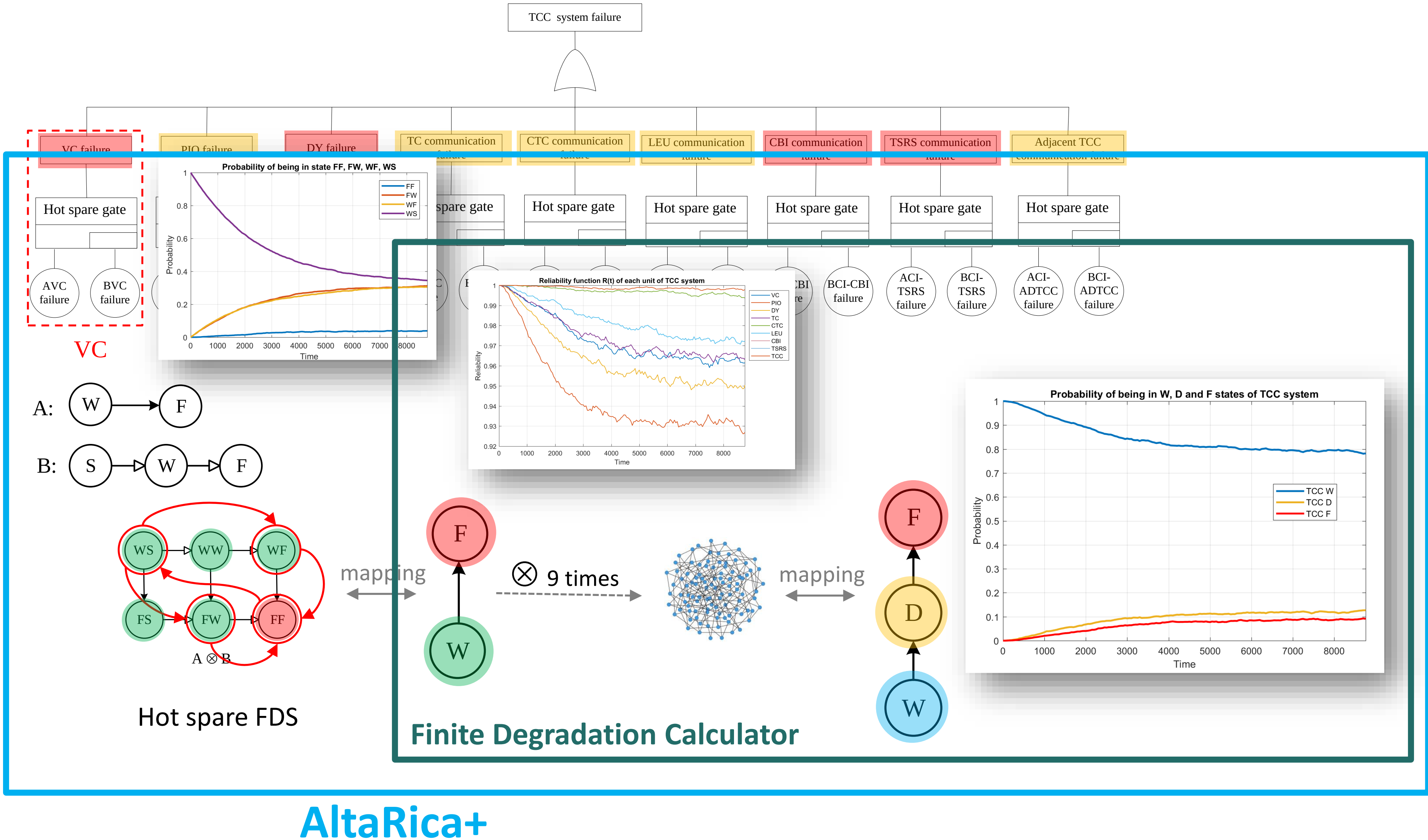
Use Case: TCC System Failure (Railway)

Finite Degradation Calculator



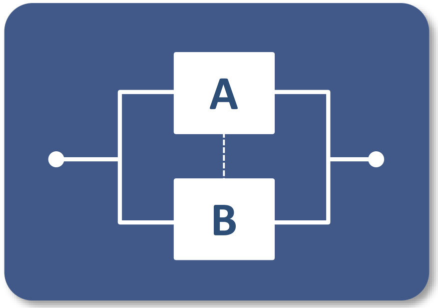
Use Case: TCC System Failure (Railway)

Finite Degradation Calculator



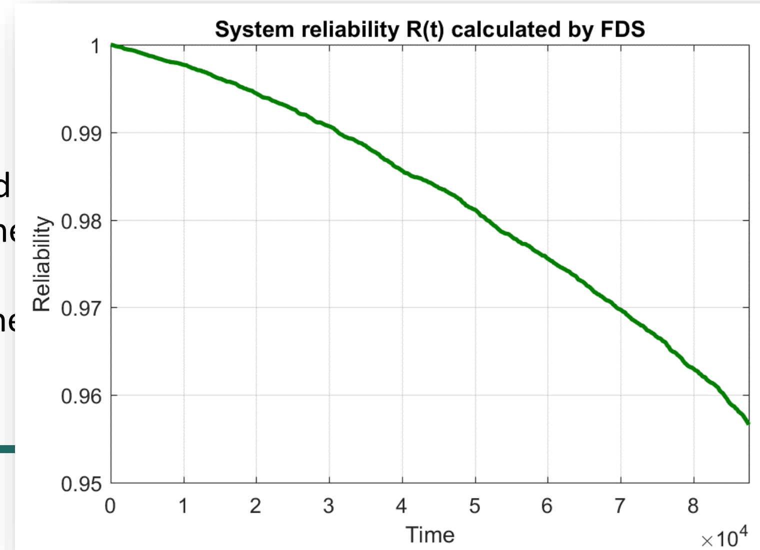
Use Case 2: Cascading Failure

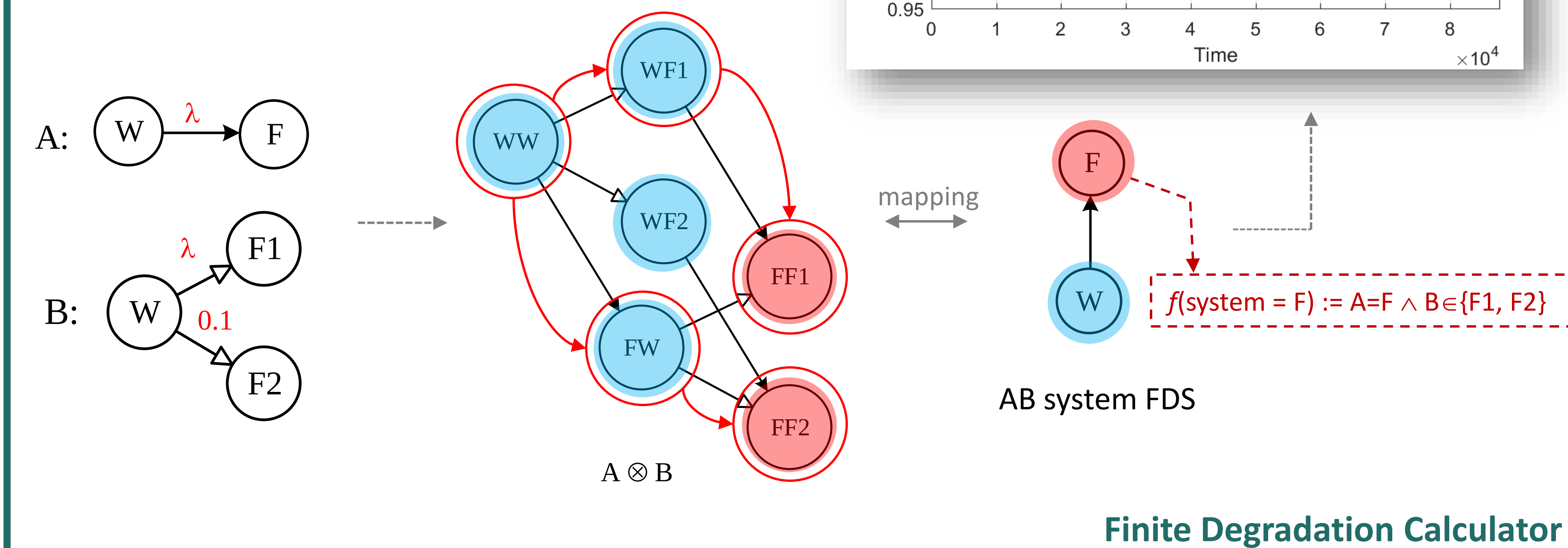
Finite Degradation Calculator



Cascading failure:

- The failure rate of A and
- If A failed firstly, then the of B failed is 0.1
- If B failed firstly, then the of B doesn't change.





Agenda

Background

Finite Degradation Structures

- Why multi-states?
- Definition
- State-space

Finite Degradation Calculator

Use Cases

Future Tool for RAMS Analysis

New Tool for RAMS Analysis

Finite Degradation Calculator

Finite Degradation Calculator

1

Calculate the reliability function R(t) of multi-states system:

- Allow having **multi-states components**
- Allow **parallel/series relationship** between components
- Allow **dependencies**
- …

New Tool for RAMS Analysis

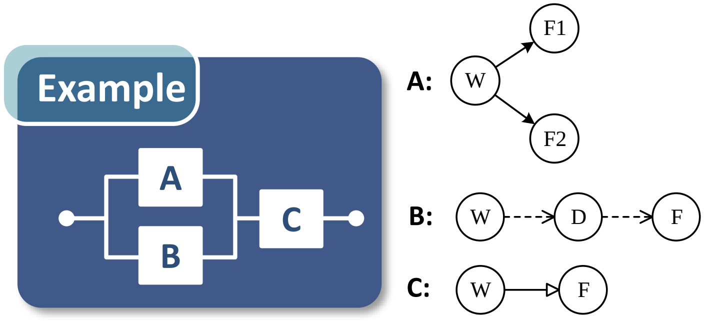
Finite Degradation Calculator

Finite Degradation Calculator

1 **Calculate the reliability function R(t) of multi-states system**

2 **RAMS analysis**

- Generate minimal cut sets
- Identify critical states
- Calculate importance measures
- …



f (system failure) = (**A** = F1 ∧ **B** ≥ D) ∨ **C** = F

New Tool for RAMS Analysis

Finite Degradation Calculator

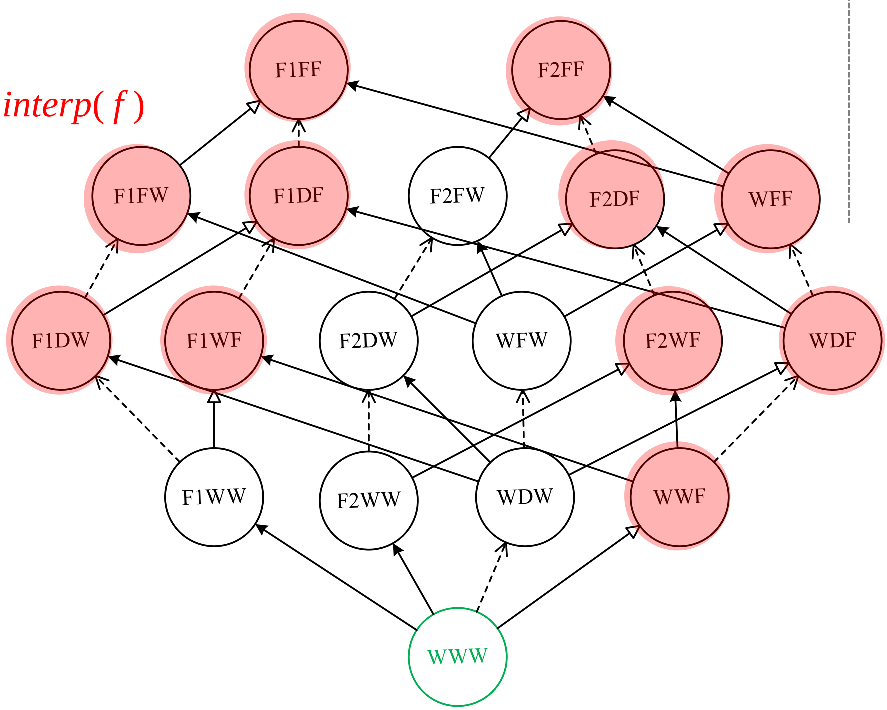
Finite Degradation Calculator

1 **Calculate the reliability function R(t) of multi-states system**

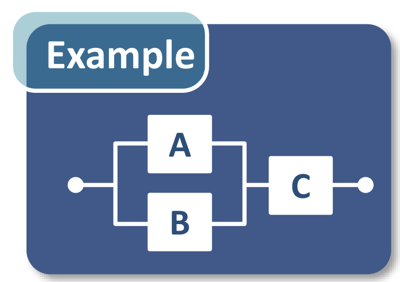
2 **RAMS analysis**

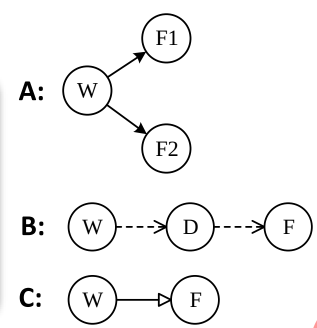
- Generate minimal cut sets
- Identify critical states
- Calculate importance measures
- …

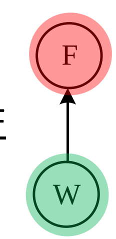
interp(f)



A ⊗ B ⊗ C







f (system failure) = (**A** = F1 ∧ **B** ≥ D) ∨ **C** = F

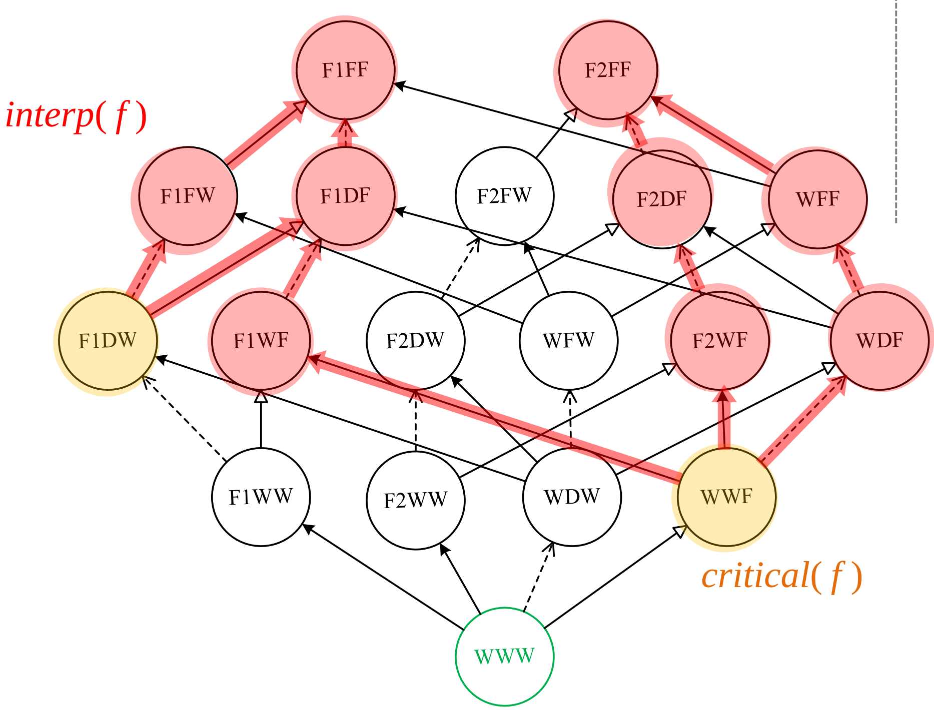
Reliability = $1 - p(f) = 1 - \sum_{s \in Interp(f)} p(s)$

NTNU

New Tool for RAMS Analysis

Finite Degradation Calculator

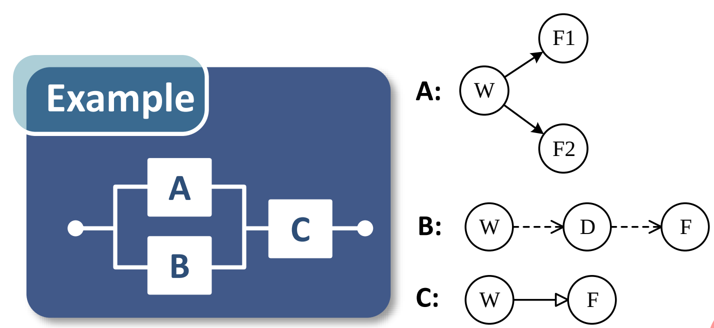
Finite Degradation Calculator



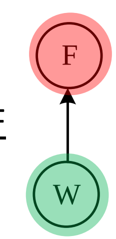
1 **Calculate the reliability function R(t) of multi-states system**

2 **RAMS analysis**

- Generate minimal cut sets
- Identify **critical states**
- Calculate importance measures
- …



f (system failure) = (<u>**A** = F1</u> ∧ <u>**B** ≥ D</u>) ∨ <u>**C** = F</u>



New Tool for RAMS Analysis

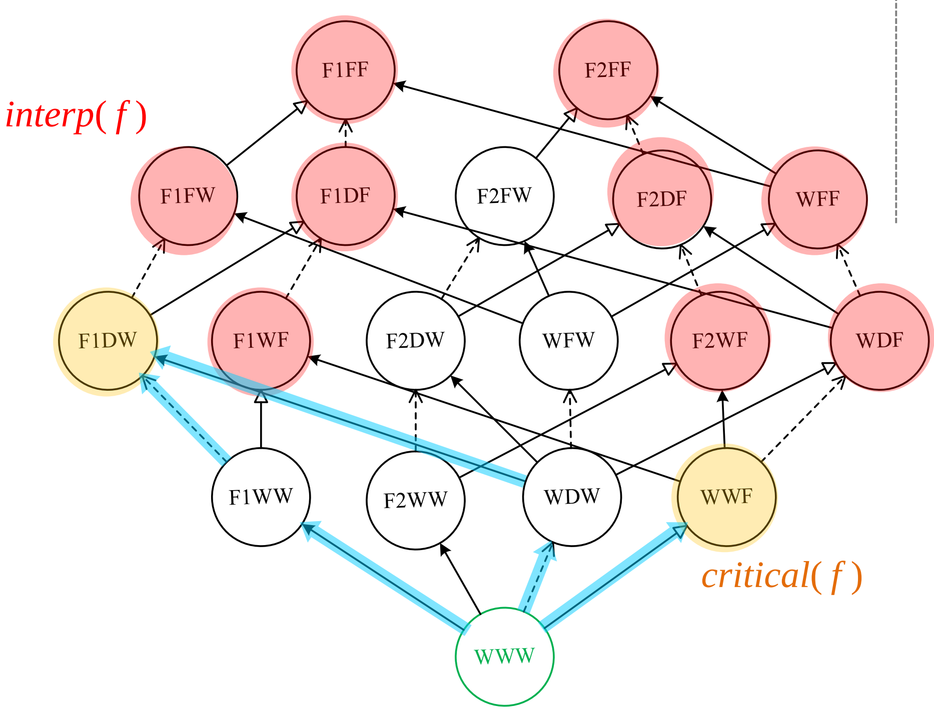
Finite Degradation Calculator

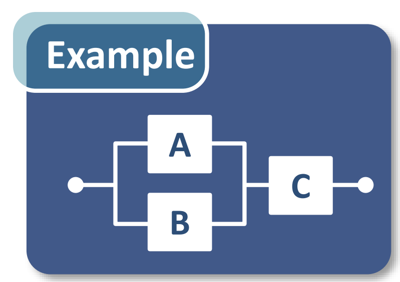
Finite Degradation Calculator

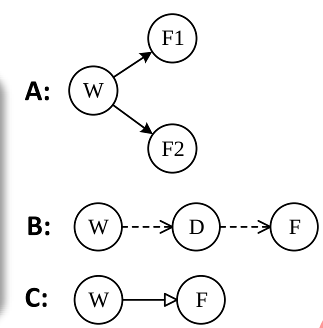
1 Calculate the reliability function R(t) of multi-states system

2 RAMS analysis

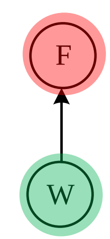
- Generate minimal cut sets
- Identify critical states
- Calculate importance measures
- …







f (system failure) = $(\underline{\mathbf{A} = F1} \wedge \underline{\mathbf{B} \geq D}) \vee \underline{\mathbf{C} = F}$



Future Tool for RAMS Analysis

Finite Degradation Calculator

Finite Degradation Calculator

+

Stochastic Simulator

AltaRica+

Category Theory
Set Theory
First-order Logic
Boolean Algebra
…

Discrete Event Simulation (Monte Carlo Simulation)
Finite-state Automata
Guarded Transition System

Theoretical Bases

- **Cover:**
  Reliability Block Diagrams, Fault Trees, Event Trees, Markov Chains, Petri Nets, Dynamic Fault Trees, Bayesian Networks.
- **Any combination of the models above**
- **Others…**

Future Tool for RAMS Analysis

Finite Degradation Calculator

Finite Degradation Calculator

\+

Stochastic Simulator

AltaRica+

- **Cover:**
  Reliability Block Diagrams, Fault Trees, Event Trees, Markov Chains, Petri Nets, Dynamic Fault Trees, Bayesian Networks.




Finite Degradation Structures

Future Tool for RAMS Analysis