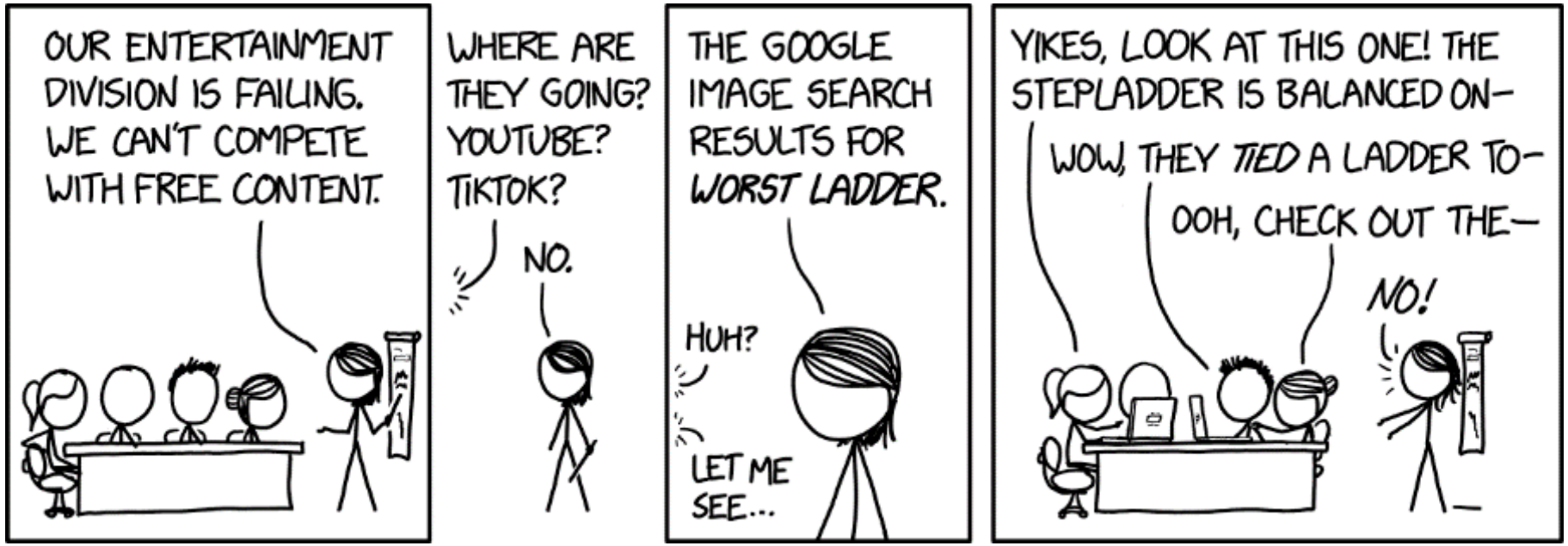
Lecture 2: Fault Tree Analysis



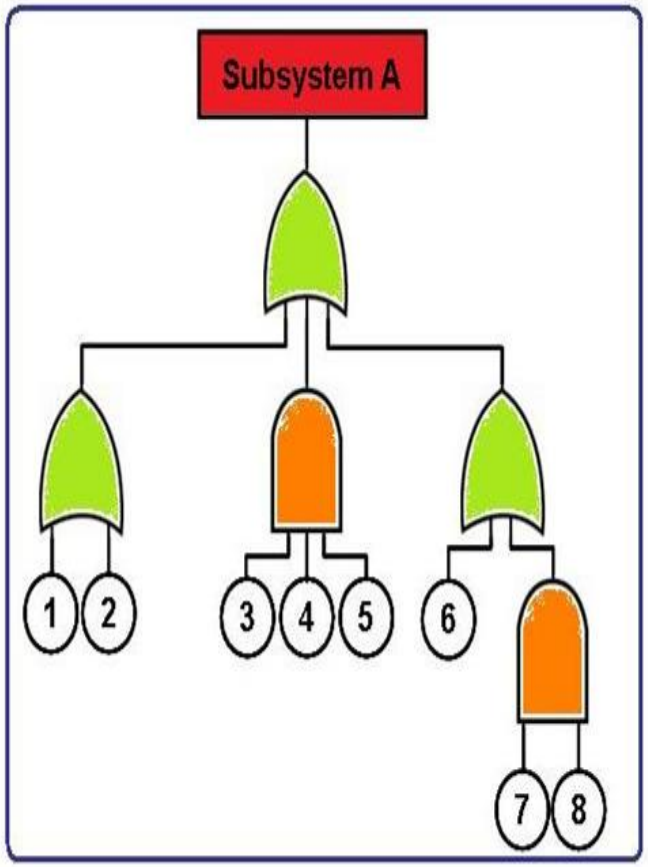
Milan Lopuhaä-Zwakenberg

Formal Methods & Tools

Last time

1. Risk = *effect of uncertainty on objectives*
2. Risk matrix
3. Mitigation strategies
4. Risk models: brainstorming + structured analysis
5. FMEA:
    1. Find many risks
    2. Prioritize them

Fault trees



- *Graphical model*
  - Why does the system fail?
  - Decompose complex risks
- *Qualitative analysis*
  - Root causes, critical components
- *Quantitative analysis*
  - Reliability analysis
  - Time-dependence

Industrial tools: IsoGraph, RiskSpectrum

FTA on Twitter

- Falcon 7 rocked, SpaceX
- Led by Elon Musk, Tesla
- Ready for launch June 2015

Elon Musk @elonmusk · Jun 28

That's all we can say with confidence right now. Will have more to say following a thorough fault tree analysis.

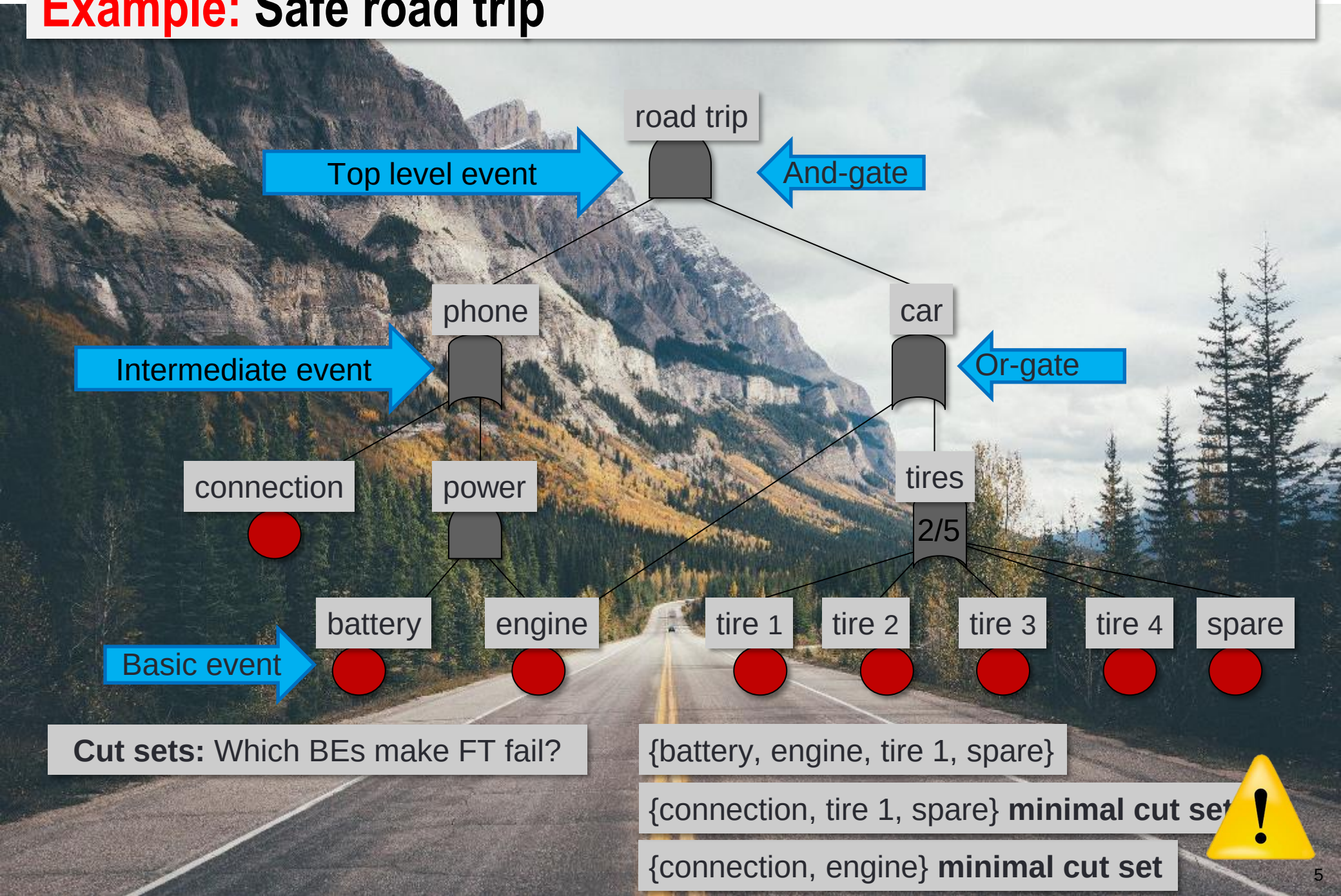
1.3K 1.9K View conversation

Elon Musk @elonmusk · Jun 28

There was an overpressure event in the upper stage liquid oxygen tank. Data suggests counterintuitive cause.

4.3K 3.3K

Example: Safe road trip



Cut sets: Which BEs make FT fail?

{battery, engine, tire 1, spare}

{connection, tire 1, spare} **minimal cut set**

{connection, engine} **minimal cut set**

Fault tree (definition)

A fault tree is:

- A directed, acyclic graph (not always a tree!)
- Nonleaves (*intermediate* events) are AND/OR-gates
- Leaves are called *basic events*

Cut set

- Subset of BEs that make the FT fail

Minimal cut sets

- Cut set that is minimal, i.e., no BE can be removed
- This is *not* the same as ‘smallest cut set’!

(Minimal) cut sets: why do we care?

1. Validation of FTs
   - If all BEs in the cut set fail, does the system fail?
   - Aka: “is this correct?”
2. Identification of weak points
   - Cut sets too small
   - 1 element CS = single point of failure
   - Ask the experts
3. Common cause failures
   - Do elements in a CS have a common cause?
4. Risk prioritization
   - Via cut set metrics (next slide)
5. Calculating failure probability
   - Later today

Risk prioritization: cut set metrics

Metrics

- How important is a cut set?
- How important is a basic event?

Cut set metrics

- **Order**
  - Number of cut set elements
  - Low order = more vulnerable
- **Frequency of basic event** *e*
  - How many minimal cut sets contain *e*?
  - High frequency = *e* has large impact on safety
- **Probability**
  - Product of BE probabilities
  - $p_1 p_{2...} p_n$ if BE e_i has probability p_i

Min cut set	prob
connection, engine	$p_{con} * p_{eng}$
battery, engine	$p_{bat} * p_{eng}$
connection, t1, t2	$p_{con} * p_{tire}^2$
connection, t1, t3	$p_{con} * p_{tire}^2$
connection, t1, t4	$p_{con} * p_{tire}^2$
connection, t1, spare	$p_{con} * p_{tire}^2$
connection, t2, t3	$p_{con} * p_{tire}^2$
connection, t2, t4	$p_{con} * p_{tire}^2$
connection, t2, spare	$p_{con} * p_{tire}^2$
connection, t3, t4	$p_{con} * p_{tire}^2$
connection, t3, spare	$p_{con} * p_{tire}^2$
connection, t4, spare	$p_{con} * p_{tire}^2$

Fault trees pros and cons

Advantages

- Graphical = easy to explain and understand
- Compositional = easy to incorporate expert input
- Detailed analysis (cut sets, probability)

Disadvantages

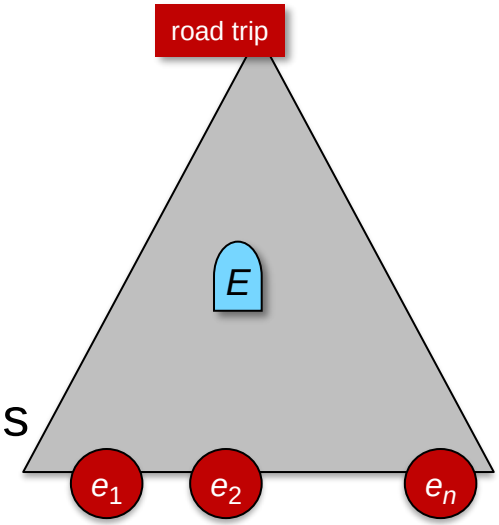
- Impact is binary
- Relation between events is binary
- Probabilistic analysis requires accurate statistics

The fault trees: structure function

Structure function $\Phi_F: \{0,1\}^{\#\text{BEs}} \rightarrow \{0,1\}$

1 = fail; **0** = operational

- Given values $e_1,\ldots,e_n$, $\Phi_F(\vec{e})$ tells whether *F* fails



Extension: Φ_F: $\{0,1\}^{\#\text{BEs}}$ x Elt → {0,1}

- Given values $e_1,,\ldots,e_n$, $\Phi_F(\vec{e}, E)$ tells whether element *E* fails

Recursive definition

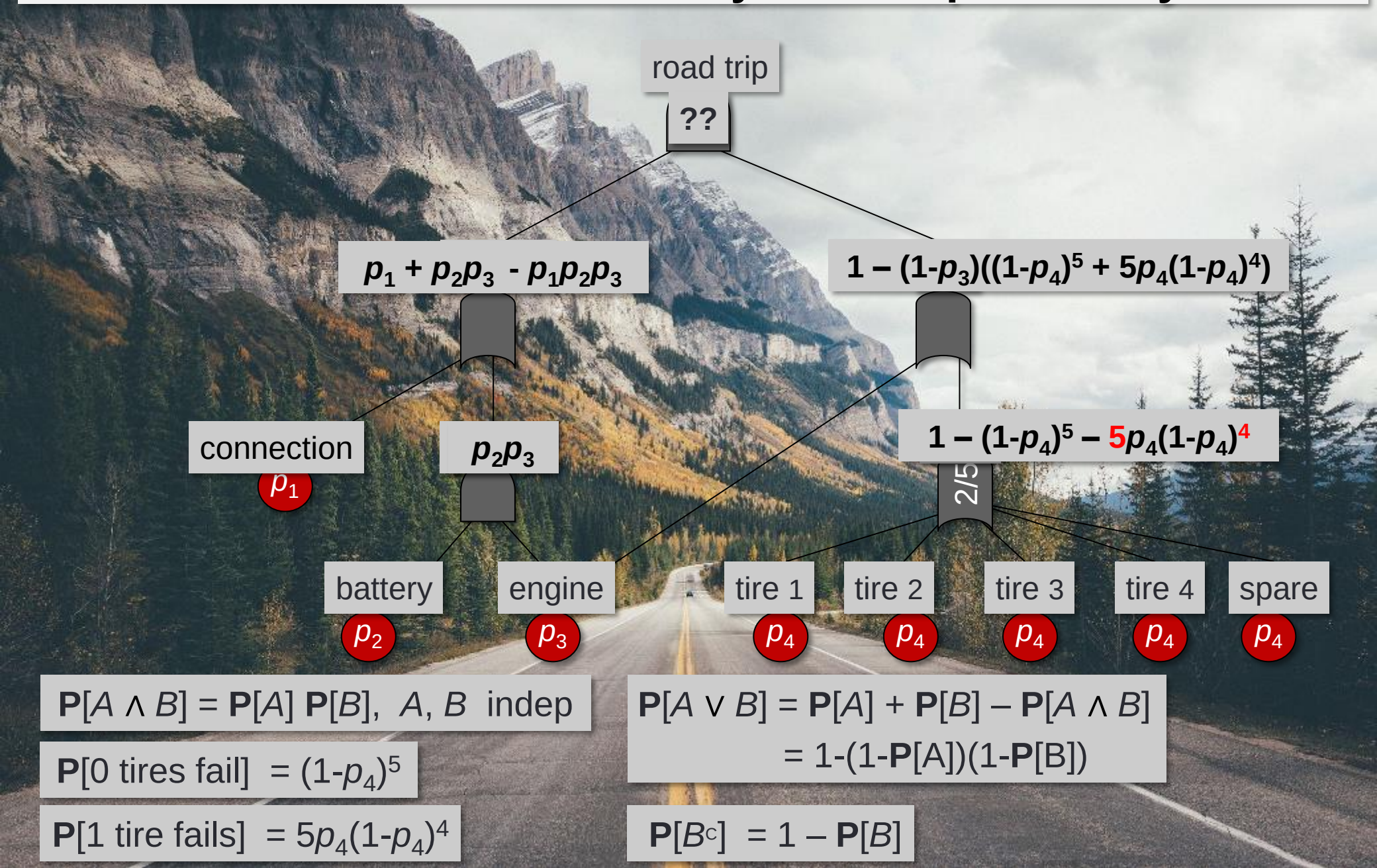
- If *E* is an AND-gate: $\Phi_F(\vec{e}, E) = 1$ iff $\Phi_F(\vec{e}, E') = 1$ for all children *E*' of *E*
- If *E* is an OR-gate: $\Phi_F(\vec{e}, E) = 1$ iff $\Phi_F(\vec{e}, E') = 1$ for some child *E*' of *E*
- …etc...

Unreliability / Failure probability

Suppose $p_i = \mathbf{P}(e_i = 1)$ are known and are ***independent***

Then $U(F) = \mathbf{P}(\Phi_F(\vec{e}, \text{top}) = 1)$.

Question: determine unreliability / failure probability



$\mathbf{P}[A \wedge B] = \mathbf{P}[A]\,\mathbf{P}[B]$, A, B indep

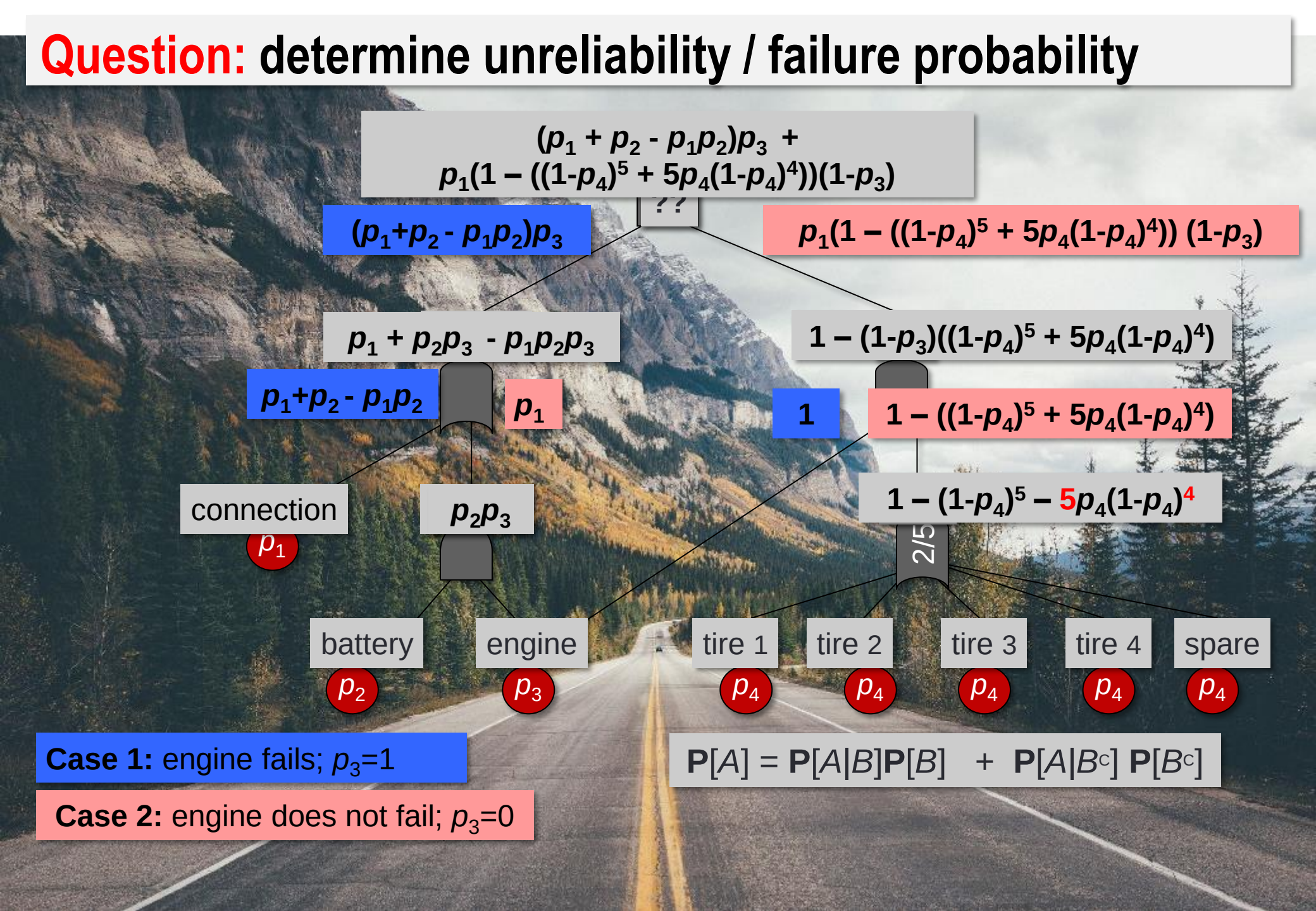
$\mathbf{P}[A \vee B] = \mathbf{P}[A] + \mathbf{P}[B] - \mathbf{P}[A \wedge B]$
$= 1-(1-\mathbf{P}[A])(1-\mathbf{P}[B])$

$\mathbf{P}[0 \text{ tires fail}] = (1-p_4)^5$

$\mathbf{P}[1 \text{ tire fails}] = 5p_4(1-p_4)^4$

$\mathbf{P}[B^c] = 1 - \mathbf{P}[B]$

Question: determine unreliability / failure probability



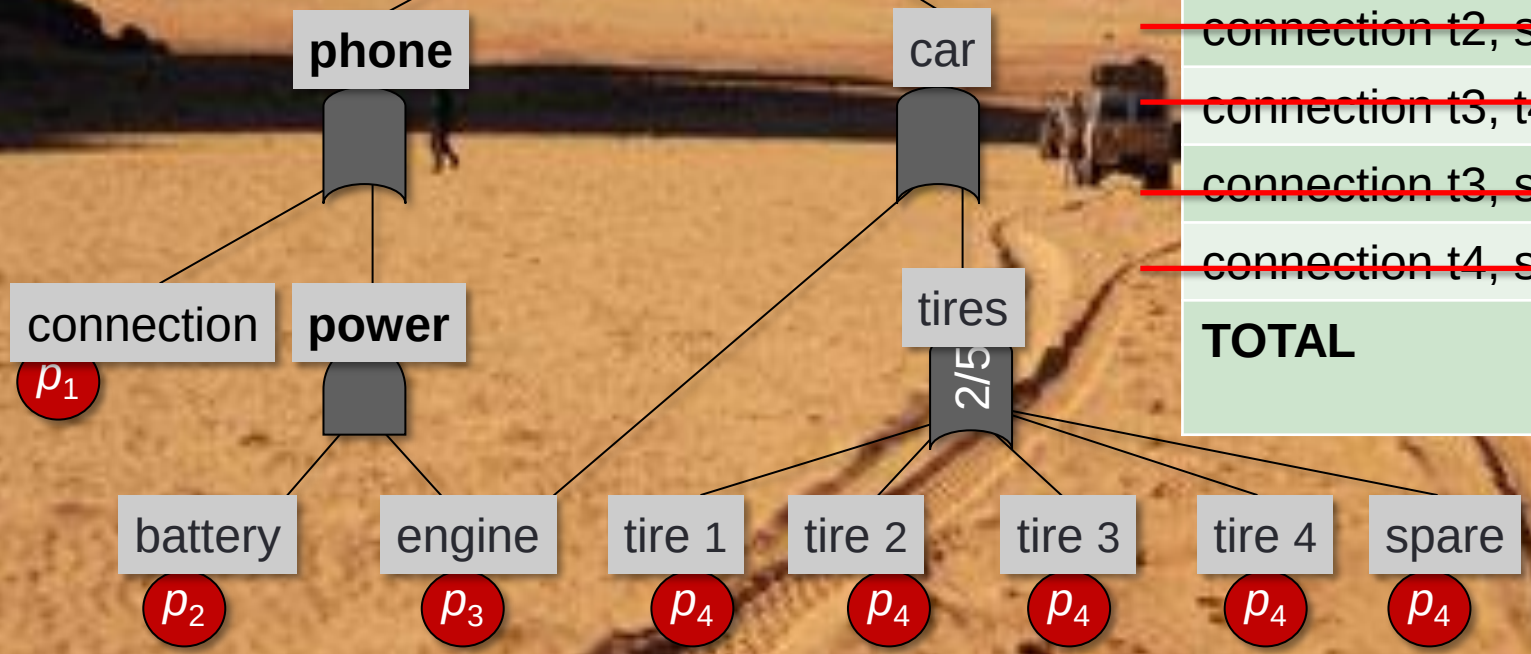
Simplify the computation

Find 3 ways to speed up this computation

1. Use minimal cut sets
2. Disregard extremely small probabilities
3. Use BDDs

Speed up: Using cut sets

$$(p_1 + p_2 - p_1p_2)p_3 + p_1(1 - ((1-p_4)^5 + 5p_4(1-p_4)^4)(1-p_3)$$



Min cut set	prob
connection, engine	$p_1 * p_3$
battery, engine	$p_2 * p_3$
~~connection, t1, t2~~	~~$p_1 * p_4^2$~~
~~connection, t1, t3~~	~~$p_1 * p_4^2$~~
~~connection t1, t4~~	~~$p_1 * p_4^2$~~
~~connection t1, spare~~	~~$p_1 * p_4^2$~~
~~connection t2, t3~~	~~$p_1 * p_4^2$~~
~~connection t2, t4~~	~~$p_1 * p_4^2$~~
~~connection t2, spare~~	~~$p_1 * p_4^2$~~
~~connection t3, t4~~	~~$p_1 * p_4^2$~~
~~connection t3, spare~~	~~$p_1 * p_4^2$~~
~~connection t4, spare~~	~~$p_1 * p_4^2$~~
TOTAL	$p_1*p_3 + p_2*p_3$ ~~$+ 10* p_1 * p_4^2$~~

Note: p_1 , p_4 should be low

Speed up 4: Using BDDs

Binary Decision Diagrams (BDDs)

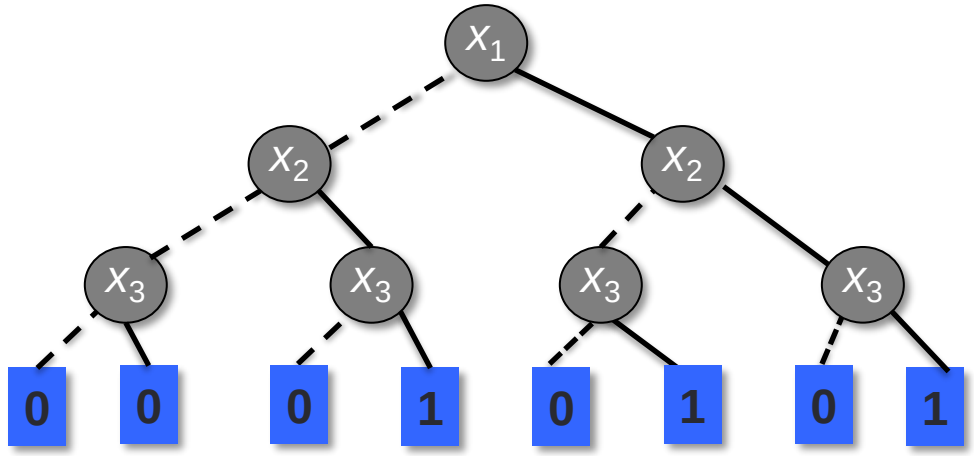
- Compact representation for Boolean functions $f(x_1,x_2,\ldots,x_n)$
  - e.g. the structure function of a FT
- Heavily used in model checking
  - state space
- **NOT** Behaviour-driven development! (testing)

Example: $(x_1 \vee x_2) \wedge x_3$

Truth Table

x_1	x_2	x_3	$f(x_1,x_2,x_3)$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

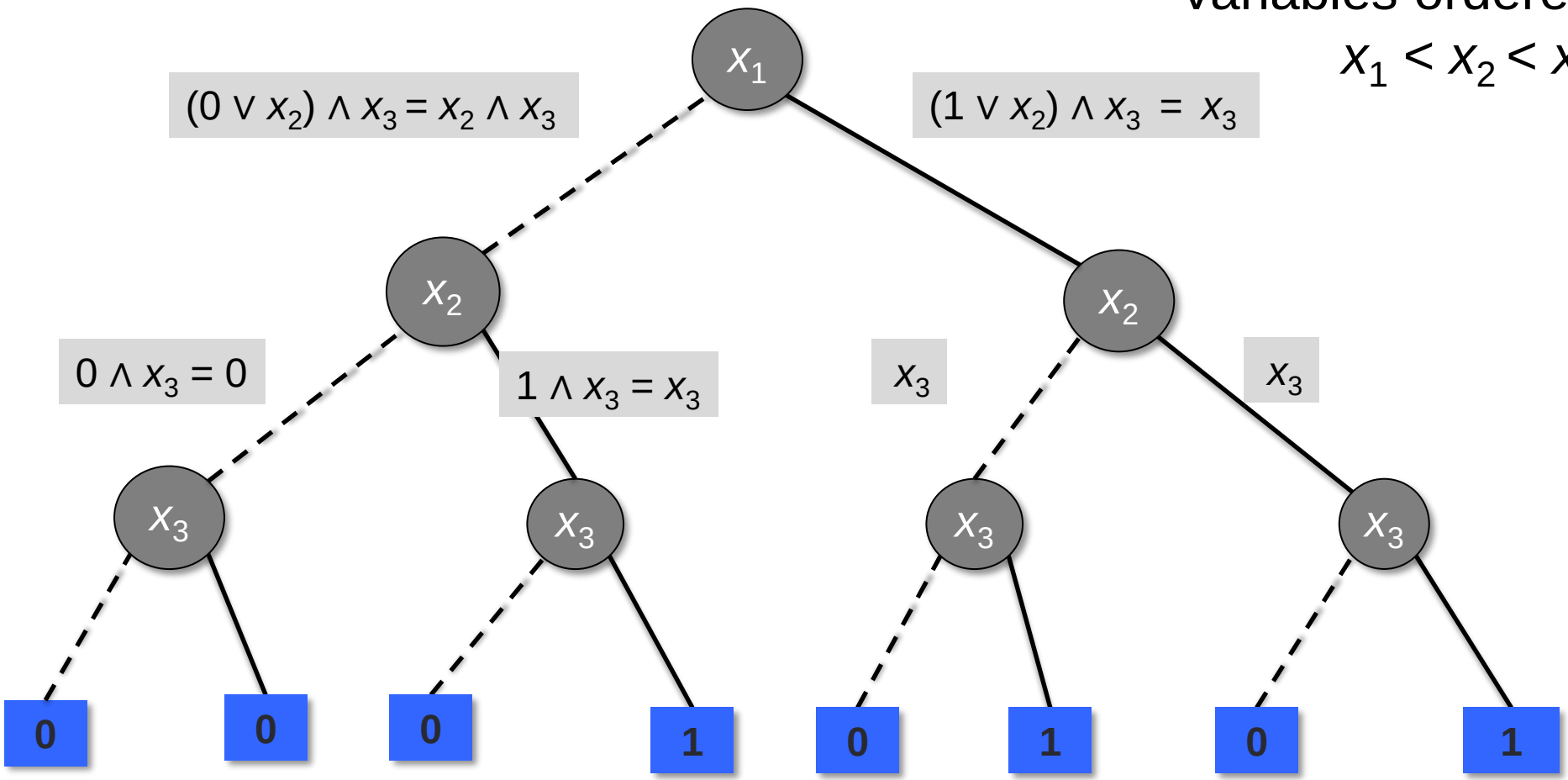
Decision Tree



- Vertex represents decision
- Follow dashed line for value 0
- Follow solid line for value 1
- Function value in leaf

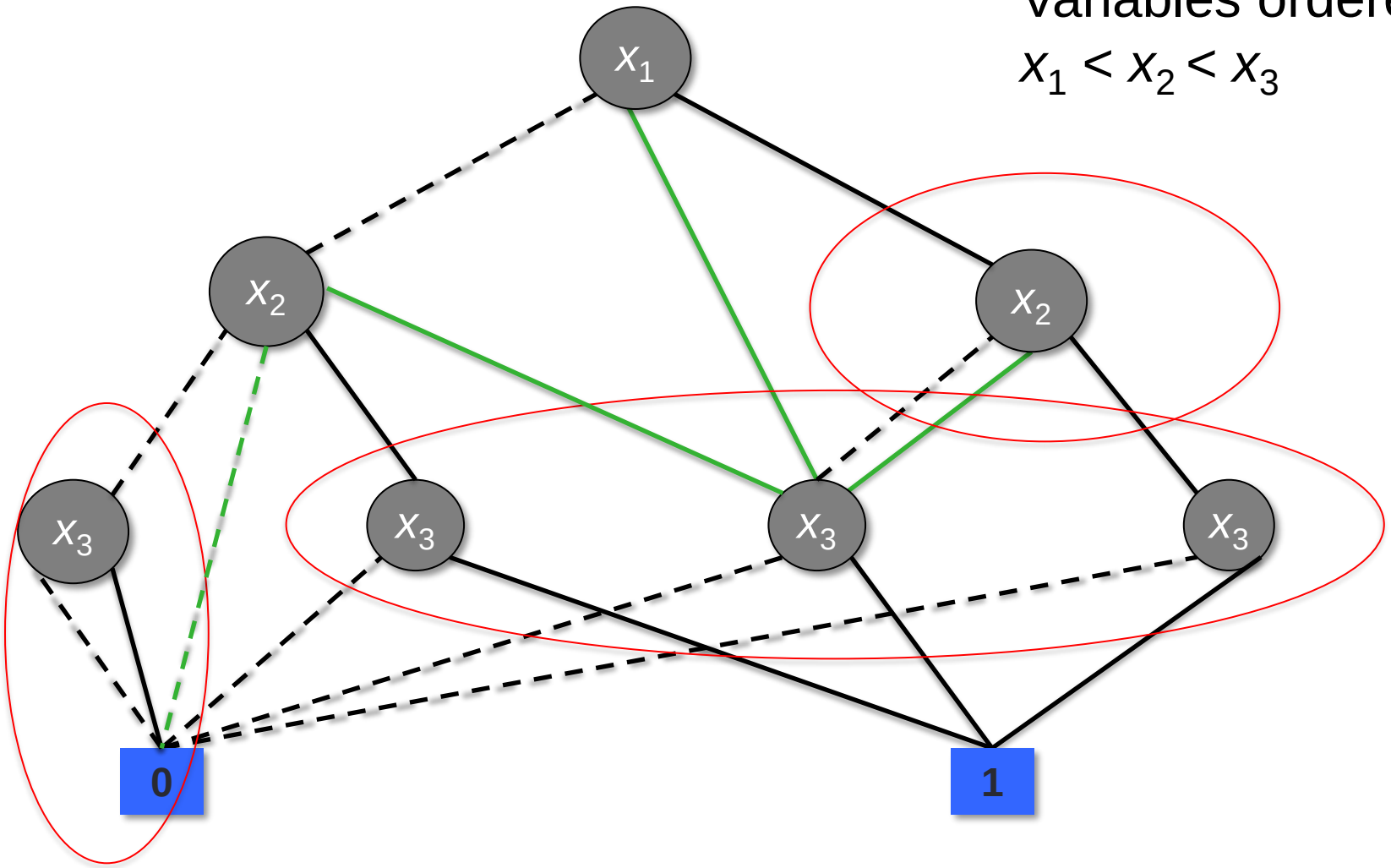
Deriving the BDD: $(x_1 \vee x_2) \wedge x_3$

Variables ordered
$x_1 < x_2 < x_3$

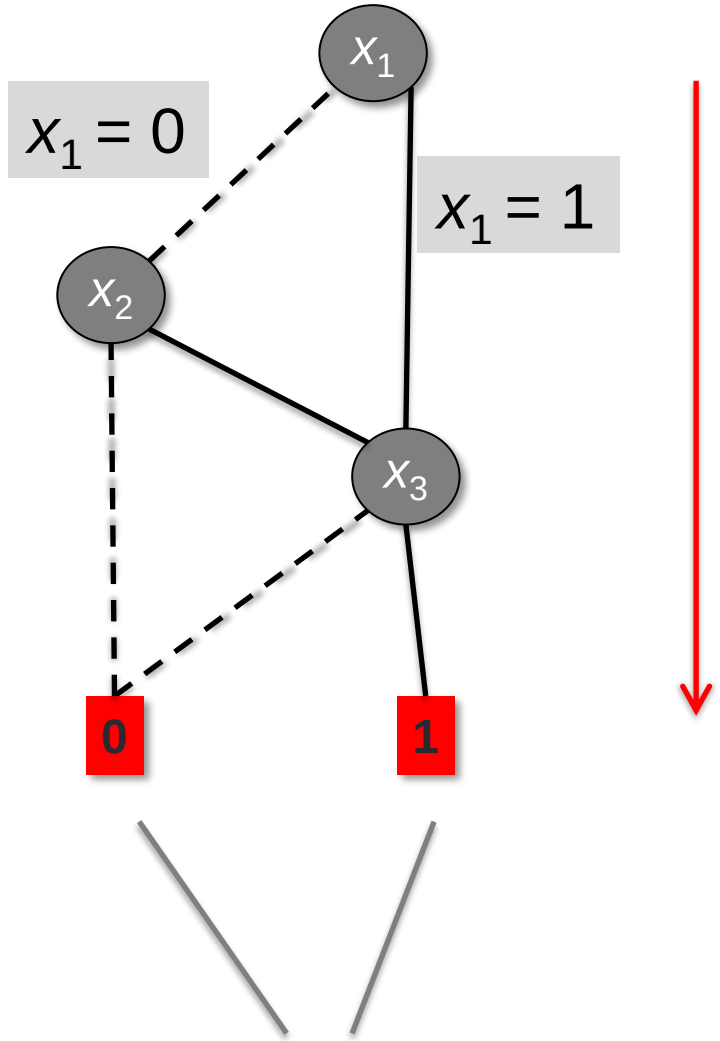


Reducing the BDD: $(x_1 \vee x_2) \wedge x_3$

Variables ordered
$x_1 < x_2 < x_3$



Example: $(x_1 \vee x_2) \wedge x_3$

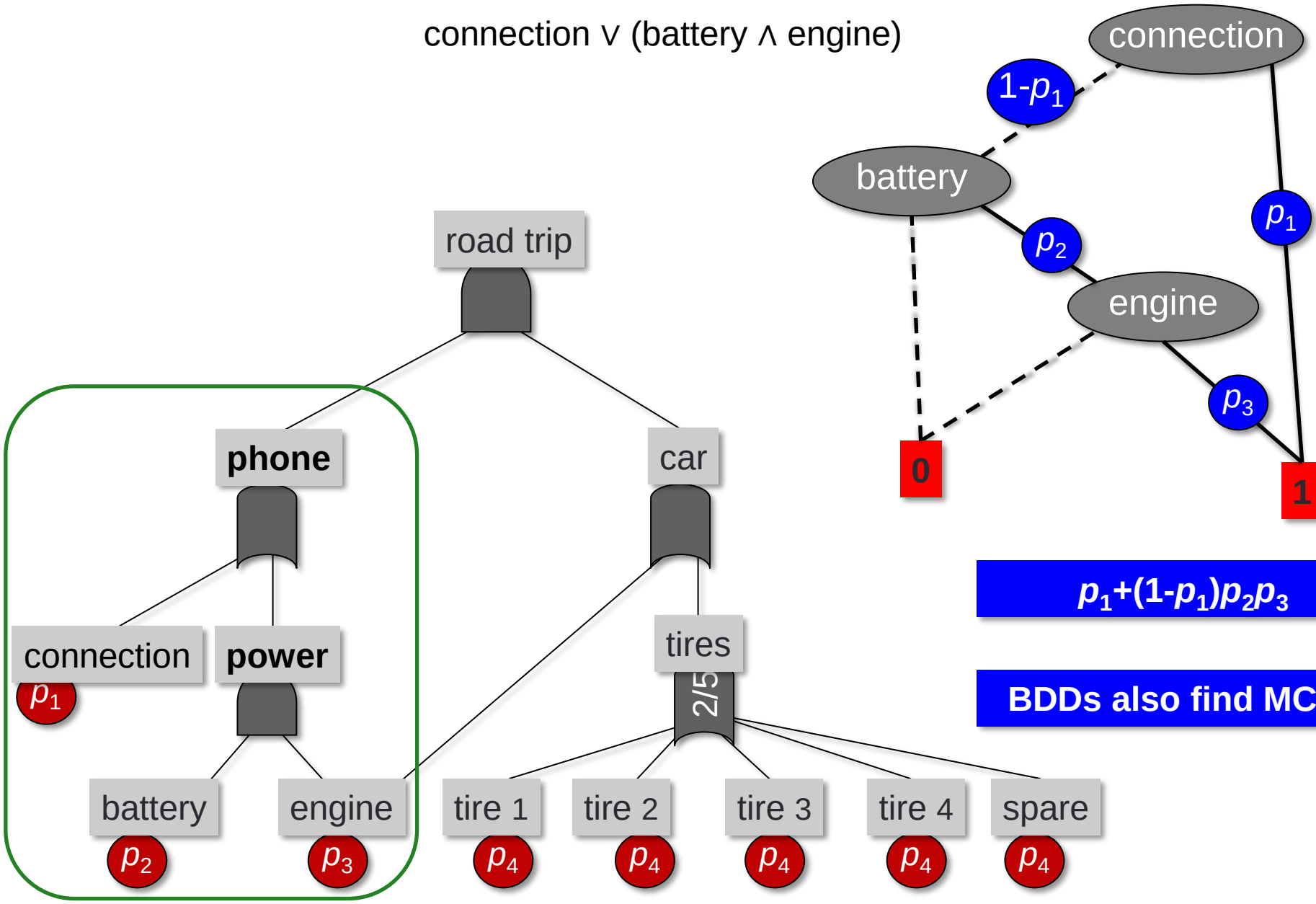


Variables ordered
$x_1 < x_2 < \ldots < x_n$

- Size of BDDs heavily depends on variable order $x_1 < x_2 < \ldots < x_n$
- Finding best order is NP-hard
- Good algos in practice

Using BDDs

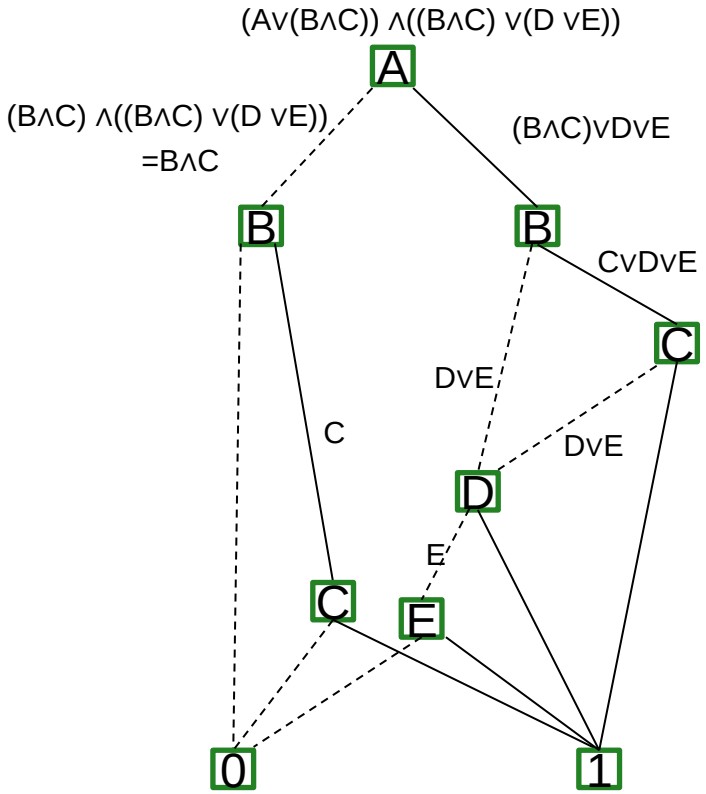
connection ∨ (battery ∧ engine)



$p_1+(1-p_1)p_2p_3$

BDDs also find MCSs

Reduced BDDs from scratch



Calculating unreliability/failure probability: summary

1. **Via MCS**
   - (slight) overapproximation
2. **Bottom-up**
   - Fast; only correct if FT is actually a tree
3. **Via BDDs**
   - Worst-case exponential size, ok in practice

Unreliability over time

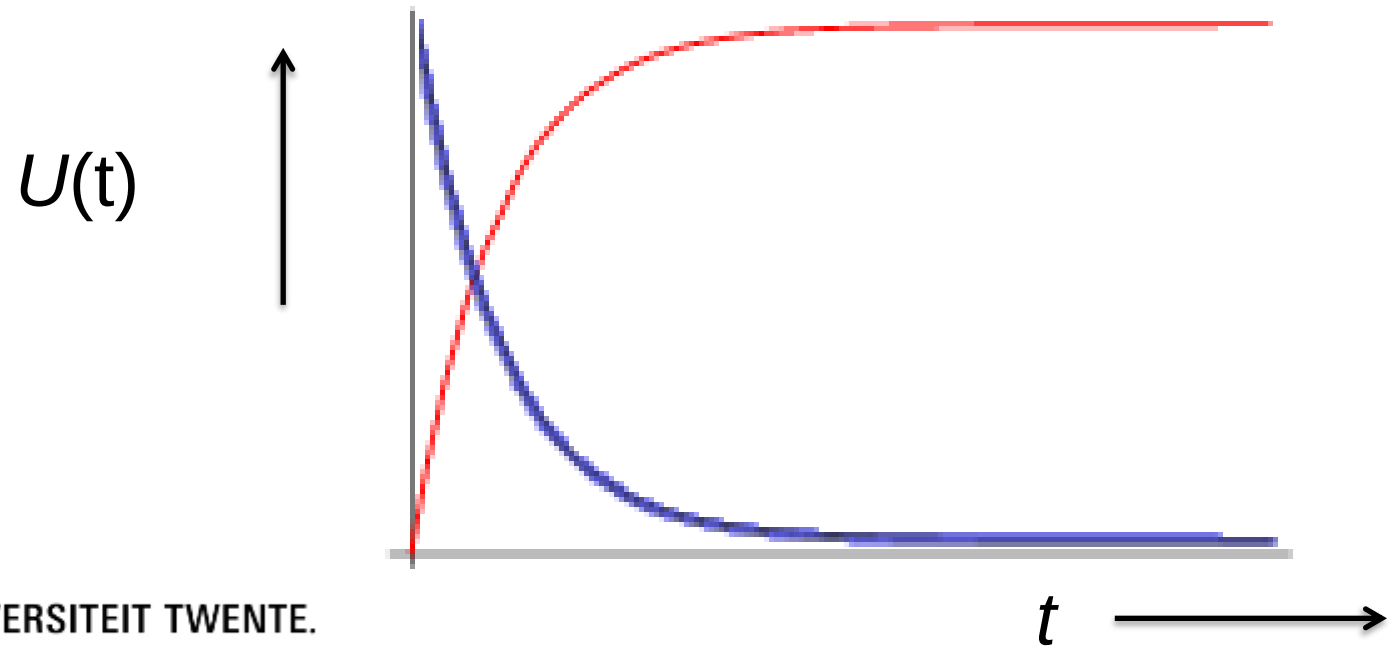
- Unreliability

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

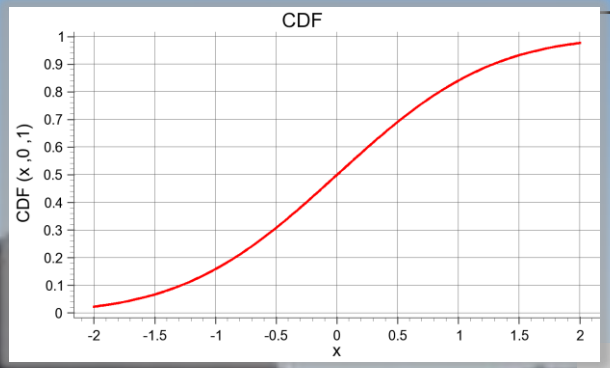
- Reliability / Survivor function

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

  = 1- $U(t)$



Continuous probability: failure behaviour over time



road trip

phone

car

connection

power

tires

2/5

battery

engine

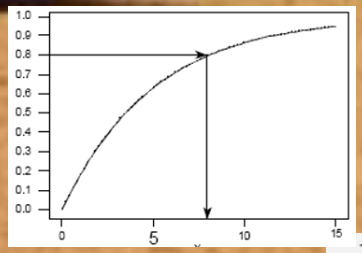
tire 1

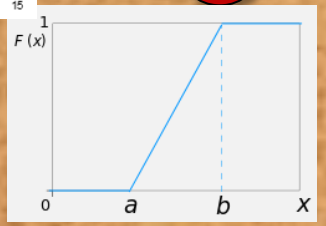
tire 2

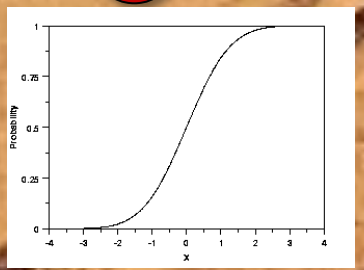
tire 3

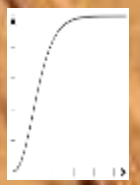
tire 4

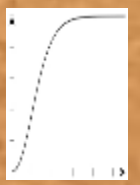
spare





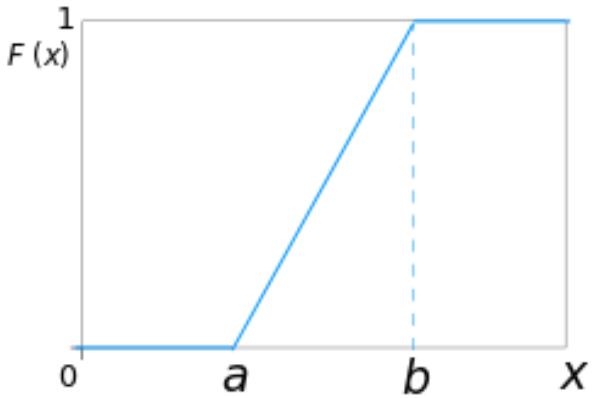


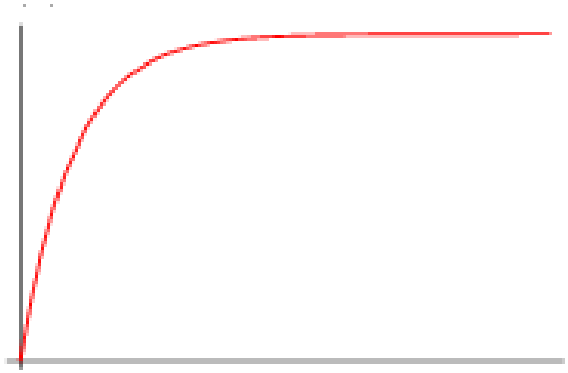




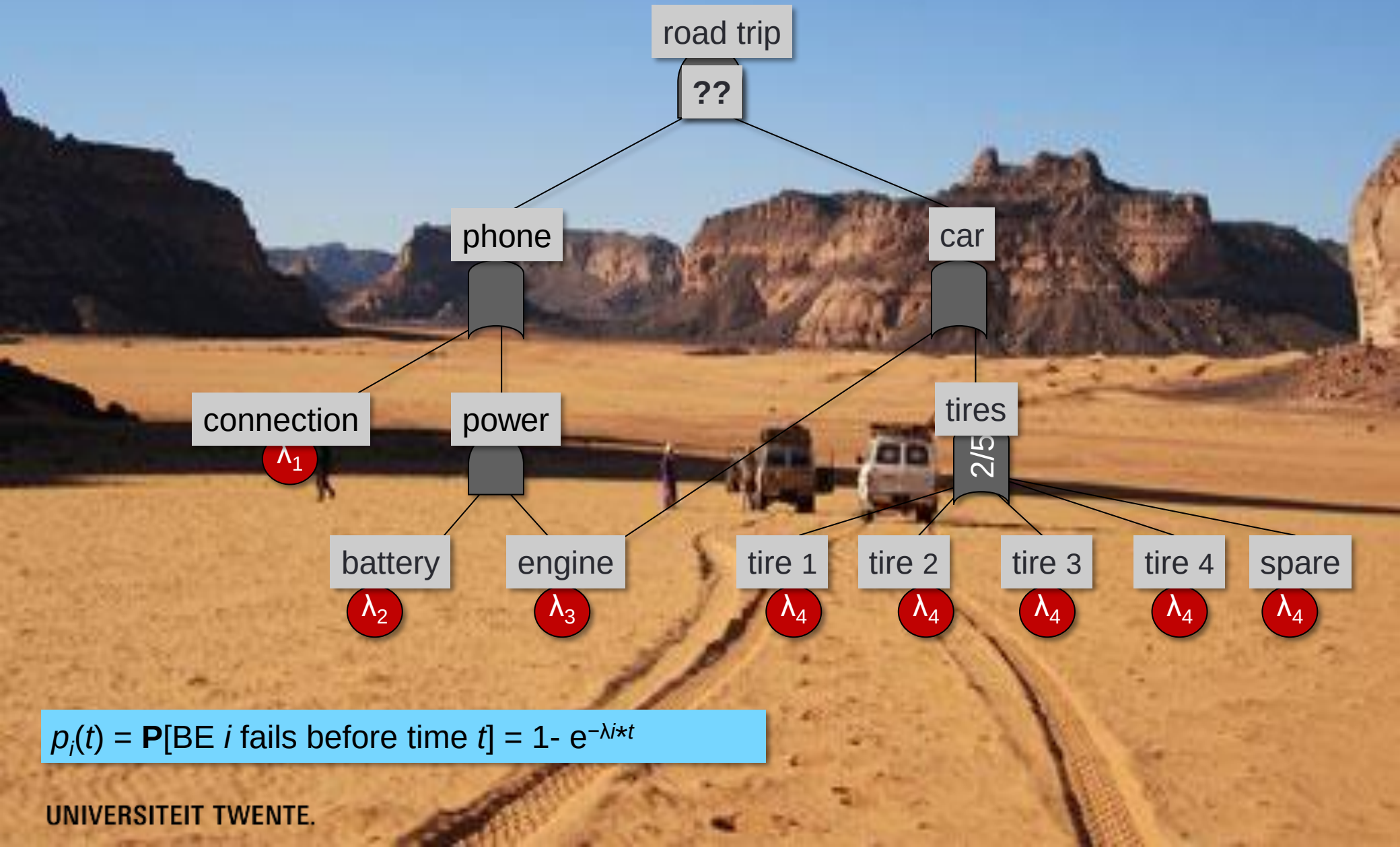
Continuous probability distributions: which one?

- **Uniform distribution**
- **Gaussian**
- **Exponential**
  - realistic model for degradation
  - mathematically tractable
  - approximation via composed exponentials
- **Weibull**
  - generalized exponential
  - often used
  - not discussed
- Use the cumulative density function:
  - $F(x) = \mathbf{P}[X \leq x]$



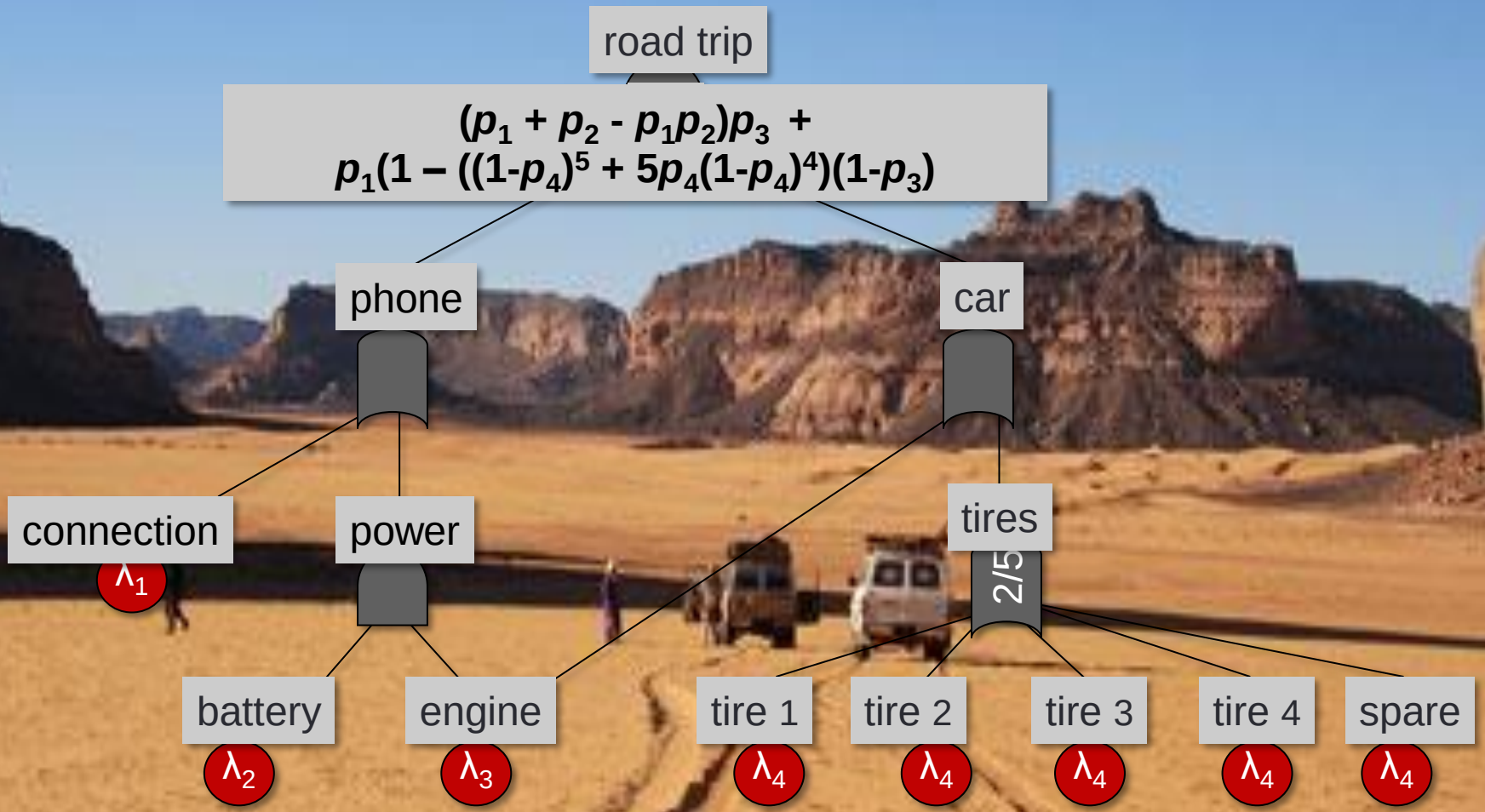


Example: probabilistic analysis



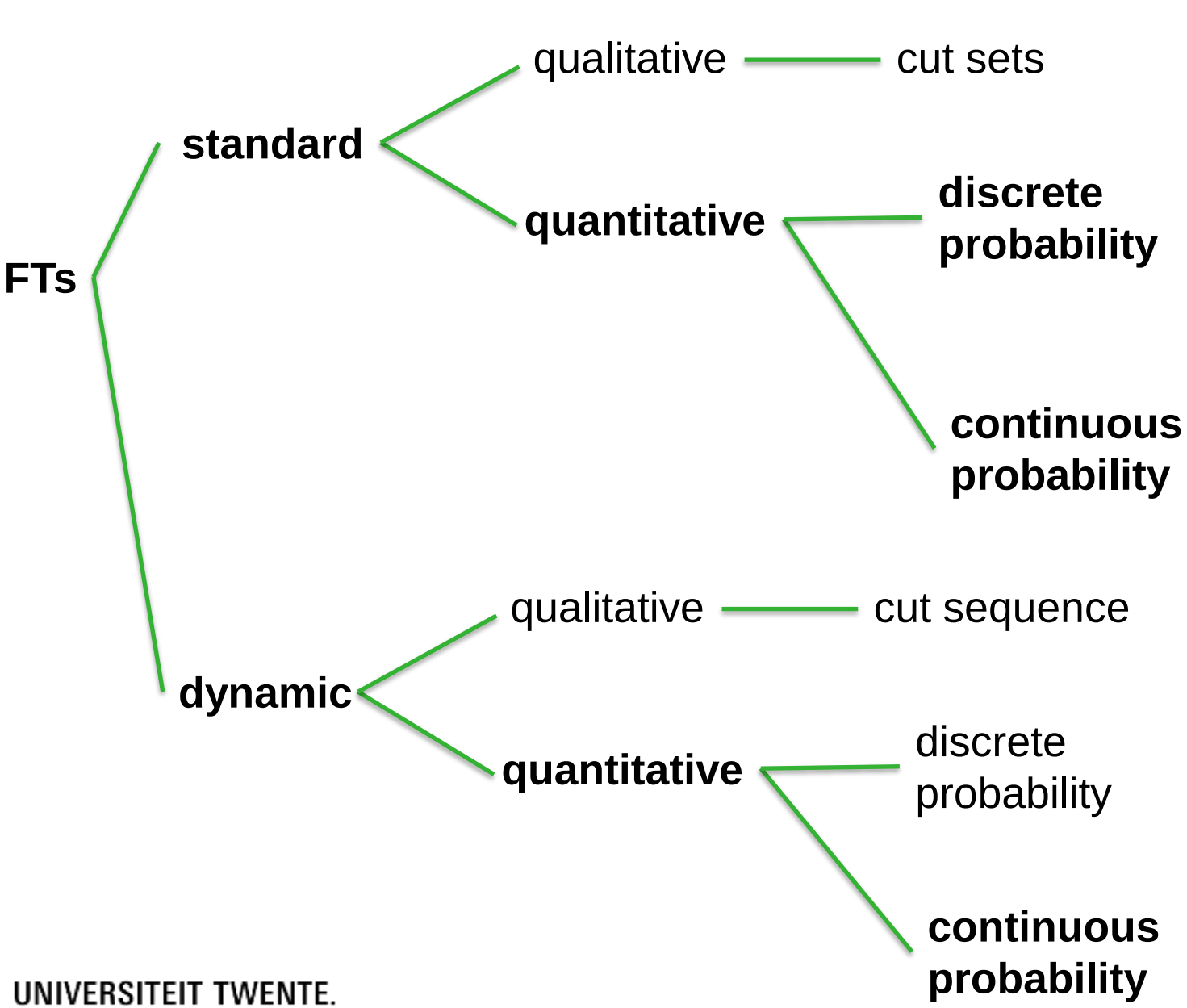
$p_i(t)$ = **P**[BE *i* fails before time *t*] = 1- $e^{-\lambda i * t}$

Example: probabilistic analysis



- $p_1(t)$ varies over time
- $p_1(t)$ probability of having failed at time t
- Substitute $p_1(t) = 1 - e^{-\lambda 1 t}$
- … works the same for other probability distributions

Overview



Technique
- Recursive
- BDDs

Technique
- Bottom up
- Cut sets
- BDDs

Technique
- The same!

Technique
- Model checking
- Next time

Next week: BREAK!

In 2 weeks: dynamic fault trees

- FTs with more complicated gate types
- More time-dependent behaviour
- Analysis using Markov chains

Tomorrow: tutorial

- Work on next homework set
- Work on additional exercises