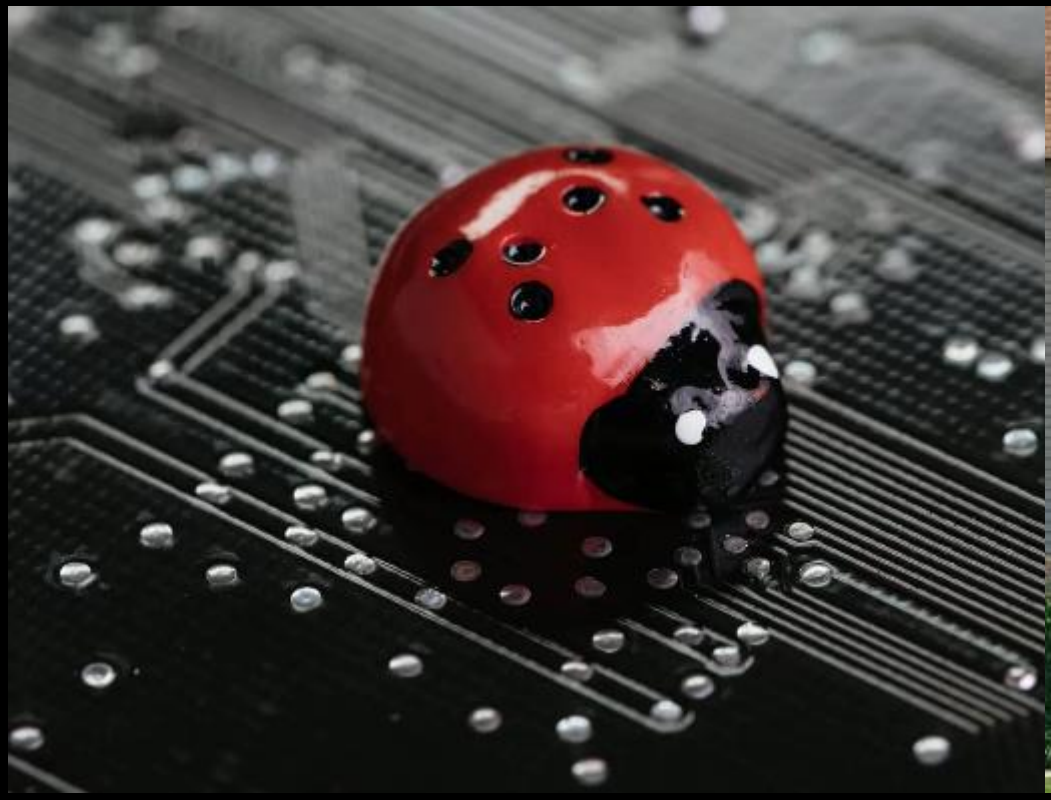
# Lecture 3: FMEAs and DFTs



Milan Lopuhaä-Zwakenberg

Formal Methods & Tools

# Today's Agenda

**Videos**

1. FMEAs
2. Continuous probability
   - Well-made, not easy
3. Dynamic fault trees

Quiz

**Live lecture**

1. Recap: FMEAs
2. Exponentials
3. Dynamic fault trees

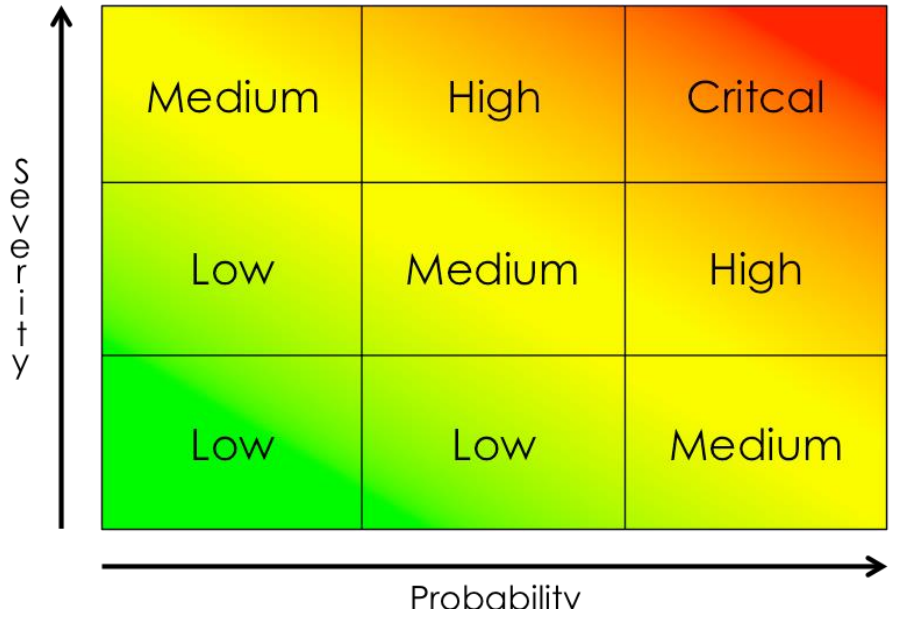
# Planning

| Week | date | Tuesday | Lecturer | Thursday | TA |
|---|---|---|---|---|---|
| 1 | Feb 7 | | | Lecture: intro | MLZ |
| 2 | Feb 14 | Fault trees | MLZ | Exercises | Matthias Volk |
| 3 | Feb 21 | Dynamic FTs, FMEA | MLZ | Exercises | LJR |
| | Feb 28 | **BREAK** | | | |
| 4 | Mar 7 | Classical testing | MLZ | Exercises | LJR,TZ |
| 5 | Mar 14 | State machines | MLZ | Exercises | TZ |
| 6 | Mar 21 | Model-based testing | Petra van den Bos | Exercises | TZ |
| 7 | Mar 28 | Student presentations | You guys | Exercises | TZ |
| 8 | Apr 4 | Guest lecture: mutation testing | Infosupport | Exam practice | LJR, TZ |

# recap: FMEA / FMECA

- Failure Mode Effect Analysis
  - aka Failure Mode,  Effect & Criticality Analysis
- Design tool
  - Assessing risks associated with different ways (modes) in which a system can fail
  - Take appropriate measures
- Textual / worksheet based
  - Many software tools available
  - Integrated in eg Six Sigma
- Focus
  - *Risk priority number (RPN)* =

    Prob of occurrence * severity * detection



# overview: FMEA steps

1. **Determine design tree**
   a. Break down design into systems + components
      - Top-level: Design = what usually we call system = process, mission, ..
      - 2nd level: System = what we usually call subsystems
      - 3rd level: Component
2. **Determine function tree**
   a. Determine functions for each layer
      - Design | System | Component | …
      - Failure = not fulfilling is function
   b. Map functions of components + systems to (functions of) design
   c. Organize functions into tree
3. **Determine failures for each function**
   - Use the function tree
   - Failures: full, partial, intermittent
   - Top-level (design) level failures = effects for user
   - 2nd/3rd system/component level failures = causes; improved to prevent effects
4. **Quantify**
   a. Rate likelihood, detectibility for each mode
   b. determine RPN
5. **Find measures**
   - Are risks acceptable? If not, treat the top risks
6. **Reassess the system**

# Terminology

| Level | Video | Computer science |
|---|---|---|
| Top-level | Design | System |
| 2nd level | System | Subsystem |
| 3rd level | Components | Components |

# FTs vs FMEAs

| **Modeling** | **FTs** | **FMEAs** |
|---|---|---|
| | Top-down | Bottom up |
| | Collecting probabilistic data is time consuming | |
| | insight in relation / dependency between failures: propagation | Insight in potential measures |
| **Analysis** | **FTs** | **FMEAs** |
| Qualitative | cut sets, *path sets* | |
| Quantitative | reliability, availability | Risk Priority Number |

# Today's Agenda

**Videos**

1. FMEAs
2. Continuous probability
   - Well-made, not easy
3. Dynamic fault trees

Quiz

**Live lecture**

1. Recap: FMEAs
2. Exponentials
3. Dynamic fault trees

Poll outcome & Outlook on testing

# System (un)reliability

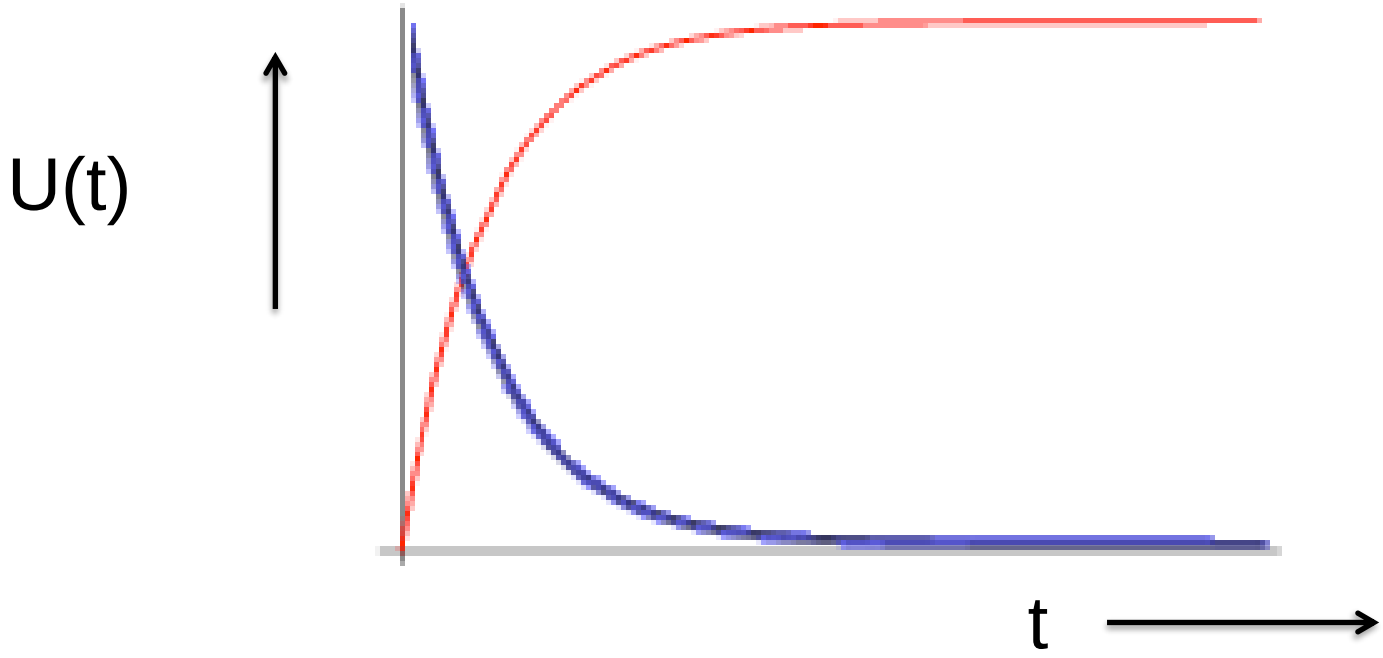
- Unreliability

  U(t) = **P**[failure before time *t*]

- Reliability / Survivor function

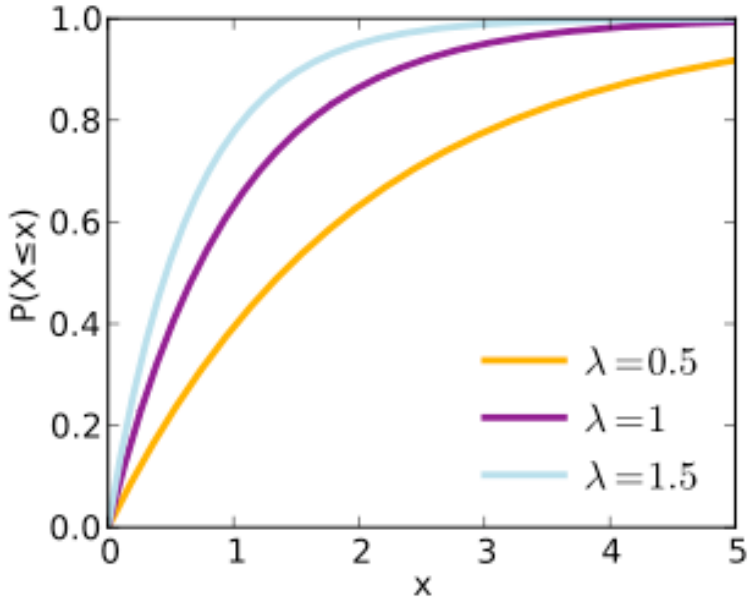
  R(t) = **P**[no failure until time *t*]

  = 1- U(t)



# Exponential distribution: cumulative density function

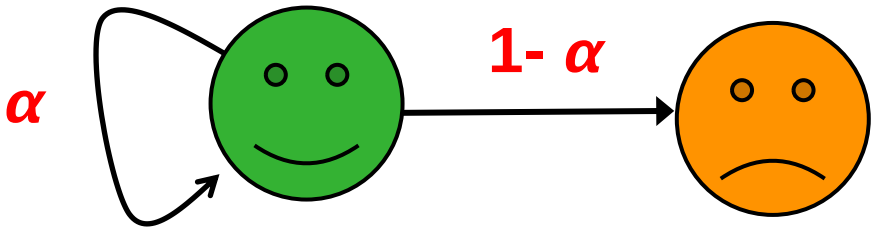
- Unreliability, CDF
  - **P**[fail before $t$] = **P**[$X \leq t$] = $1 - e^{-\lambda t}$
  - $X$: random variable denoting failure time
- Parameter: $\lambda$
  - Failure rate, $\lambda > 0$
  - $\lambda$ = expected number of fails per time unit
  - **E**[$X$] = $1/\lambda$
- Reliability / survivors function, CDF
  - **P**[fail after $t$] =
    = 1 – **P**[fail before $t$]
    = 1- $(1 - e^{-\lambda * t})$
    = $e^{-\lambda * t}$
    = $(e^{-\lambda})^t$ {write $e^{-\lambda} = \boldsymbol{\alpha}$}
    = $\boldsymbol{\alpha}^t$

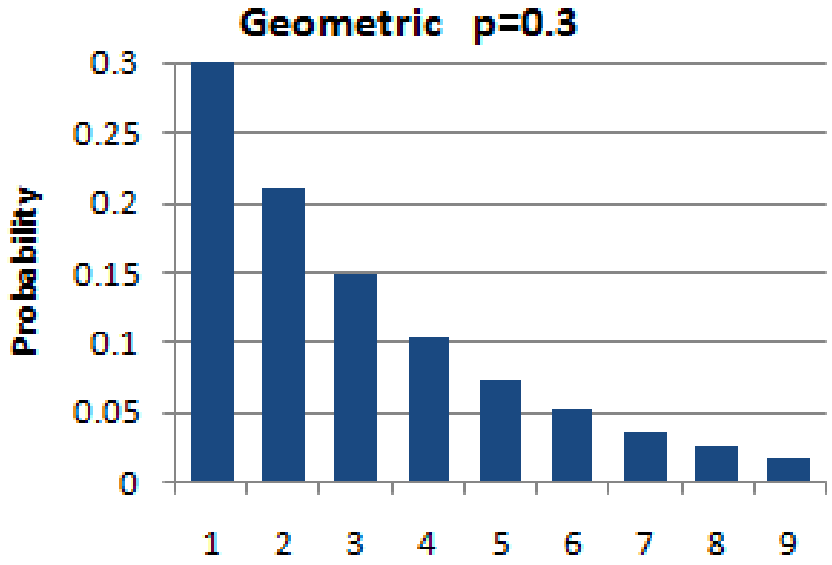


# Exponential distribution: cumulative density function

## Reliability / survivors function, CDF

- **P**[fail after $t$] = **P**[$X > t$] = $\boldsymbol{\alpha}^t$
- "Survive" each time unit with probability $\boldsymbol{\alpha}$
  - t=0: $\boldsymbol{\alpha}^0 = 1$
  - t=1: $\boldsymbol{\alpha}$
  - t=2: $\boldsymbol{\alpha}^2$
  - t=3: $\boldsymbol{\alpha}^3$
  - …
  - t=1/2: $\boldsymbol{\alpha}^{1/2}$
  - t=14.7302: $\boldsymbol{\alpha}^{14.7302}$




*Note:* picture displays the discrete version, for fixed time steps

# Exponential distribution: memoryless

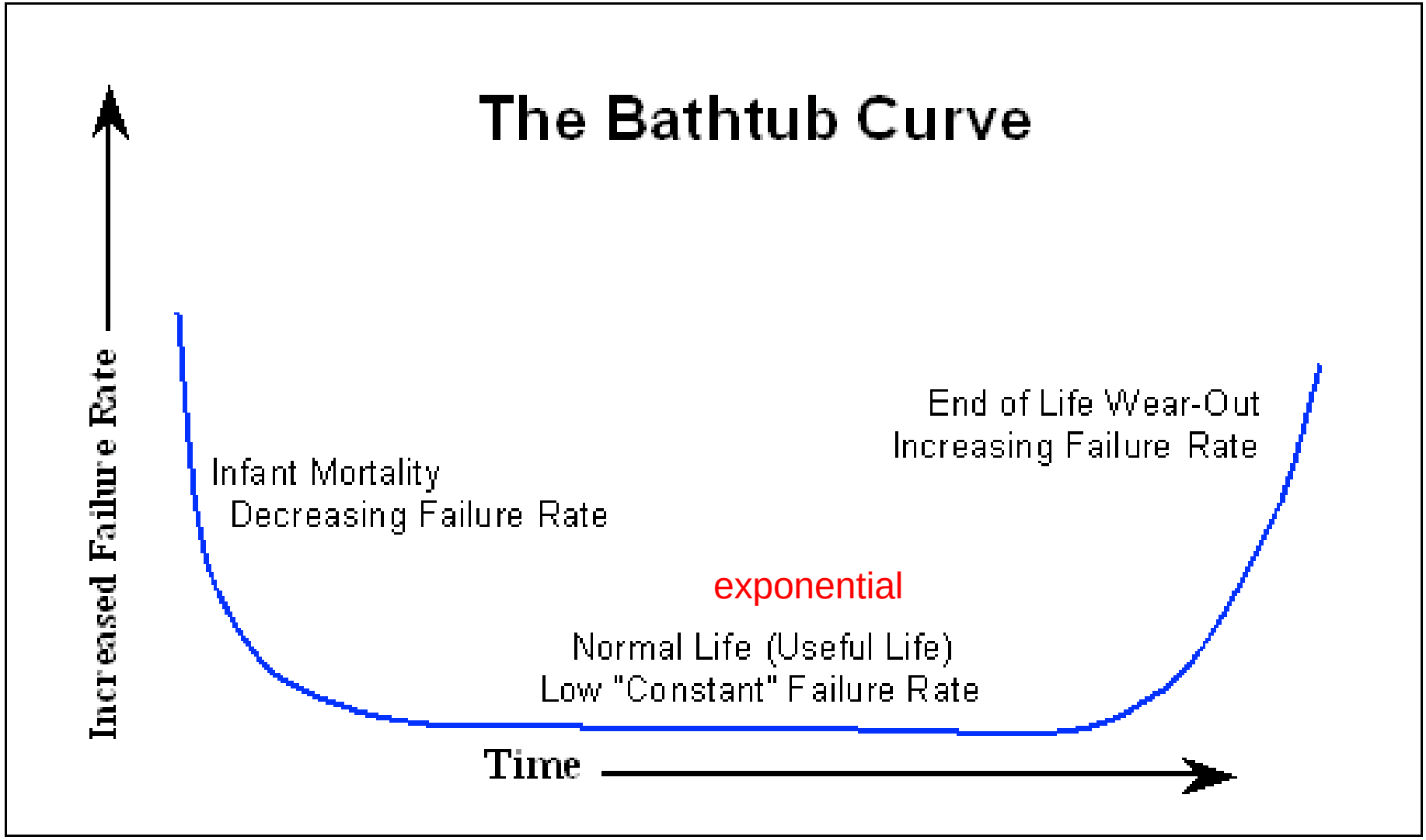
- **Memoryless**
  - Makes this distribution easier
  - Crucial for use in transition systems
- $\mathbf{P}[X > t + s \mid X > s] = \mathbf{P}[X > t]$

*No matter how long you have already been operational,*

*the probability that remain for operational for $t$ more time units is always $\mathbf{P}[X > t]$*

$$\underbrace{\mathbf{P}[X > t+s \mid X > s]}_{\text{Given that you were operational for } s \text{ time units, remain operational for at least } t+s \text{ most units = you must remain operational for at least } t \text{ more time units}} = \underbrace{\mathbf{P}[X > t]}_{\text{Remain operational for } t \text{ more time units Independent of } s}$$

# Bath tub curve: degradation behaviour of systems



# Today's Agenda

**Videos**

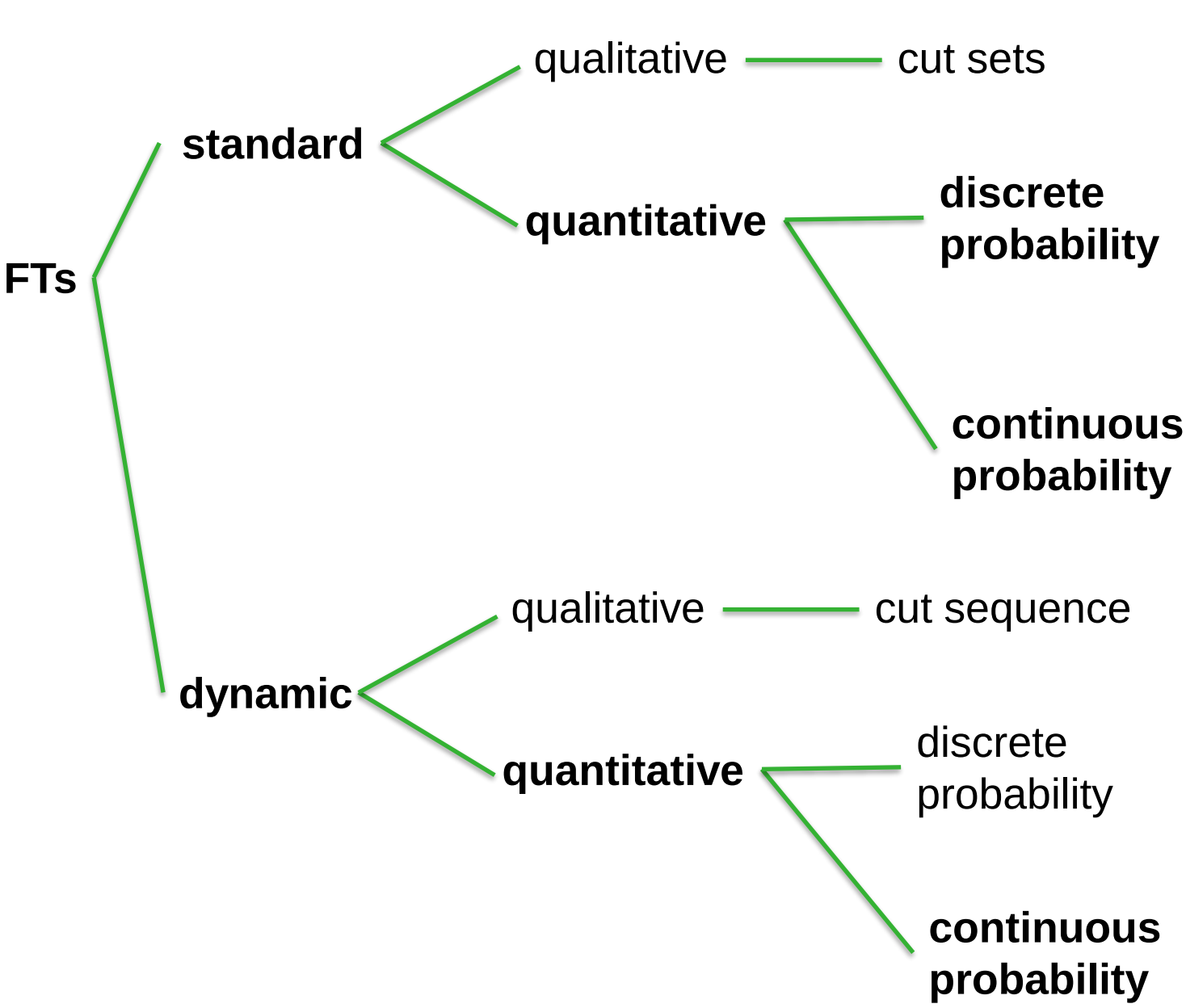
1. FMEAs
2. Continuous probability
   - Well-made, not easy
3. Dynamic fault trees

Quiz

**Live lecture**

1. Recap: FMEAs
2. Exponentials
3. Dynamic fault trees

# Overview

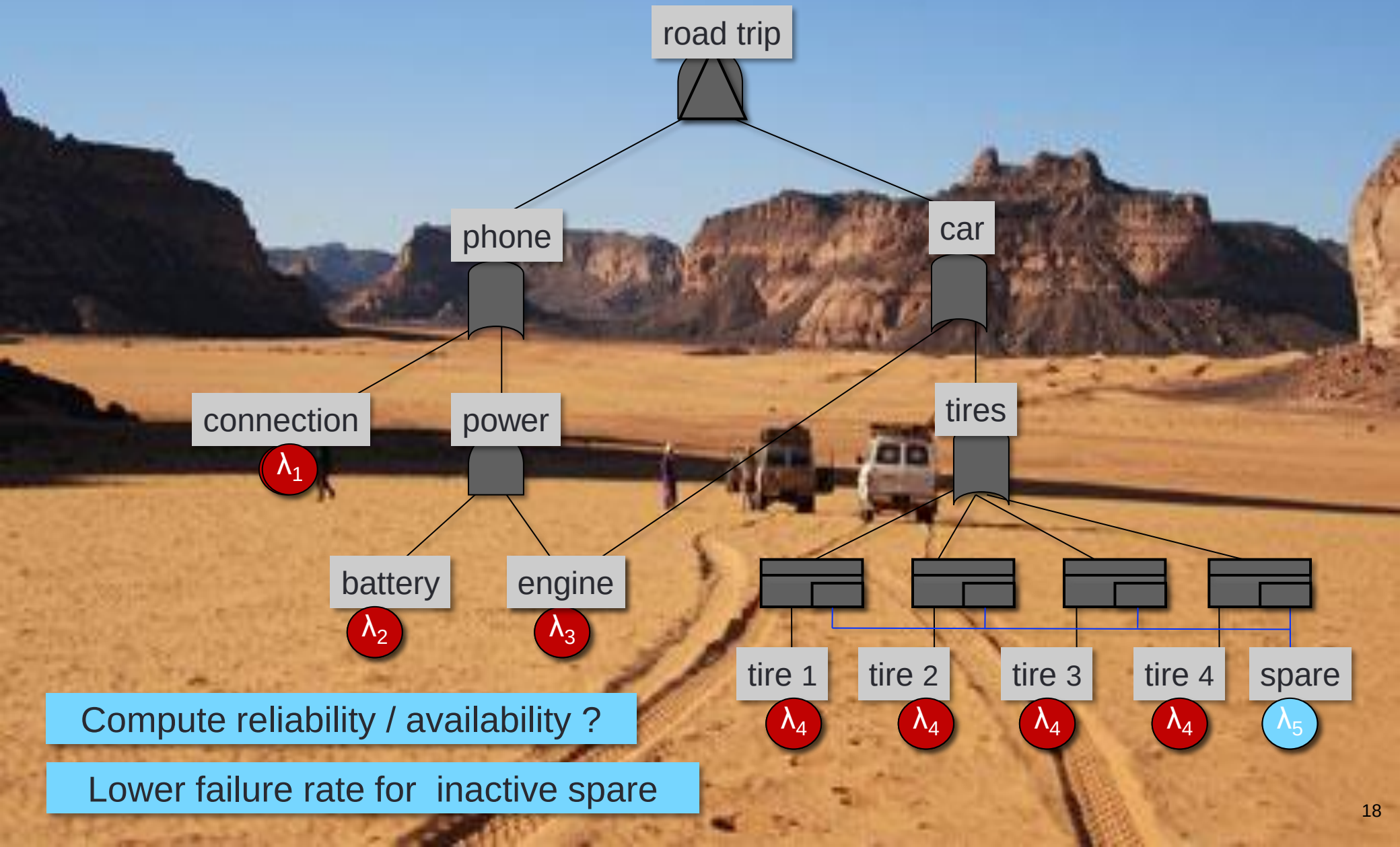



**Technique**
- Recursive
- BDDs

**Technique**
- Bottom up
- Cut sets
- BDDs

**Today**
- Composition
- Minimization
- Exponentials

# Example: Safe road trip




Compute reliability / availability ?

Lower failure rate for inactive spare

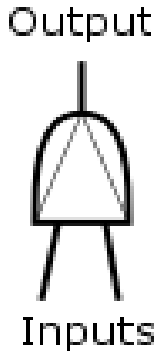
# How to analyse DFTs?

**Problem**

- Failure order matters → BDDs do not work

**Solution: (Stochastic) transition systems**

- States
  - Component failures
  - active | failed | dormant
- Transitions
  - Failure order
- Failure rates
  - Speed of transition
  - **P**[transition within *t* sec] = $1 - e^{-\lambda t}$

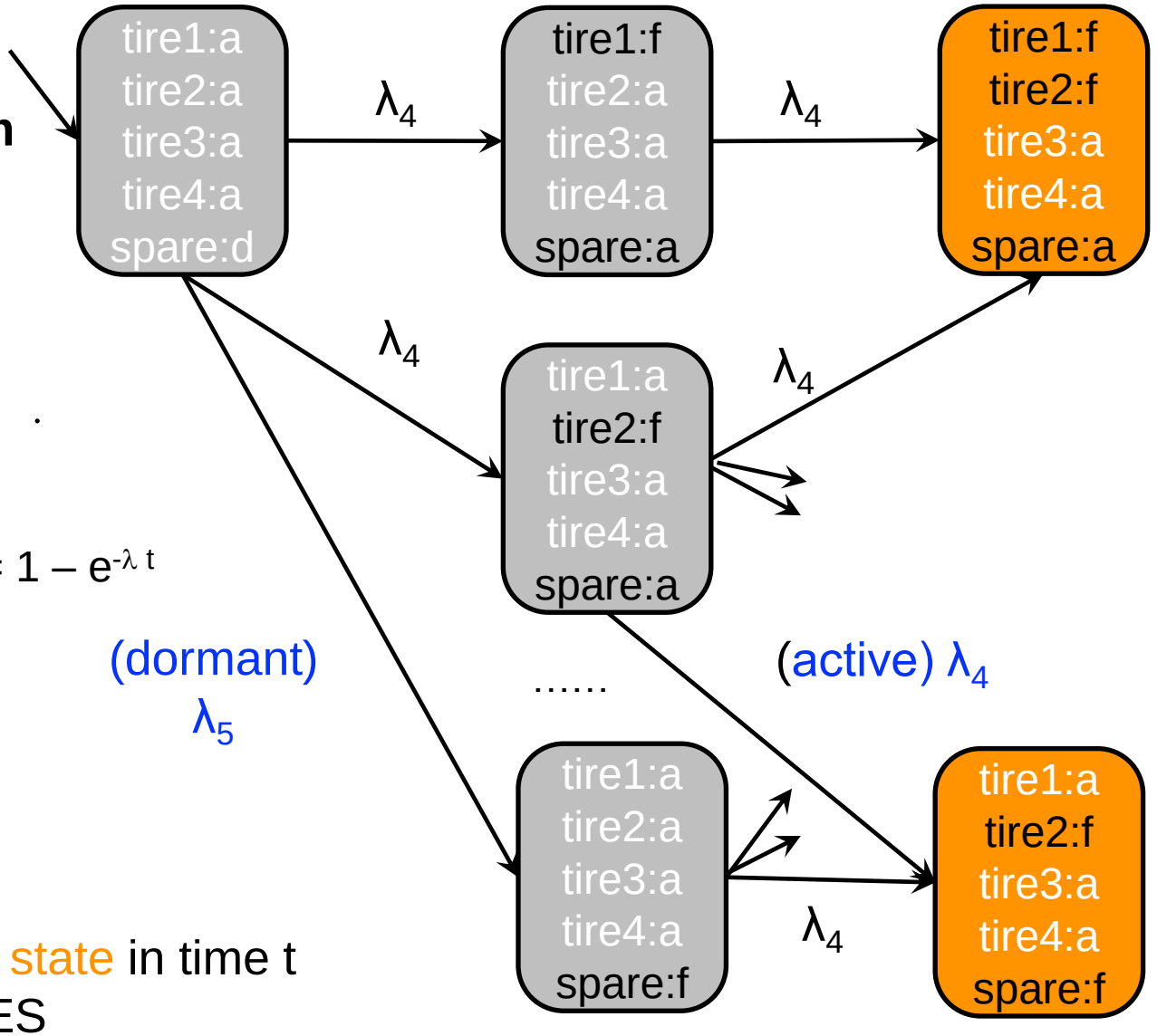



PAND

# Example: TIRES subtree

**Stochastic transition system**

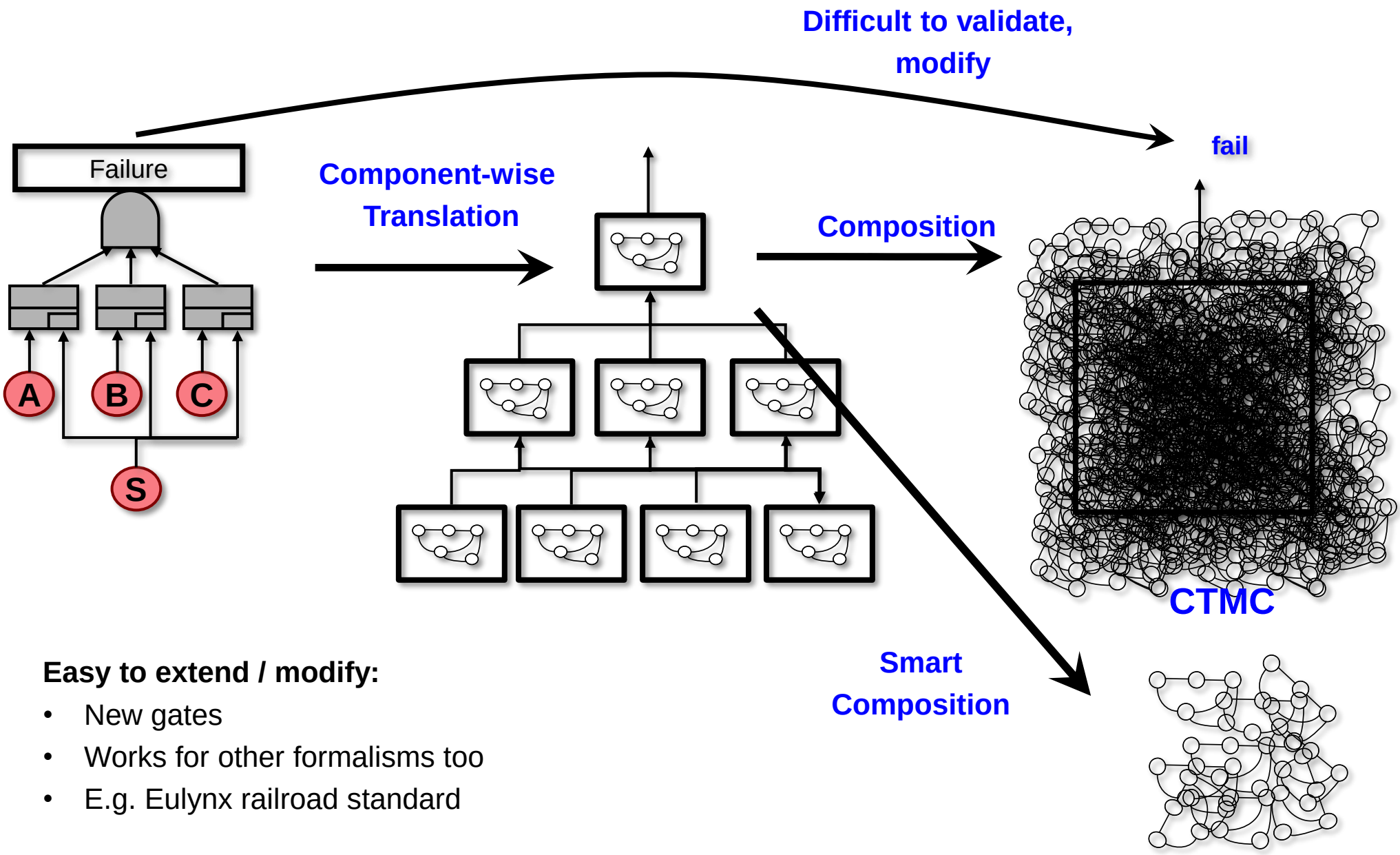
- States
  - Component failures
  - active | failed | dormant
- Transitions
  - Failure order
- Failure rates
  - Speed of transition
  - **P**[transition within t sec] = $1 - e^{-\lambda t}$




**Reliability**

- Probability to reach failed state in time t
- Many algorithms: see QEES

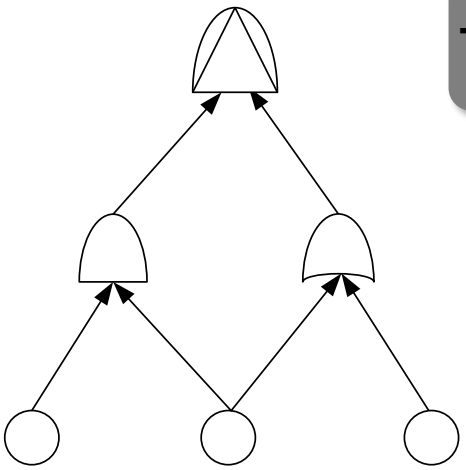
# Fault tree analysis via deep compositionality



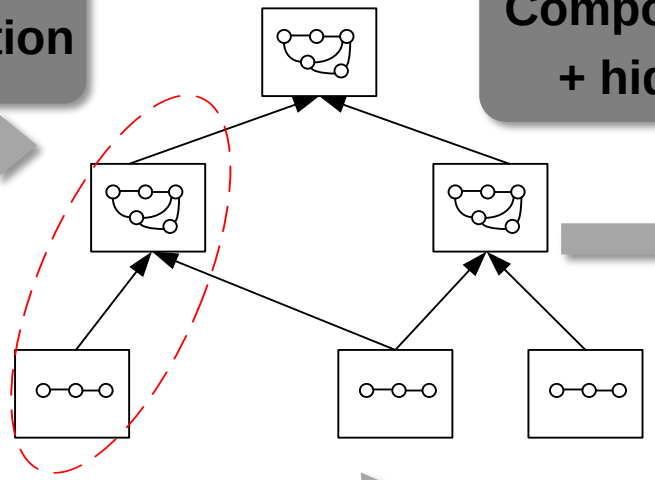

**Easy to extend / modify:**

- New gates
- Works for other formalisms too
- E.g. Eulynx railroad standard

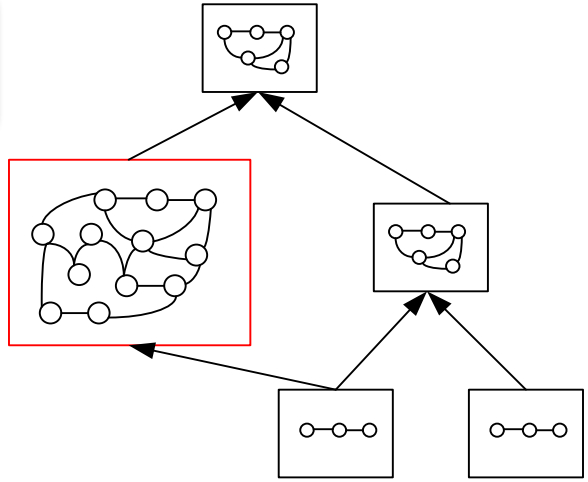
# Deep compositionality



Translation



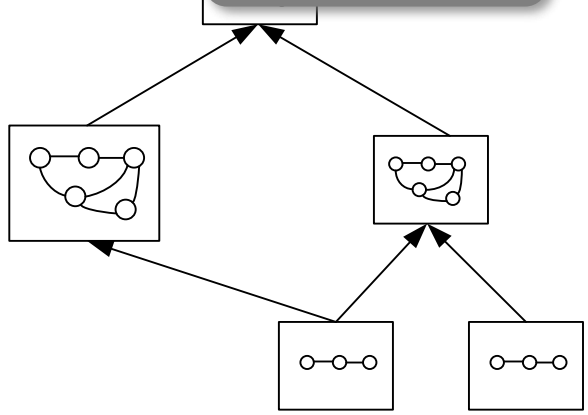
Composition + hiding



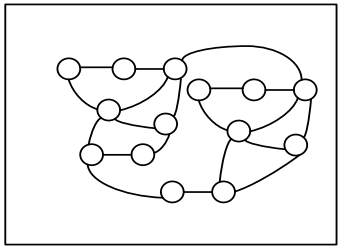
Order matters (size only!)

Repeat

minimization



State space generation



Stochastic mode

Analysis

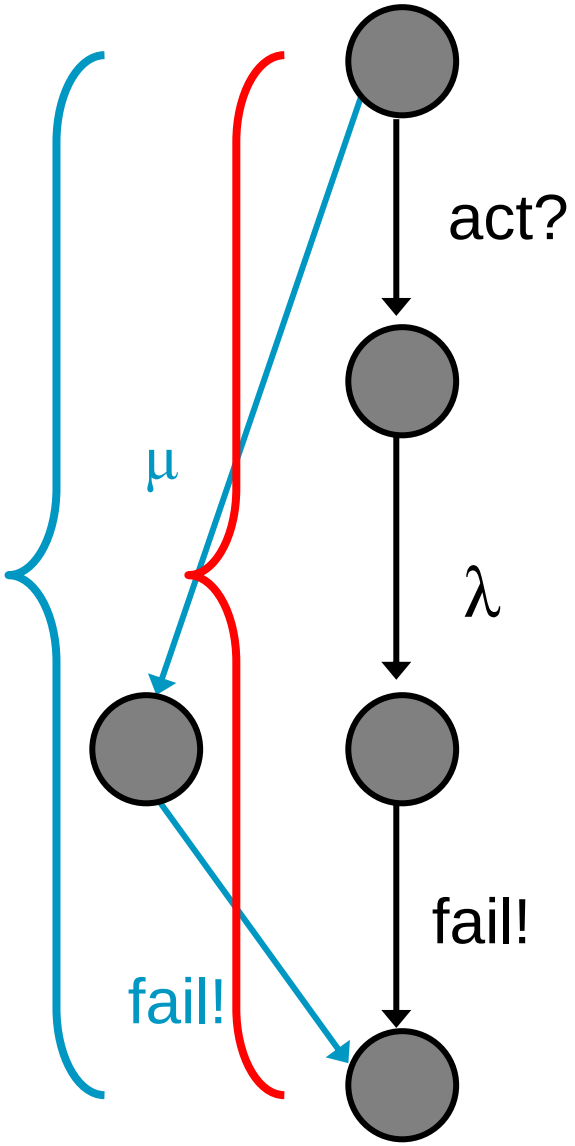
Result: unreliability

# Interactive Markov Chains (I/O-IMC)

- Transition sytems + I/O + CTMC
- Markovian transitions (CTMC)
  - labeled with rates λ
  - delays governed by exponential distribution
  - **P**[transition within t sec] = $1 - e^{-\lambda t}$
- Interactive transitions (I/O)
  - labeled with actions
  - synchronization
- Action signature
  - ? - Input actions: *delayable*
  - ! - Output actions: *immediate*
  - ; - Internal actions: *immediate*

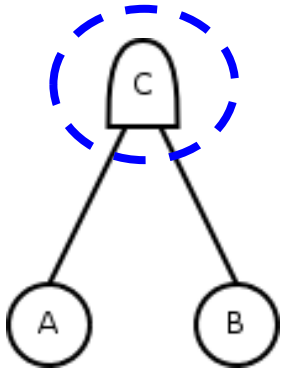
I/O-IMC for cold basic event

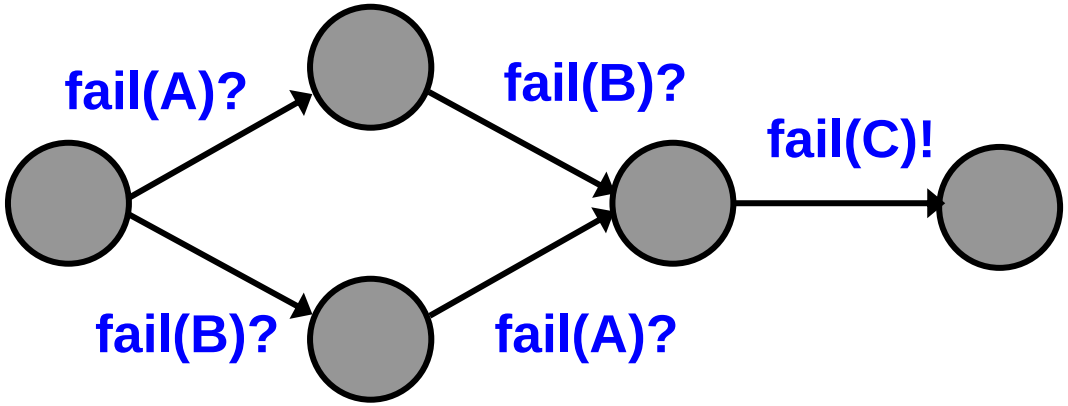
Warm BE: can fail before activation, at reduced rate



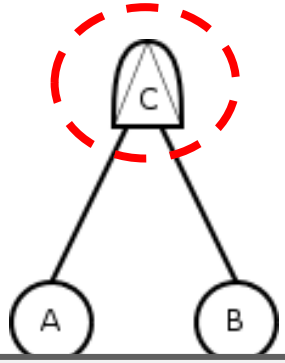
# Semantics for FT gates

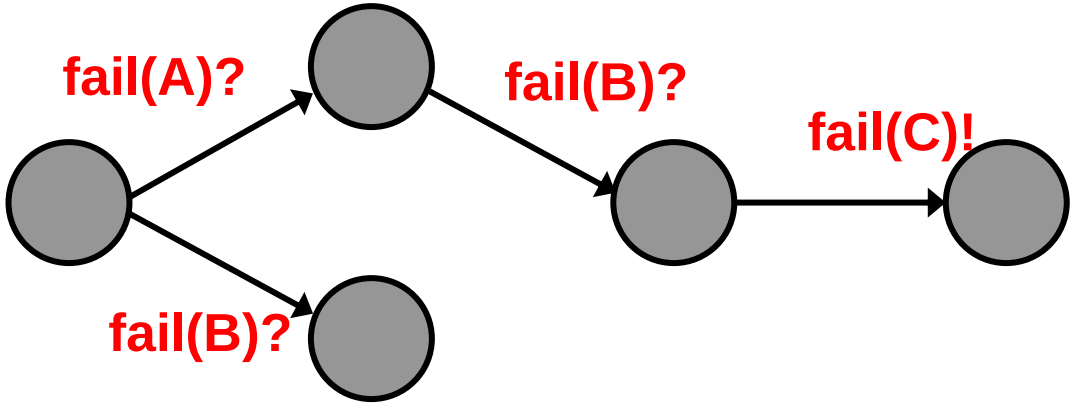
**AND gate**



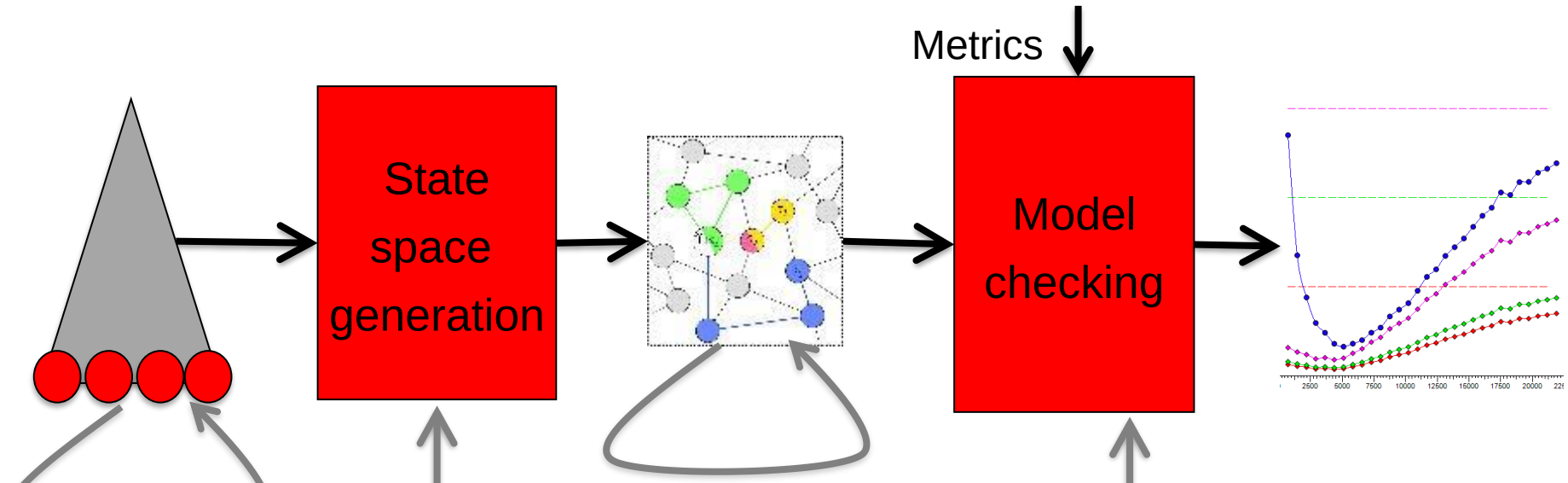

**PAND gate**




**Similar, but more complex for other gates**

# FT analysis: more, faster, bigger, better




**FTs on a diet**

Graph rewriting

**Reduce, reduce, reduce**

- symmetry reduction
- partial-order reduction
- don't care detection

**Smart state space generation**

- Compositionality
- don't care detection
- Partial: optimistic / pessimistic

**Smart algos**

- Stochastic MC
- Metric specific computations
- Statistical MC
- Rare event simuluation

**Fitting all together: meta modeling**


HAPPY
WEEK!