Risk Analysis of Highly-integrated Systems

RA II: Methods (FTA, ETA)

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Fault Tree Analysis (FTA)

Problem description
It is not possible to analyse complicated, highly-reliable or novel systems as “black box”; i.e. there is a lack of knowledge at system level but predictions of failure probability, reliability and risk at system level are needed.

Approach: System decomposition
The behaviour of the overall system is determined by known behaviour as well as known logical and functional linking of system units.

Method of FTA

Starting point of FTA is a **predefined** system state (failed state as "top event"). The subsequent task is to find event combinations leading to the "top event". The branches are tracked top-down (top event -> intermediate events -> basic events); the reasoning is **deductive**.

Goals

• Systematic identification of failure modes (causes) and combinations as well as associated unit failures (basic events) leading to a "top event"
• Computation of "top event" probability where appropriate

Working steps of a FTA

• Definition of the "top event"
• Identification of all basic event combinations which result in the "top event"

If quantitative

• Assignment of failure probabilities to basic events
• Boolean modelling and calculations of probabilities
• Analysis of dominating failure combination and impacts (importance analysis), proposals for system improvement/optimisation

(1) Definition of the "top event"

In general: System Failure

In particular: Loss of specific functions and services meaning the failure of the overall system, (e.g. rupture of a gas storage tank).

(2) Identification of basic event combinations

The formal combination of events constitutes the logical structure of the system considered or the derived Boolean model (fault tree). The model consists of:

• Input events: lower event ("input" to the gate)

• Gates (logic operation): show the relationship of lower events needed to result in a higher event (logic AND, OR)

• Output events: higher event ("output" of the gate).

The behaviour of the gates is determined by the **Rules of Boolean Algebra**.

Boolean Algebra

Boolean Algebra is an algebraic system, in which the logical variables x, y, ... with the definition range (0, 1) can be linked with the following functions:

• Conjunction (AND, $\wedge$)
• Disjunction (OR, $\vee$)
• Negation (NOT, $\bar{\ }$)

$x \wedge y$

y \ x	0	1
0	0	0
1	0	1

$x \vee y$

x \ y	0	1
0	0	1
1	1	1

$\bar{x}$

x	0	1
$\bar{x}$	1	0

•These operations are defined by truth tables, describing the output depending on the different combinations of values for the variables.

•Time is not explicitly included in boolean algebra

Boolean Algebra

Statement	Description	Statement	Description
$X \cap Y = X \cap Y$ $X \cup Y = X \cup Y$	commutativity	$\bar{\bar{X}} = X$	
$X \cap (Y \cap Z) = (X \cap Y) \cap Z$ $X \cup (Y \cup Z) = (X \cup Y) \cup Z$	associativity	$\left(\overline{X \cap Y}\right) = \bar{X} \cup \bar{Y}$ $\left(\overline{X \cup Y}\right) = \bar{X} \cap \bar{Y}$	**de-Morgan Theorem**
$X \cap (Y \cup Z) = (X \cap Y) \cup (X \cap Z)$ $X \cup (Y \cap Z) = (X \cup Y) \cap (X \cup Z)$	distributivity	$O \cap X = O$ $O \cup X = X$	
$X \cap X = X$ $X \cup X = X$	**Idempotent**	$L \cap X = X$ $L \cup X = L$	
$X \cap (X \cup Y) = X$ $X \cup (X \cap Y) = X$	absorption	$X \cup \left(\bar{X} \cap Y\right) = X \cup Y$	
$X \cup \bar{X} = L$ $X \cap \bar{X} = O$		$X \cap \left(\bar{X} \cup Y\right) = X \cap Y$	

Requirements of the Boolean model

- The function of the system can be represented formaly by characterising the state of the components with two values (true, false)
- No time dependencies of the failure rate of the components (boolean models are static models)
- No repair
- Stochastic independent failure of components

Remark:
Often systems not fulfilling these requirements can be described with boolean algebra.
E.g. by additional information as „compnent fails within a time interval Δt".

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Boolean Function

Mapping *f* between a dependent variable y and independent boolean variables $x_0, x_1,, \ldots, x_{n-1}$

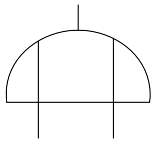
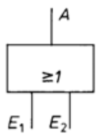
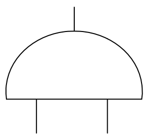
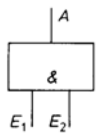
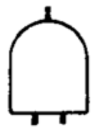
$$y = f(x_0, x_1,, \ldots, x_{n-1}) = f\left(\underline{x}\right) \qquad \forall \quad x_i = \begin{cases} 1 \\ 0 \end{cases}; \; y = \begin{cases} 1 \\ 0 \end{cases}$$

Example

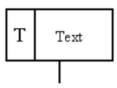
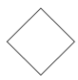
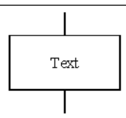
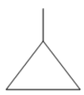
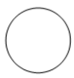
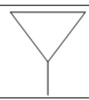
Exclusive-Or $\quad y = \left(x_0 \wedge \overline{x}_1\right) \vee \left(\overline{x}_0 \wedge x_1\right)$

Remark:
•In Boolean Algebra we mostly use the operators $\wedge$ and $\vee$ instead of the set operators $\cap$ and $\cup$.
•Often we do not use the AND operator, but note it as "·" ($X \wedge Y \equiv X \cdot Y$)

Logic Gate Symbols

Symbol	Alternative symbols		Description
			**OR-Gate** Output fault occurs if at least **one** of the input faults occurs
			**AND-Gate** Output fault occurs if **all** of the input faults occur

Primary event, intermediate event and transfer symbols

Symbol	Description	Symbol	Description
	TOP EVENT (failed system state)		UNDEVELOPED EVENT (of insufficient consequence or information is unavailable)
	INTERMEDIATE EVENT (fault event occurring because of antecedent causes)		TRANSFER IN (input from a further developed tree, e.g. on a different page)
	BASIC EVENT (basic initiating fault requiring no further development)		TRANSFER OUT (output that is further processed in an other tree)

Required information for a FTA

Component level:

• Different relevant failure modes of individual units (to fix most relevant one)

• Relevant external "influences", e.g. environmental impacts

• For quantitative analyses: failure probabilities

System level:

• Precise definition of the operation mode in question and the system boundaries

(3) Assignment of failure probabilities problems

• Lack of data (e.g. reliability figures of highly reliable tailor-made components in nuclear power plants, components designed to work under changing operating conditions in the chemical industry, etc.)

• Development of the database usually causes an extensive amount of work

(4) Boolean modelling and calculation of probabilities Boolean Model

Functional model, describing the interaction and dependencies in form of a boolean function with binary variables. Characterisation of the states of the system components.

In Reliability Analysis

• **System state:** Operational (working) or failed (not working)

• **Question:** Is the system operational or has it failed. Do we know the state of the components?

Summary of the assumptions/preconditions

- A technical system consists of units (components)
- The units are both technically and logically connected
- The state of each unit follows a binary logic (true/false, on/off, intact/defect)

Available logic operators are:

- conjunction: AND ($\wedge$, $\cap$)
- disjunction: OR ($\vee$, $\cup$)

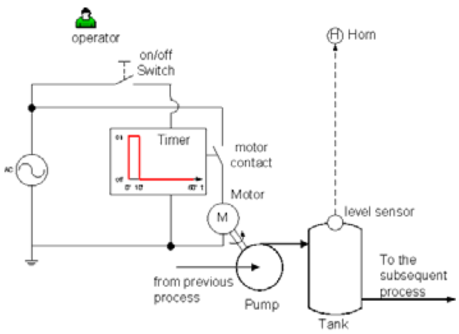
Labelling of the probabilities:

p_i: probability of survival of the *i*-th unit

q_i: probability of failure of the *i*-th unit

Example: Fault Tree of a Pumping System

In a pumping system a tank is filled with a fluid needed for a chemical reactor in 10 and emptied in 50 minutes; hence, a complete cycle takes 1 hour. After the on/off switch is turned on the motor contacts are turned on. The timer gives ‘on’ signal for 10’ and then ‘off’ signal for 50’. If this mechanism fails an alarm signal sounds and the operator turns the switches off to prevent a tank failure due to overfilling.



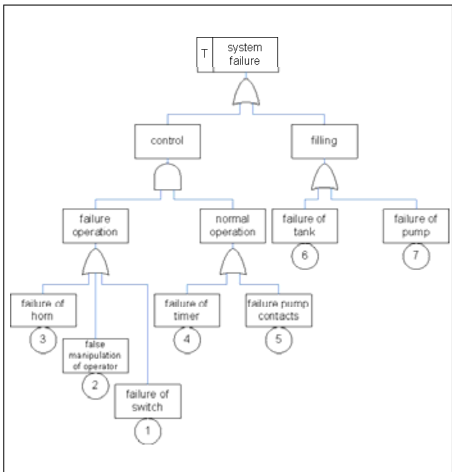
Examples of probabilities used in quantitative FTAs

Unit or functional components	Survival Probability p_i	Failure Probability q_i
Electromechanical parts: switches, timer, horn, contacts	0.9995	$5 \cdot 10^{-4}$
Passive element: storage tank	0.999999	10^{-6}
Active element: pump	0.9999	10^{-4}
„Functional element human being": operator	0.99973	$2.7 \cdot 10^{-4}$

$q_{operator}$: Probability of a wrong operator response on a perceived signal
q_{pump}: Probability of pump operation despite of being switched off

R(t) = 1 – F(t) => q_i = 1- p_i

Example from industry: Pumping-storage system



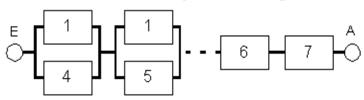
Boolean function: failure

$\bar{y} = \left[(\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3) \wedge (\bar{x}_4 \vee \bar{x}_5)\right] \vee (\bar{x}_6 \vee \bar{x}_7)$

multiply

$\bar{y} = \bar{x}_1\bar{x}_4 \vee \bar{x}_1\bar{x}_5 \vee \bar{x}_2\bar{x}_4 \vee \bar{x}_2\bar{x}_5 \vee \bar{x}_3\bar{x}_4 \vee \bar{x}_3\bar{x}_5 \vee \bar{x}_6 \vee \bar{x}_7$

This is a serial-parallel- and serial system:
Reliability block diagram



Computation

$$F_{SP} = 1 - \prod_{i=1}^{6}(1 - q_{1i}q_{2i}) = 1 - \left[(1 - q_1 q_4)(1 - q_1 q_5)....\text{etc}\right]$$

$$F_s = 1 - \left[(1 - q_6)(1 - q_7)\right]$$

$$F = F_{SP} + F_s = ... = 1.0227 \cdot 10^{-4}$$

Simplifications for simple systems only

Pr($A \cap B$) = P(A) · P(B)	Pr($A \cup B$) = Pr(A) + Pr(B) - Pr($A \cap B$)
	Approximation with small probabilities: Pr($A \cup B$) ≈ Pr(A) + Pr(B)

Note

For any number of random events A_i (i = 1, 2, ..., n), the equation after Poincaré is applied

$$\Pr\left(\bigcup_{i=1}^{n} A_i\right) = \sum_{i=}^{n} \Pr(A_i) - \sum_{\substack{i_1 i_2 = 1 \\ i_1 < i_2}}^{n} \Pr(A_{i1} \cap A_{i2}) + \sum_{\substack{i_1 i_2, i_3 = 1 \\ i_1 < i_2 < i_3}}^{n} \Pr(A_{i1} \cap A_{i2} \cap A_{i3}) + \ldots + (-1)^{n-1} \Pr(A_1 \cap A_2 \cap \ldots \cap A_n)$$

Rare event approximation for small *Pr(A$_i$)*

$$\sum_{i=}^{n} \Pr(A_i) - \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \Pr(A_i \cap A_j) \le \Pr\left(\bigcup_{i=1}^{n} A_i\right) \le \sum_{i=}^{n} \Pr(A_i)$$

Advantages of a FTA

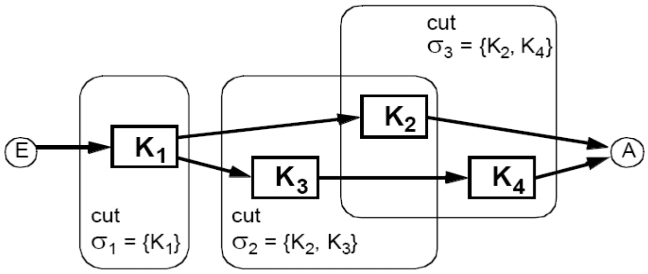
- Well suited for modelling of binary (Boolean) mechanical processes, e.g. valve fails to open/close
- Events occurring on component level due to interaction of multiple failures are easily representable
- Provides reliability figures of a system (if adequate data are available)
- Encourages a methodical way of thinking
- Applicable to a wide field of systems and processes.

Disadvantages

- Dynamic processes are not representable (a system is considered as "static")
- Complicated systems usually result in an unmanageable amount of basic events and combinations
- Reliability figures are often difficult to get.

Minimal Cut

A set of components cutting all paths from *E* to *A* in the reliability diagram, is called a **cut**. A cut is **minimal**, if it does not contain another cut as a subset.



The cuts σ_1, σ_2 and σ_3 are minimal.

Computation of highly complicated systems

Approach

Identification of those units which must at least operate for total system operability or units whose failure result in the total system failure.

→ Minimal Paths and Minimal Cut Sets respectively.

Notation

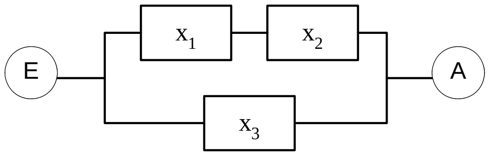
State x_i of unit *i, where*

$$x_i = \begin{cases} 0: \text{Unit state "failure" (in short} : \overline{x_i}) \\ 1: \text{Unit state "in operation" (in short} : x_i) \end{cases}$$

Negative Logic | Positive Logic

Minimal Cut Sets	Minimal Path Sets
Smallest set of failed units, which blocks the path from input to output in a reliability block diagram.	Smallest set of (operating) units, that leaves open a path from input to output in a reliability block diagram.

Example



Cuts σ_i: $\sigma_1=\{\bar{x}_1;\bar{x}_3\};\ \sigma_2=\{\bar{x}_2;\bar{x}_3\}$	Paths π_j: $\pi_1=\{x_1;x_2\};\ \pi_2=\{x_3\}$

Negative Logic | Positive Logic

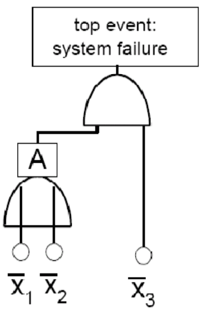
Negative Logic	Positive Logic
Each **cut set** ***i*** consists of the intersection of the minimum number of failed units required to cause the system failure, i.e. $\sigma_i=\bigcap_{k=1}^{l}\bar{x}_k$	Each **path set** ***j*** consists of the intersection of the minimum number of operating units required to ensure system operation, i.e. $\pi_j=\bigcap_{m=1}^{r}x_m$
System failure:: union of cut sets σ_i $\bar{y}=\bigcup_{i=1}^{n}\sigma_i$	**System operation**: union of paths π_j $y=\bigcup_{j=1}^{s}\pi_j$
Boolean algebra: De Morgan's Theoreme	
$\bar{y}=1-\bigcap_{j=1}^{n}(1-\sigma_j)=1-\left[(1-\bar{x}_1\bar{x}_3)(1-\bar{x}_2\bar{x}_3)\right]$	$\bar{y}=\bigcap_{j=1}^{s}(1-\pi_j)=(1-x_1x_2)(1-x_3)$
multiply, Idempotent law ...	
$\bar{y}=1-\left[(1-\bar{x}_1\bar{x}_3)(1-\bar{x}_2\bar{x}_3)\right]$ $=1-(1-\bar{x}_1\bar{x}_3-\bar{x}_2\bar{x}_3+\bar{x}_1\bar{x}_3\bar{x}_2\bar{x}_3)$ $=\bar{x}_1\bar{x}_3+\bar{x}_2\bar{x}_3-\bar{x}_1\bar{x}_2\bar{x}_3$	$\bar{y}=1-x_1x_2-x_3+x_1x_2x_3$

Negative Logic / Positive Logic

Negative Logic	Positive Logic
Cuts σ_i: $\sigma_1=\{\bar{x}_1;\bar{x}_3\}$; $\sigma_2=\{\bar{x}_2;\bar{x}_3\}$	Paths π_j: $\pi_1=\{x_1;x_2\}$; $\pi_2=\{x_3\}$
Boolean functions	
$\bar{y}=1-\bigcap_{j=1}^{n}(1-\sigma_j)=1-[(1-\bar{x}_1\bar{x}_3)(1-\bar{x}_2\bar{x}_3)]$	$\bar{y}=\bigcap_{j=1}^{s}(1-\pi_j)=(1-x_1x_2)(1-x_3)$
multiply, Idempotent law ...	
$\bar{y}=1-[(1-\bar{x}_1\bar{x}_3)(1-\bar{x}_2\bar{x}_3)]$ $=1-(1-\bar{x}_1\bar{x}_3-\bar{x}_2\bar{x}_3+\bar{x}_1\bar{x}_3\bar{x}_2\bar{x}_3)$ $=\bar{x}_1\bar{x}_3+\bar{x}_2\bar{x}_3-\bar{x}_1\bar{x}_2\bar{x}_3$	$\bar{y}=1-x_1x_2-x_3+x_1x_2x_3$
...	*Note:* Calculations in order to get the same formal representation as for cut sets. $\bar{y}=1-(1-\bar{x}_1)(1-\bar{x}_2)-(1-\bar{x}_3)+(1-\bar{x}_1)(1-\bar{x}_2)(1-\bar{x}_3)$ $=$...multiply... $=\bar{x}_1\bar{x}_3+\bar{x}_2\bar{x}_3-\bar{x}_1\bar{x}_2\bar{x}_3$
System failure probability	
$F=q_1q_3+q_2q_3-q_1q_2q_3$	$F=q_1q_3+q_2q_3-q_1q_2q_3$

Correlation between Fault Tree und minimal cut sets

A minimum cut set is defined as the smallest combination of failures which, if they all occur, will cause the top event to occur. Therefore, minimal cut sets can be evolved from a Fault Tree.

Fault Tree	Algorithm
	We start with the top event gate inputs and substitute and expand until the minimum cut set expression for the top event is obtained. AND gate inputs are listed in a row. Each input of an OR gate results in an additional row, whereby basic events remain. • Row: Idempotent law • Column: Absorption law $Z1 \vee (Z1 \cdot Z2) = Z1$ Example Step 1: a row (because of the AND gate) $\{A,\ \bar{x}_3\}$ Step 2: add a row (because of two OR gate inputs) and substitute A $\{\bar{x}_1;\bar{x}_3\}$ $\{\bar{x}_2;\bar{x}_3\}$ q.e.d.

Event Tree Analysis (ETA)

Purpose of an ETA

Organise, characterise, and quantify potential accident sequences that result from a single initiating event in a methodical manner:
•Graphical representation of logical and physical interactions of events in a system
•Inductive determination of final system states caused by defined causes
•Calculation of the resulting system state frequencies
⇨ Human behaviour, physical/chemical events and other Boolean events can be modelled by event trees.

Working steps of a quantitative ETA

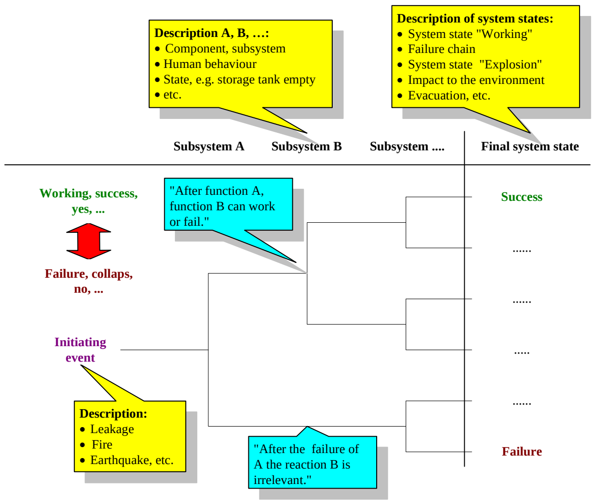
- List all possible initiating events
- Identify functional system responses, which evolve of a function/no function (i.e., Boolean) answer of a subsystem
- Grouping initiating events with all system responses
- Identification of event chains: Each system response has a corresponding branch that indicates weather or not it was successful. At the end of each sequence is an indication of the consequences than can be expected.
- Assigning of event frequencies P_u for the initiating event and the probabilities for success / failure
- Event frequencies calculation of the final system state

Event tree calculation

- The sum of all *n* “chain-probabilities” is equal to the probability of the initiating event
- The event probability of chain A with n subsequent subsystems is calculated like

$$\Pr(Chain\ A) = \Pr(Initiating\ event) \cdot \prod_{j=1}^{n} \Pr(Subsystem\ \ j)$$

Construction of an event tree



Advantages of an ETA

- Procedural steps easy to handle
- Applicable to all kind of (technical) systems, specially for larger facilities with active and passive security measures and unknown physical/chemical system states
- Scenarios and event sequences are listed and analysed
- Combination of function and failure
- Simplified visualisation of dynamic processes (domino effects in event sequences)

Disadvantages

- Difficulty in application: practical knowledge and a detailed system analysis needed
- Reduced readability of large event trees
- Even large event trees may contain errors
- Modifications of an event tree (by inserting a new subsystem) are difficult