The state of the art in fault tree modeling and analysis

Enno Ruijters

November 11, 2014
About me

- Enno Ruijters
- PhD Student at University of Twente
- ArRangeer project
  - ProRail / STW
  - Improving railroad maintenance using Dynamic Fault Trees and Stochastic Model Checking
Group members

Enno Ruijters

Dennis Guck

prof. dr. ir. Joost-Pieter Katoen
dr. Mariëlle Stoelinga
Introduction to fault trees

- Developed in 1961 by Nuclear Regulatory Agency
- Question: How reliable is your system?
- Now used by:
Introduction to fault trees

- Developed in 1961 by Nuclear Regulatory Agency
- Question: How reliable is your system?
- Now used by:
Why fault trees?

- Some things really should not fail
- Risk assessment is sometimes mandatory
  - Probability of catastrophic failures?
  - Biggest risk factors?
  - Possible mitigations?
Why fault trees?

- Some things really should not fail

  Reliability  Probability of failing within given time
Why fault trees?

- Some things really should not fail
  - **Reliability** Probability of failing within given time
Why fault trees?

- Some thing should not fail for long
  
  **Availability**  Proportion of time in functioning state
Why fault trees?

- Some thing should not fail for long

Availability  Proportion of time in functioning state
What do we want to know?

**Qualitative:**
- Insight into biggest risks
- Relatively fast to perform
- Easy to understand
- Limited information

**Quantitative:**
- Quantify total risk
- Quantify effect of mitigation
- Time consuming
- Hard to estimate numbers
What do we want to know?

Qualitative:

- **Cut sets**: Sets of components causing failure
  
  *Example*: Airplane fails when both engines fail
What do we want to know?

Qualitative:

- **Cut sets**: Sets of components causing failure  
  *Example*: Airplane fails when both engines fail

- **Common cause failures**: Multiple failures with one cause  
  *Example*: Redundant computers running same program
What to we want to know?

Quantitative:

- **Reliability** $\equiv$ Probability of failure within time $t$
  
  *Example*: Probability of containment failure within 25 year nuclear plant lifetime
What do we want to know?

Quantitative:

- **Reliability** \(\equiv\) Probability of failure within time \(t\)
  
  *Example*: Probability of containment failure within 25 year nuclear plant lifetime

- **Availability** \(\equiv\) Proportion of time (in \([0, \infty)\) or \([0, t]\)) spent not failed
  
  *Example*: Amazon EC2 cloud offers SLA of 99.95% uptime
What to we want to know?

Quantitative:

- **Reliability** $\equiv$ Probability of failure within time $t$
  
  *Example:* Probability of containment failure within 25 year nuclear plant lifetime

- **Availability** $\equiv$ Proportion of time (in $[0, \infty)$ or $[0, t]$) spent not failed
  
  *Example:* Amazon EC2 cloud offers SLA of 99.95% uptime

- **MTBF** $\equiv$ Expected time between two successive failures (in finite or infinite horizon)
  
  *Example:* How frequently will my car break down?
Quantitative:

\[ \text{MTTF} \equiv \text{Expected time between system becoming functioning and failing} \]

*Example*: How long will my car run after a service?
What to we want to know?

Quantitative:

\[ \text{MTTF} \equiv \text{Expected time between system becoming functioning and failing} \]

*Example:* How long will my car run after a service?

\[ \text{MTTFF} \equiv \text{Expected time before first failure} \]

*Example:* How long will my new car without failing?
What to we want to know?

Quantitative:

**MTTF** ≡ Expected time between system becoming functioning and failing

*Example*: How long will my car run after a service?

**MTTFF** ≡ Expected time before first failure

*Example*: How long will my new car without failing?

**ENF** ≡ Expected number of failures

*Example*: How many switches will fail in the country per year?
Time-dependent metrics
Fault tree example

- Redundant CPUs
- 1 shared spare memory unit
Fault tree elements

- Basic events (leaves)
- Intermediate Events (gates)
- Top (Level) Event (gate)
- DAG, but often shown as tree with duplicated events

Figure: Images of the gates types in a static fault tree
Example of fault tree failure propagation

- No failures
Example of fault tree failure propagation

- Failure of M1
Example of fault tree failure propagation

- Failure of M1
- Failure of C1
Example of fault tree failure propagation

- Failure of M1
- Failure of C1
Example of fault tree failure propagation

- Failure of M1
- Failure of C1
- Failure of M2
Example of fault tree failure propagation

- Failure of M1
- Failure of C1
- Failure of M2
Example of fault tree failure propagation

- Failure of M1
- Failure of C1
- Failure of M2
Outline

1 Introduction
2 Fault tree analysis
   - Qualitative analysis
   - Quantitative analysis
3 Dynamic fault trees
4 DFT analysis
   - Qualitative analysis
   - Quantitative analysis
5 Other FT extensions
   - FT with uncertainty
   - FTs with dependent events
   - Repairable fault trees
   - FTs with temporal restrictions
   - State-Event fault trees
Measures of interest

Qualitative:
- Cut sets
- Path sets
- Common Cause Failures

Quantitative:
- Reliability
- Availability
- MTBF/MTTF/MTTFF
- Expected number of failures
- Importance values
Qual. analysis: Cut sets

- Set of components causing failure
- Usually minimal cut sets
- Small cut sets like candidates for system improvement
Qual. analysis: Cut sets

- Set of components causing failure
- Usually minimal cut sets
- Small cut sets like candidates for system improvement
- Examples: \{U, B\}
Qualitative analysis: Cut sets

- Set of components causing failure
- Usually minimal cut sets
- Small cut sets like candidates for system improvement
- Examples: \{U,B\}, \{U,C1,C2\}, \ldots
Qualitative analysis: Path sets

- Set of components NOT causing failure
- Usually minimal path sets
- No small path sets can indicate low redundancy
Qual. analysis: Path sets

- Set of components NOT causing failure
- Usually minimal path sets
- No small path sets can indicate low redundancy
- Example: \{B, PS, C1, M1, M2\}
Outline

1. Introduction
2. Fault tree analysis
   - Qualitative analysis
   - Quantitative analysis
3. Dynamic fault trees
4. DFT analysis
   - Qualitative analysis
   - Quantitative analysis
5. Other FT extensions
   - FT with uncertainty
   - FTs with dependent events
   - Repairable fault trees
   - FTs with temporal restrictions
   - State-Event fault trees
   - Cut sets: Boolean manipulation
   - Cut sets: Binary Decision Diagrams
   - Common cause failures
Cut set analysis: Boolean manipulation

- Use boolean algebra to construct DNF
- Bottom-up: Start with leaves
- Top-down: Start with root
Cut set analysis: Boolean manipulation

- Use boolean algebra to construct DNF
- Bottom-up: Start with leaves
- Top-down: Start with root
- Example (top-down):
  
  \[ G_1 = U \land G_2 \quad \text{and} \quad G_2 = B \lor G_3 \]
Use boolean algebra to construct DNF
- Bottom-up: Start with leaves
- Top-down: Start with root
- Example (top-down):
  - $G_1 = U \land G_2$ and $G_2 = B \lor G_3$
  - $G_1 = U \land (B \lor G_3)$
Cut set analysis: Boolean manipulation

- Use boolean algebra to construct DNF
- Bottom-up: Start with leaves
- Top-down: Start with root
- Example (top-down):
  - $G_1 = U \land G_2$ and $G_2 = B \lor G_3$
  - $G_1 = U \land (B \lor G_3)$
  - $G_1 = (U \land B) \lor (U \land G_3)$

![Fault Tree Diagram]

System Failure

- $G_1$
- $G_2$
- $G_3$
- $B$
- $G_4$
- $G_5$
- $G_6$
- $C_1$
- $PS$
- $M_1$
- $C_2$
- $PS$
- $M_2$
- $M_3$

In Use (U)
Cut set analysis: Boolean manipulation

- Use boolean algebra to construct DNF
- Bottom-up: Start with leaves
- Top-down: Start with root
- Example (top-down):
  - \( G_1 = U \land G_2 \) and \( G_2 = B \lor G_3 \)
  - \( G_1 = U \land (B \lor G_3) \)
  - \( G_1 = (U \land B) \lor (U \land G_3) \)
  - \( G_1 = (U \land B) \lor (U \land (G_4 \land G_5)) \)
Cut set analysis: Binary Decision Diagrams

- DAG representing boolean function
- Leaves are 0 or 1
- All paths from the root have the same variable ordering

![Binary Decision Diagram example](image.png)
Common cause failures

- Simultaneous failures of multiple components
- Examples: fire, earthquake, wear of identical components
- Cannot be derived from FT structure
- Expert insight to determine CCF within cut sets
Outline

1. Introduction
2. Fault tree analysis
   - Qualitative analysis
   - Quantitative analysis
3. Dynamic fault trees
4. DFT analysis
   - Qualitative analysis
   - Quantitative analysis
5. Other FT extensions
   - FT with uncertainty
   - FTs with dependent events
   - Repairable fault trees
   - FTs with temporal restrictions
   - State-Event fault trees
   - Fault tree types
   - Bottom-up method
   - Rare-event approximation
   - Bayesian networks
   - Monte Carlo Simulation
Fault tree types

**Time:**
- Discrete-time (one-shot)
- Continuous-time without repairs
- Continuous-time with independent repairs

**Failure distributions:**
- Single probability (discrete-time only)
- Exponential distribution
- Arbitrary distribution
Quantitative analysis: Bottom-up method

- When no events are shared:
  \[ P[X_{AND}(X_1, X_2, \cdots, X_n) = 1] \]
  \[ = P[X_1 = 1 \land X_2 = 1 \land \cdots \land X_n = 1] \]
  \[ = P[X_1 = 1]P[X_2 = 1] \cdots P[X_n = 1] \]
- Likewise for other gates
- Same for availability
Quantitative analysis: Rare event approximation

- Assuming failures are infrequent (e.g. $10^{-9}$)
  - Approximate using $P(A \lor B) \approx P(A) + P(B)$
  - Sum unavailabilities or unreliabilities of cut sets
- Can be made exact using inclusion-exclusion principle:
  - $P(A \lor B) = P(A) + P(B) - P(A \land B)$
Quant. analysis: Bayesian Networks

- General technique used in many probabilistic analyses
- Express fault tree in conditional probabilities
Quantitative analysis: Bayesian Networks

- General technique used in many probabilistic analyses
- Express fault tree in conditional probabilities
- Example (A or (B and C and D)):
Quant. analysis: Bayesian Networks

Advantages:
- Inference using existing tools
- Allows diagnosis
- FT structure persists into model
- Easy to extend with e.g. probabilistic gates

Disadvantages:
- Conditional probability table exponentially large in nr. of inputs
Quantitative analysis: Monte Carlo simulation

- Simulation used in many applications
- Sample failures or failure times, and repair times if needed
- Propagate failures through the tree at every failure or repair
- Track measure of interest through repeated simulations
Monte Carlo Simulation example

- All BEs have failure probability 0.2
- Runs: 0
- Failures: 0
- Estimated reliability:
Monte Carlo Simulation example

- All BEs have failure probability 0.2
- Runs: 1
- Failures: 0
- Estimated reliability: 1
Monte Carlo Simulation example

- All BEs have failure probability 0.2
- Runs: 2
- Failures: 1
- Estimated reliability: 0.5
Monte Carlo Simulation example

- All BEs have failure probability 0.2
- Runs: 3
- Failures: 1
- Estimated reliability: 0.666
Summary

Quantitative analysis techniques:
- Bottom-up method
- Rare-event approximation
- Bayesian networks
- Monte Carlo Simulation

Other techniques:
- Algebraic analysis
- Algebraic approximation
Outline

1. Introduction
2. Fault tree analysis
   - Qualitative analysis
   - Quantitative analysis
3. Dynamic fault trees
4. DFT analysis
   - Qualitative analysis
   - Quantitative analysis
5. Other FT extensions
   - FT with uncertainty
   - FTs with dependent events
   - Repairable fault trees
   - FTs with temporal restrictions
   - State-Event fault trees
Shortcomings of fault trees

- No information about failure sequences
- Poor modeling of shared spare components
- Dependencies cause large trees
- One solution: Dynamic fault trees (DFTs)
Three new gates:

- PAND gate
- FDEP gate
- SPARE gate
DFT Example

Bus

Power Supply

CPU 1

MEM 1

MEM 3

CPU 2

MEM 2

MEM 3

G3

PS

FDEP

C1

SPARE

M1

M3

C2

SPARE

M2

M3

PS

FDEP

C1

SPARE

M1

M3

C2

SPARE

M2

M3
DFT Example

Power Supply

Bus

CPU 1
MEM 1
MEM 2
MEM 3

CPU 2

MEM 1
MEM 2
MEM 3

G3
PS
FDEP
C1
SPARE
M1
M3
C2
SPARE
M2
DFT Example

Power Supply

Bus

CPU 1

MEM 1

CPU 2

MEM 2

MEM 3

G3

PS

FDEP

C1

SPARE

M1

C2

SPARE

M2

M3
DFT Example

Power Supply

Bus

CPU 1

MEM 1

MEM 3

CPU 2

MEM 2

MEM 3

FDEP

PS

G3

C1

SPARE

M1

M3

C2

SPARE

M2

M3
DFT Example

Power Supply

Bus

CPU 1

CPU 2

MEM 1

MEM 2

MEM 3

G3

PS

FDEP

C1

SPARE

M1

M2

M3

C2

SPARE

M1

M2

M3
DFT Example

Power Supply

Bus

CPU 1

MEM 1

MEM 3

CPU 2

MEM 2

G3

FDEP

SPARE

M1

M2

M3

SPARE

C1

C2
**DFT Example**

- Power Supply
- Bus
- CPU 1
- CPU 2
- MEM 1
- MEM 2
- MEM 3
- C1
- SPARE
- C2
- M1
- SPARE
- M2
- M3
- PS
- G3
- FDEP
DFT Example

Power Supply
- Bus
  - CPU 1
  - CPU 2
  - MEM 1
  - MEM 2
  - MEM 3

G3
- C1
  - SPARE
- C2
  - SPARE
- M1
- M2
- M3

PS
- FDEP

DFT Example

Bus
- CPU 1
- CPU 2
- MEM 1
- MEM 2
- MEM 3

Power Supply

G3

C1
- SPARE

C2
- SPARE

M1
- M2
- M3

PS
- FDEP
Outline

1. Introduction
2. Fault tree analysis
   - Qualitative analysis
   - Quantitative analysis
3. Dynamic fault trees
4. DFT analysis
   - Qualitative analysis
   - Quantitative analysis
5. Other FT extensions
   - FT with uncertainty
   - FTs with dependent events
   - Repairable fault trees
   - FTs with temporal restrictions
   - State-Event fault trees
Measures of interest

**Qualitative:**
- Cut/path sets
- Cut sequences

**Quantitative:**
- Reliability
- Availability
- MTBF/MTTF/MTTFF
- Expected number of failures
Outline

1. Introduction
2. Fault tree analysis
   - Qualitative analysis
   - Quantitative analysis
3. Dynamic fault trees
4. DFT analysis
   - Qualitative analysis
   - Quantitative analysis
5. Other FT extensions
   - FT with uncertainty
   - FTs with dependent events
   - Repairable fault trees
   - FTs with temporal restrictions
   - State-Event fault trees

- Cut sets
- Cut sequences
Qualitative analysis

Cut sets for DFTs:

- Failures of cut sets CAN cause system failures, depending on ordering
  - Due to shared spares, failure not always caused by cut sets
- Convert DFT into SFT, by replacing:
  - PAND $\rightarrow$ AND
  - SPARE $\rightarrow$ AND
  - FDEP $\rightarrow$ OR
Example cut sets:

- \{PS\}
DFT cut sets Example

Example cut sets:

- \{PS\}
Example cut sets:

- \{PS\}
DFT cut sets Example

Example cut sets:
- \{PS\}
- \{C1, M1, M2\}
Example cut sets:
- \{PS\}
- \{C1, M2, M3\}
DFT cut sets Example

Example cut sets:

- \{PS\}
- \{C1, M2, M3\}
- NOT \{C1, M1, M2\}
Qualitative analysis

Cut sequences:
- Like cut sets, but include sequence information
- Failure of a cut sequence always causes system failure
- Any system failure is caused by a cut set failure
Example cut sequence:
- \( \langle C1, M1, M2 \rangle \)
Example cut sequence:  
\[ \langle C_1, M_1, M_2 \rangle \]
Example cut sequence:

\[ \langle C_1, M_1, M_2 \rangle \]
Outline

1 Introduction
2 Fault tree analysis
   - Qualitative analysis
   - Quantitative analysis
3 Dynamic fault trees
4 DFT analysis
   - Qualitative analysis
   - Quantitative analysis
5 Other FT extensions
   - FT with uncertainty
   - FTs with dependent events
   - Repairable fault trees
   - FTs with temporal restrictions
   - State-Event fault trees
   - Markov analysis
   - I/O IMC
Analysis by markov chain: 

\[ \lambda_1 = 1 \]
\[ \lambda_2 = 2\lambda_3 = 3 \]
Quantitative analysis: Markov chain

Advantages:
- Exact semantics
- No nondeterminacy
- Reuse of existing modelcheckers (PRISM, etc.)

Disadvantages:
- Semantics are ca. 20 pages long
- Combinatorial explosion
Quantitative analysis: Compositional Markov Analysis

- **Input/Output Interactive Markov Chains** exist of gates and basic events
- Input/Output signals allow parallel composition
- Models of FT elements are composed into one large model
Quantitative analysis: I/O IMC example

\[
\begin{align*}
\lambda_1 &= 1 \\
\lambda_2 &= 2\lambda_3 = 3
\end{align*}
\]

\[
M_{E_2} = \begin{cases} 
\lambda = 2 & f_{E_2}! \\
\text{and} & \\
\text{and} & \\
\end{cases}
\]

\[
M_B = \begin{cases} 
\lambda = 2 & f_{E_2}! \\
\text{and} & \\
\text{and} & \\
\end{cases}
\]

\[
M_{E_2} \parallel M_B = \begin{cases} 
\lambda = 2 & f_{E_2}! \\
\text{and} & \\
\text{and} & \\
\end{cases}
\]

\[
\begin{cases} 
\lambda = 2 & f_{E_2}! \\
\text{and} & \\
\text{and} & \\
\end{cases}
\]

\[
\begin{cases} 
\lambda = 2 & f_{E_2}! \\
\text{and} & \\
\text{and} & \\
\end{cases}
\]

\[
\begin{cases} 
\lambda = 2 & f_{E_2}! \\
\text{and} & \\
\text{and} & \\
\end{cases}
\]
Quant. analysis: Compositional Markov Analysis

**Advantages:**

- Semantics easier to understand
- Intermediate minimization reduces state-space explosion
- Easy to add new gates or events
- Can model nondeterminacy

Note: This is the approach used in DFTCalc and the ArRangeer project.
Quant. analysis: Compositional Markov Analysis

**Advantages:**
- Semantics easier to understand
- Intermediate minimization reduces state-space explosion
- Easy to add new gates or events
- Can model nondeterminacy

**Disadvantages:**
- Still has state-space explosion
- Nondeterminacy
Quantitative analysis: Compositional Markov Analysis

**Advantages:**
- Semantics easier to understand
- Intermediate minimization reduces state-space explosion
- Easy to add new gates or events
- Can model nondeterminacy

**Disadvantages:**
- Still has state-space explosion
- Nondeterminacy

*Note: This is the approach used in DFTCalc and the ArRangeer project*
Other quantitative analysis methods

- Petri Nets
- Dynamic Bayesian Networks
- Modularization of static and dynamic subtrees
- Monte Carlo Simulation
Fuzzy numbers

- Uncertainty and variation in BE probabilities
- Expert judgement not exact
- Possible solution: BE probabilities in fuzzy sets
- Several frameworks for computations on fuzzy numbers
- Can compute same measures as for non-fuzzy FTs.

![Membership function for fuzzy numbers](image_url)
Fuzzy computations

- Combine fuzzy sets using mathematical operations
- Problem: probability distribution unknown
- Various assumptions exist, partly for computational efficiency
- Example: medium + medium = (not low, maybe medium, likely high)

![Graph showing membership functions for low, medium, and high categories with corresponding probability ranges.]
Fuzzy addition

\[ \mu_{A+B}(z) = \max_{z=x+y} (\min\{\mu_A(x), \mu_B(y)\}) \]

Example: \[ \mu_{\text{low}+\text{medium}}(1) = 0.5 \]
Fuzzy arithmetic

- Problem: Fuzzy arithmetic does not return original values
- Various methods to may fuzzy sets back onto descriptors
- In practice: expert judgement
Other uncertain FTs

- ‘Intuitionistic fuzzy set theory’: Membership function uncertain
- Probability distribution for BE failure rates
- Multi-state BE with uncertain states
- Normal distribution approximation
Outline

1. Introduction
2. Fault tree analysis
   - Qualitative analysis
   - Quantitative analysis
3. Dynamic fault trees
4. DFT analysis
   - Qualitative analysis
   - Quantitative analysis
5. Other FT extensions
   - FT with uncertainty
   - FTs with dependent events
   - Repairable fault trees
   - FTs with temporal restrictions
   - State-Event fault trees
Normal FTs assume independent BEs
- Not always realistic (‘valve stuck open’ and ‘valve stuck closed’ are not independent)
- Component failures and degradation may propagate
Extended fault trees

- Components have multiple states (between ‘failed’ and ‘perfectly working’)
- Component state can affect other component failure rates
- Gates allow for different combinations of states
- Textual DSL needed for specification
- Only quantitative continuous-time analysis defined
Extended fault trees

```
DEFINE FAILDEP pump1:
    CAUSE = P1.slow;
    EFFECT = RATECHANGES P2:*2;
END

DEFINE FAILDEP pump2:
    CAUSE = P2.slow;
    EFFECT = RATECHANGES P1:*2;
END
```
Boolean Driven Markov Processes

- BEs and gates represented as multiple Markov Processes (MPs)
- States in the MPs can trigger other elements to switch MPs
- Applications: Changing failure rates, multistate components, new gates
- Disadvantage: Harder to quickly oversee
- Only quantitative continuous-time analysis defined
Other dependent event extensions

- Multiple FTs for different failure modes
- Specifying mutually exclusive events
- Replace BEs by Petri nets
Outline

1. Introduction
2. Fault tree analysis
   - Qualitative analysis
   - Quantitative analysis
3. Dynamic fault trees
4. DFT analysis
   - Qualitative analysis
   - Quantitative analysis
5. Other FT extensions
   - FT with uncertainty
   - FTs with dependent events
   - Repairable fault trees
   - FTs with temporal restrictions
   - State-Event fault trees
Repairable fault trees

- Simple repair model: Simultaneous independent repairs
- Problem: Limited resources for repairs in real life
- Problem: Hidden failures
- Solution method: Repairable Fault Trees
Repairable fault trees

- Add *Repair Boxes* to tree (now becomes cyclic)
- Repair box has one input: repair starts when input fails
- Repair box specifies multiple components to repair
- Repair policy determines how repairs proceed (simultaneous, sequential, combination, etc.)
- Qualitative analysis (cut sets) possible but less useful
- Several quantitative analysis techniques defined
Repair shared components when system fails
Repair CPUs when cluster fails
Outline

1. Introduction
2. Fault tree analysis
   - Qualitative analysis
   - Quantitative analysis
3. Dynamic fault trees
4. DFT analysis
   - Qualitative analysis
   - Quantitative analysis
5. Other FT extensions
   - FT with uncertainty
   - FTs with dependent events
   - Repairable fault trees
   - FTs with temporal restrictions
   - State-Event fault trees
Fault trees with temporal properties

- Static FTs do not consider timing information
- DFTs are one approach to include them, others exist
Add new gate types:

- AND-THEN gate: Requires one event ‘immediately after’ another
  - Formal description with informal predicate
  - Only qualitative analysis defined (extended cut sequence)
FTs with temporal gates

Add new gate types:
- **AND-THEN gate**: Requires one event ‘immediately after’ another
  - Formal description with informal predicate
  - Only qualitative analysis defined (extended cut sequence)
- **POR**: fail when first input fails before others
- **SAND**: fail on simultaneous failure of all inputs
  - PAND + POR + SAND strictly more expressive than AND-THEN gate
  - Only qualitative analysis defined
  - Quantitative analysis seems easy to add
Several approaches add temporal logics to FTs:

- **Cause-consequence gates**
  - Allows indeterminate delays
  - Qualitative analysis for failure-preventing cut sets
  - No other analysis possible

- **Duration calculus**
  - Calculus allows reasoning about delays
  - Not proven decidable
  - No automated analysis available

- **Propositional Linear Temporal Logic**
  - Adds single-input gates like PREV and SOMETIME-PAST
  - Qualitative analysis defined
  - Quantitative analysis probably also possible
Outline

1. Introduction
2. Fault tree analysis
   - Qualitative analysis
   - Quantitative analysis
3. Dynamic fault trees
4. DFT analysis
   - Qualitative analysis
   - Quantitative analysis
5. Other FT extensions
   - FT with uncertainty
   - FTs with dependent events
   - Repairable fault trees
   - FTs with temporal restrictions
   - State-Event fault trees
State-Event Fault Trees

- Practical system failures are sometimes state-dependent
- Especially true of computer software
- SEFT combine state machines with FT gates
- State transitions cause events
- Events and states are combined in gates
- Events can cause state transitions
- Later additions include delays, probabilistic gates
- Quantitative analysis by Petri Nets
State-Event Fault Trees

System Failure

\[
\begin{align*}
\lambda &= 10 \\
\delta &= 1
\end{align*}
\]
Outline

1. Introduction
2. Fault tree analysis
   - Qualitative analysis
   - Quantitative analysis
3. Dynamic fault trees
4. DFT analysis
   - Qualitative analysis
   - Quantitative analysis
5. Other FT extensions
   - FT with uncertainty
   - FTs with dependent events
   - Repairable fault trees
   - FTs with temporal restrictions
   - State-Event fault trees
Future work

- Inclusion of preventive maintenance
- More complex failure models
- Synthesis of maintenance and repair policies