How do we use these representations ?

Build a fault tree

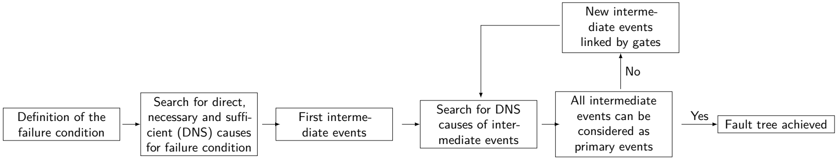


FIGURE – Fault tree construction process

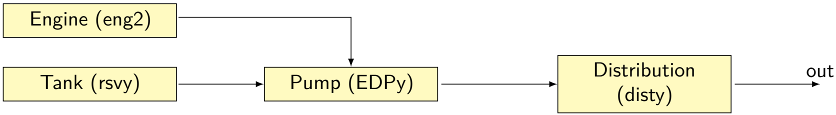


FIGURE – Yellow hydraulic system

Build a fault tree

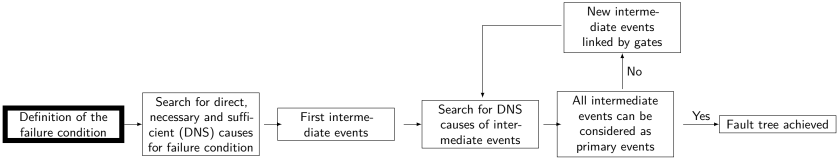


Figure – Fault tree construction process

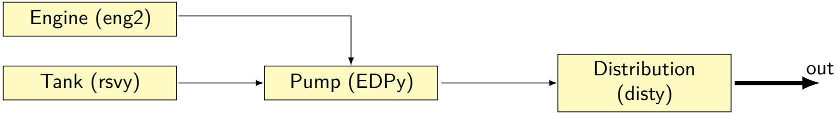


Figure – Yellow hydraulic system

Build a fault tree

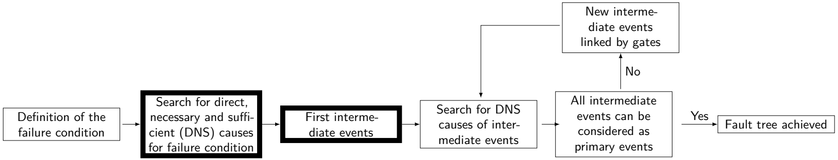


FIGURE – Fault tree construction process

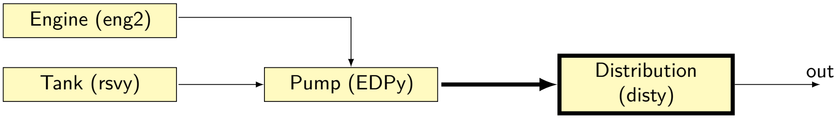


FIGURE – Yellow hydraulic system

Build a fault tree

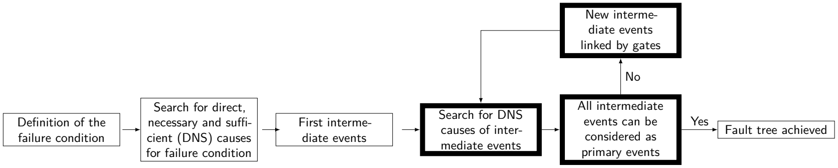


FIGURE – Fault tree construction process

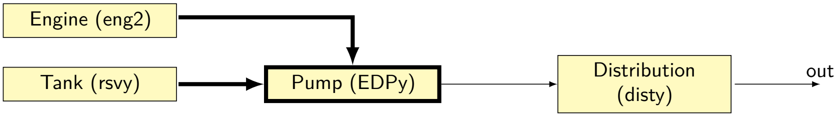


FIGURE – Yellow hydraulic system

Build a fault tree

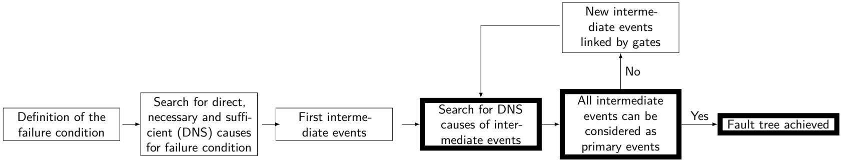


FIGURE – Fault tree construction process

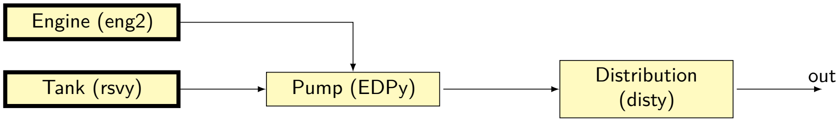
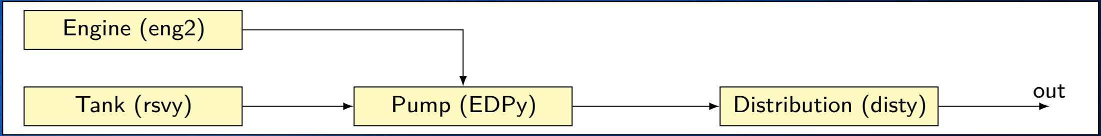


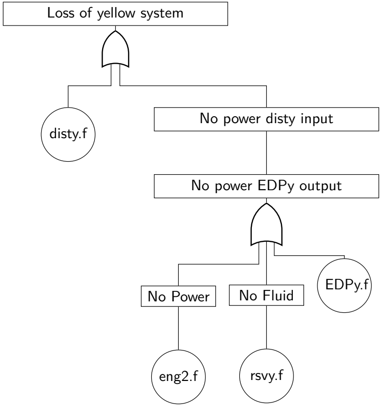
FIGURE – Yellow hydraulic system

According to the previous slides, build the fault tree of

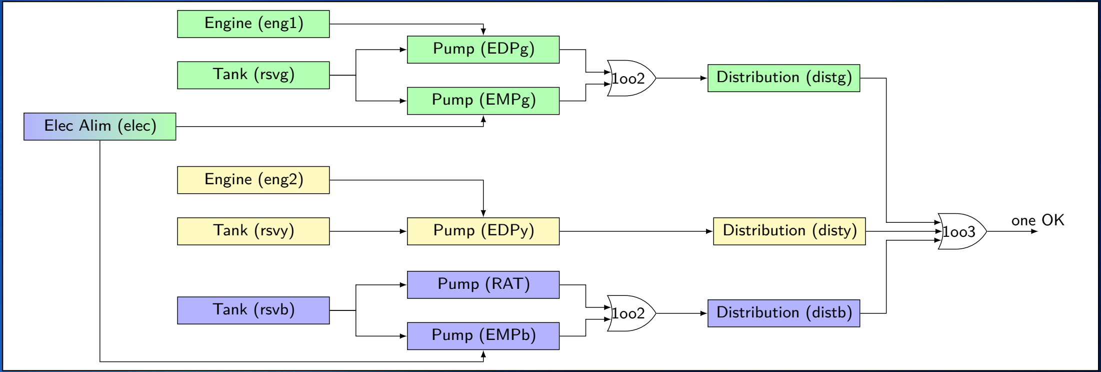
Loss of the yellow system



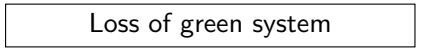
Solution


Loss of yellow system
disty.f
No power disty input
No power EDPy output
No Power
No Fluid
EDPy.f
eng2.f
rsvy.f

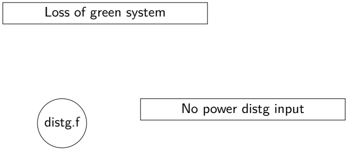
Try to build the fault tree of
Loss of the green system



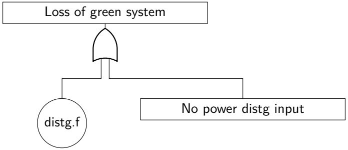
Solution



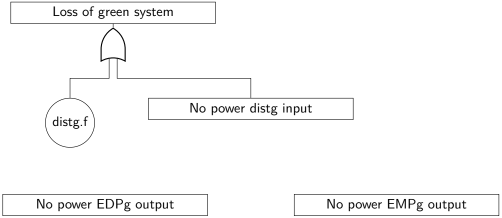
Solution



Solution

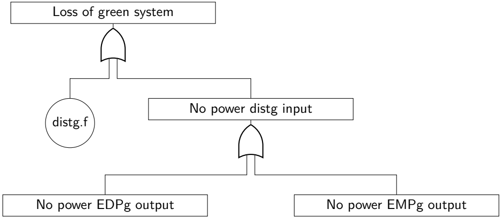


Solution



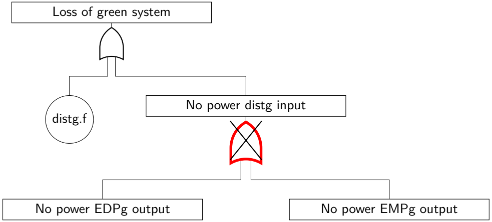
Loss of green system
distg.f
No power distg input
No power EDPg output
No power EMPg output

Solution

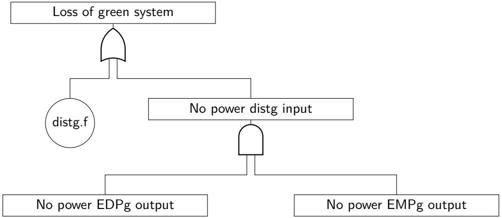


Loss of green system
distg.f
No power distg input
No power EDPg output
No power EMPg output

Solution

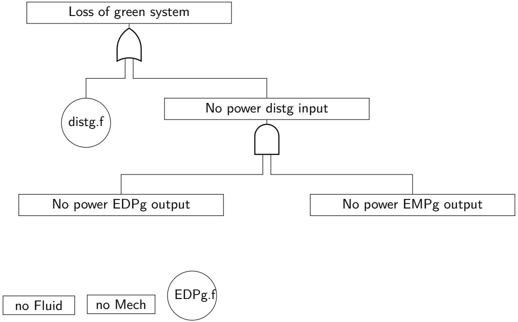

Loss of green system
distg.f
No power distg input
No power EDPg output
No power EMPg output

Solution

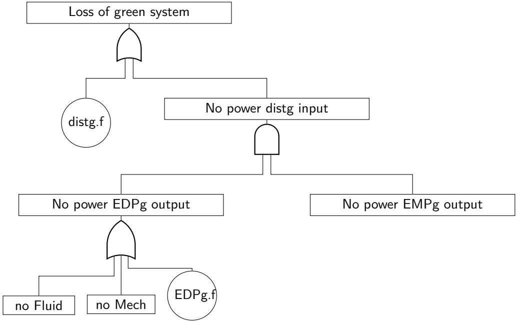


Loss of green system
distg.f
No power distg input
No power EDPg output
No power EMPg output

Solution

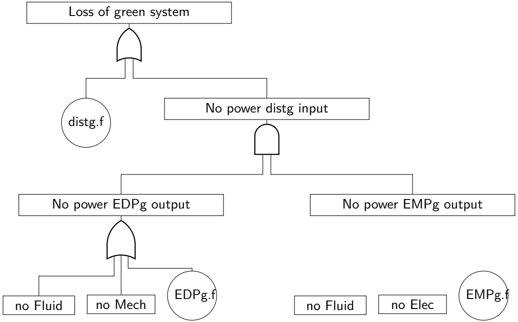

Loss of green system
distg.f
No power distg input
No power EDPg output
No power EMPg output
no Fluid
no Mech
EDPg.f

Solution



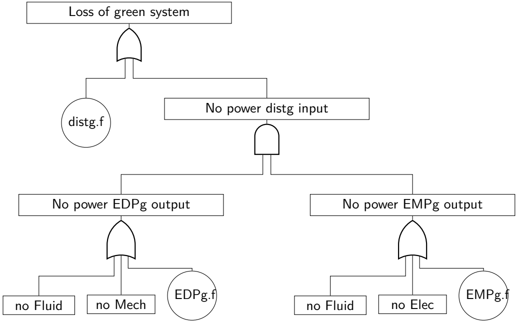
Loss of green system
distg.f
No power distg input
No power EDPg output
No power EMPg output
no Fluid
no Mech
EDPg.f

Solution



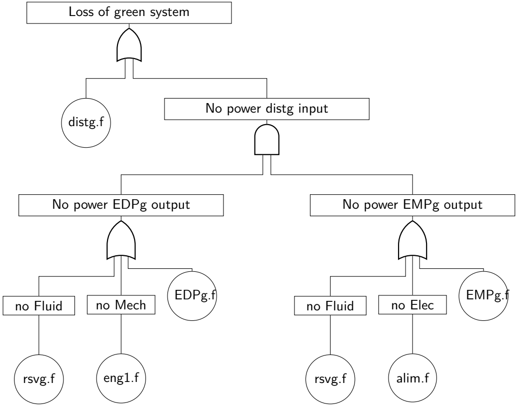
Loss of green system
distg.f
No power distg input
No power EDPg output
No power EMPg output
no Fluid
no Mech
EDPg.f
no Fluid
no Elec
EMPg.f

Solution



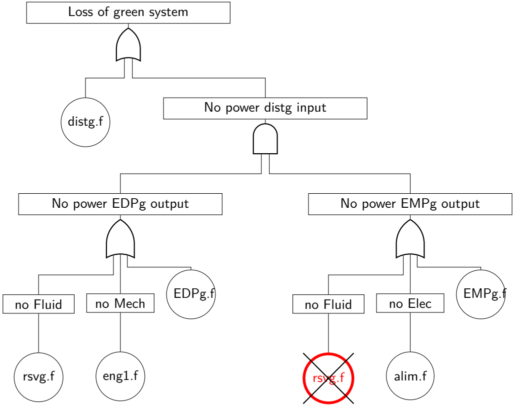
Loss of green system
distg.f
No power distg input
No power EDPg output
No power EMPg output
no Fluid
no Mech
EDPg.f
no Fluid
no Elec
EMPg.f

Solution



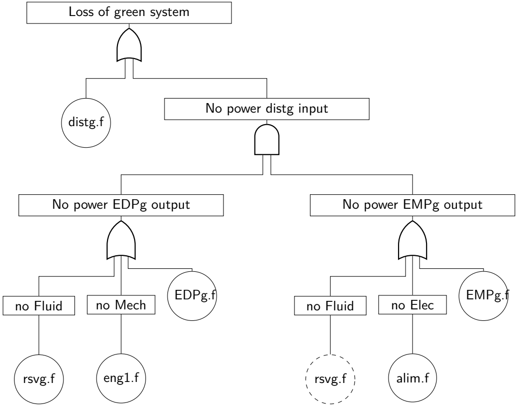
Loss of green system
distg.f
No power distg input
No power EDPg output
No power EMPg output
no Fluid
no Mech
EDPg.f
no Fluid
no Elec
EMPg.f
rsvg.f
eng1.f
rsvg.f
alim.f

Solution



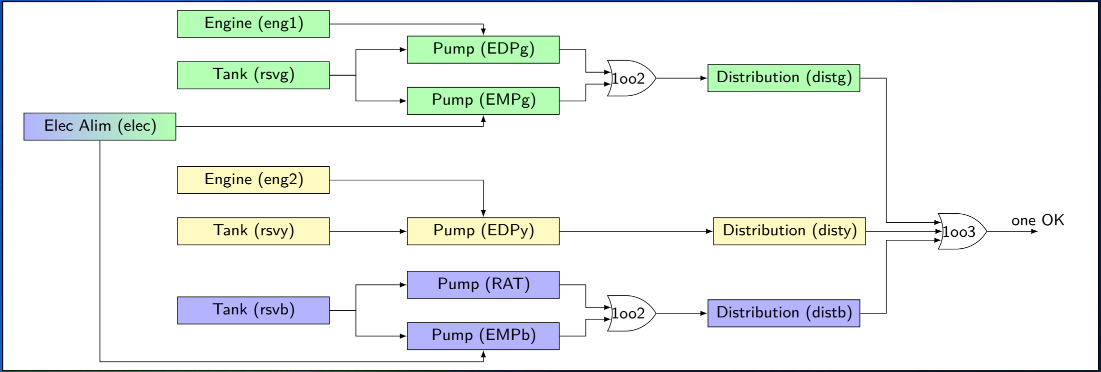
Loss of green system
distg.f
No power distg input
No power EDPg output
No power EMPg output
no Fluid
no Mech
EDPg.f
no Fluid
no Elec
EMPg.f
rsvg.f
eng1.f
rsvg.f
alim.f

Solution

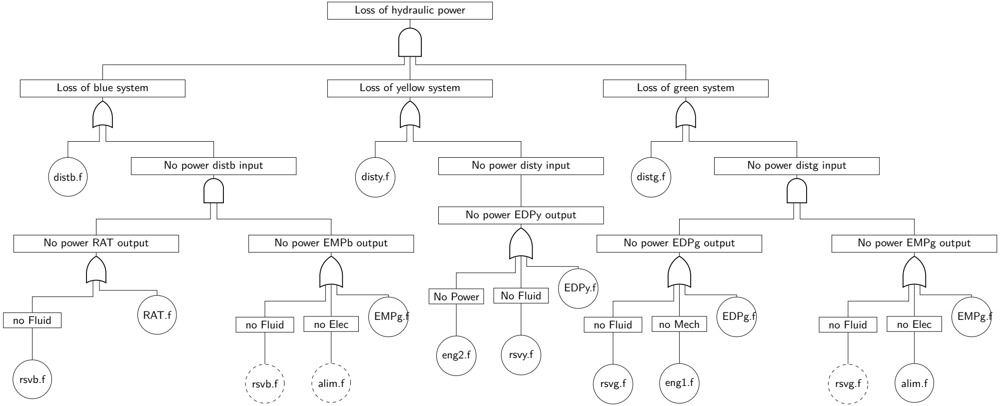


Loss of green system
distg.f
No power distg input
No power EDPg output
No power EMPg output
no Fluid
no Mech
EDPg.f
no Fluid
no Elec
EMPg.f
rsvg.f
eng1.f
rsvg.f
alim.f

Try to build the fault tree of
Loss of hydraulic power



Solution



ONERA THE FRENCH AEROSPACE LAB

Now a recap!

Today's lesson in 30"

- Dependability ⇒ ability to avoid unacceptable failures
- Acceptability defined by regulatory texts
- Dependability integrated trough safety process ⇒ What should we do and when
  - Assess system failures & criticality ⇒ FHA
  - Analyse contribution of system's components failures to system failure ⇒ PSSA (FTA, ...)
  - Quantify dependability with safety indicators $(R, \cdots)$

You understand highlighted terms
⇒ congratulations you've got the idea
Otherwise check out the slides !

How to perform safety assessment out of fault trees ?

Why using propositional logic in safety ?

To find the failure combinations leading to failure conditions

Is propositional logic expressive enough ?

Yes because fault trees are meant to model static systems : failure state does not depend on the order of occurrence of failures

Otherwise $\Rightarrow$ class on dynamic system modeling

How to define a logic ?

Syntax

- Does the sentence belong to the language ?
  Does $a \looparrowright b$ belong to propositional logic ?
- Notions : propositions, connectors, formulae

Semantics

- What is the meaning of the sentence ?
  if *b* ***and*** *c* ***then*** *a* ***and*** *b* ***or not*** *a* ***and*** *c* is *always true* ?
- Notions : formulae valuations, validity, logical consequence

Example of logic Propositional logic, First-order logic, Temporal logic

What can we write ?

$$
\begin{array}{lll}
\varphi ::= & proposition & \text{basic observations (ex :eng1.f)} \\
& \mid \textbf{not } \varphi & \text{negation (ex :}\textbf{not}\text{ eng1.f)} \\
& \mid \varphi_1 \textbf{ and } \varphi_2 & conjunction(ex: eng1.f \textbf{ and } eng2.f) \\
& \mid \varphi_1 \textbf{ or } \varphi_2 & disjunction(ex: eng1.f \textbf{ or } eng2.f) \\
& \mid \textbf{if } \varphi_1 \textbf{ then } \varphi_2 & implication(ex: \textbf{if } rsvg.f \textbf{ then } green.f) \\
& \mid \varphi_1 = \varphi_2 & equivalence(ex: rsvg.f = green.f) \\
& \mid (\varphi) & parenthesis(ex: (eng1.f))
\end{array}
$$

formulae sentences built using φ rule

literal $proposition \mid \textbf{not } proposition$

What does that mean ?

Define a valuation function $[\![\varphi]\!] \rightarrow \{\mathbf{T}, \mathbf{F}\}$

$$
\begin{array}{lcl}
[\![proposition]\!] & = & v \in \{\mathbf{T}, \mathbf{F}\} \\
 & & ex : [\![eng1.f]\!] = \mathbf{T} \text{ means "eng1 is lost" is true} \\
[\![\textbf{not } \varphi]\!] & = & \mathbf{T}\ iff\ [\![\varphi]\!]\ is\ \mathbf{F} \\
[\![\varphi_1 \textbf{ and } \varphi_2]\!] & = & \mathbf{T}\ iff\ [\![\varphi_1]\!]\ is\ \mathbf{T}\ and\ [\![\varphi_2]\!]\ is\ \mathbf{T} \\
[\![\varphi_1 \textbf{ or } \varphi_2]\!] & = & \mathbf{T}\ iff\ [\![\varphi_1]\!]\ is\ \mathbf{T}\ or\ [\![\varphi_2]\!]\ is\ \mathbf{T} \\
[\![\textbf{if } \varphi_1 \textbf{ then } \varphi_2]\!] & = & \mathbf{T}\ iff\ [\![\varphi_1]\!]\ is\ \mathbf{F}\ or\ [\![\varphi_2]\!]\ is\ \mathbf{T} \\
[\![\varphi_1 = \varphi_2]\!] & = & \mathbf{T}\ iff\ [\![\varphi_1\ ]\!] and\ [\![\varphi_2]\!]\ are\ both\ \mathbf{T}\ or\ both\ \mathbf{F} \\
[\![(\varphi)]\!] & = & [\![\varphi]\!]
\end{array}
$$

Satisfiability

Satisfiability

A formula φ is satisfiable iff it exists one valuation V of its propositions such that $[\![\varphi]\!]_V = \mathbf{T}$

Satisfiability

Let $\varphi =$ eng1.f **and not** eng2.f

$\Rightarrow$ for $V = \{[\![eng1.f]\!] = \mathbf{T}, [\![eng2.f]\!] = \mathbf{F}\}$ we have $[\![\varphi]\!]_V = \mathbf{T}$

$\Rightarrow$ φ is satisfiable

Logical consequence

Logical consequence

A formula φ_2 is a logical consequence of φ_1 iff for all valuation V such that $[\![\varphi_1]\!]_V = \mathbf{T}$ we have $[\![\varphi_2]\!]_V = \mathbf{T}$

Logical consequence

Let $\varphi_2 =$ eng1.f and φ_1 =eng1.f **and not** eng2.f.

- $V = \{[\![eng1.f]\!] = \mathbf{T}, [\![eng2.f]\!] = \mathbf{F}\}$ is the only valuation satisfying φ_1
- $[\![\varphi_2]\!]_V = \mathbf{T}$

$\Rightarrow$ φ_2 is a logical consequence of φ_1

Implicant

Product

A product is a set of literals that does not contain both a variable and its negation.

Product

$\{eng1.f,$ **not** $eng2.f\}$ is a product

Implicant

A product P is an implicant of formula φ iff φ is a logical consequence of P.

Implicant

$\{eng1.f,$ **not** $eng2.f\}$ is an implicant of *eng1.f* ***and not*** *eng2.f*

Prime implicant

Prime implicant

An implicant P of φ is a prime implicant if there is no implicant P' of φ such that P' is strictly included into P.

Prime implicant

$\{eng1.f,$**not** $eng2.f\}$ is a prime implicant of *eng1.f* ***and not*** *eng2.f*

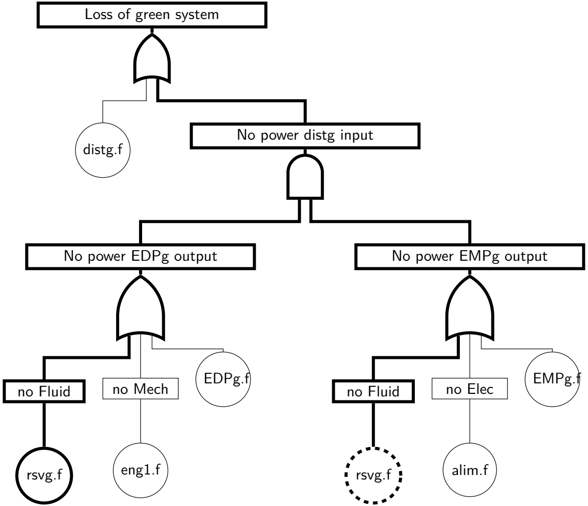
Safety Assessment

Fault tree ⇔ formula φ describing the failure combinations leading to a failure condition

- accident can occur ⇔ φ satisfiable
- situations where accident occurs ⇔ implicants of φ
- causes of the accident ⇔ prime implicants of φ

1. Is Loss of the green system possible ?
2. If yes, find a combination of failures where Loss of the green system occurs ?
3. Is your combination minimal ?
4. If possible, find prime implicants of size two, three.

Satisfiability & Implicant



Can we compute automatically satisfiability and prime implicants of φ

Shannon Decomposition

ite operator

$$\mathbf{ite}(v, \varphi_1, \varphi_2) = \mathbf{if}\ v\ \mathbf{then}\ \varphi_1\ \mathbf{else}\ \varphi_2$$

partial valuation $\varphi|_{v=x}$ is the formula φ where all occurrences of the proposition v are replaced by the value $x \in \{\mathbf{T}, \mathbf{F}\}$.

Shannon Decomposition

Let φ be a formula containing a proposition v then the Shannon decomposition on v is :

$$\mathbf{ite}(v, \varphi|_{v=\mathbf{T}}, \varphi|_{v=\mathbf{F}})$$

Shannon decomposition is applied recursively on the proposition contained in φ

Shannon Decomposition

Let $\varphi =$ eng1.f **and not** eng2.f, the step of the decomposition are :

1. Decompose on eng1.f :
   $\varphi|_{eng1.f=\mathbf{T}} = \textbf{not } eng2.f$
   $\varphi|_{eng1.f=\mathbf{F}} = \mathbf{F}$, so
   $\varphi = \textbf{ite}(eng1.f, \textbf{not } eng2.f, \mathbf{F})$
2. Decompose on eng2.f :
   $\textbf{not } eng2.f|_{eng2.f=\mathbf{T}} = \mathbf{F}$
   $\textbf{not } eng2.f|_{eng2.f=\mathbf{F}} = \mathbf{T}$,
   and **F** does not depend on $eng2.f$, so
   $\varphi = \textbf{ite}(eng1.f, \textbf{ite}(eng2.f, \mathbf{F}, \mathbf{T}), \mathbf{F})$

Binary Decision Diagram (BDD)

What's that ?

BDD

A BDD is a directed, oriented and acyclic graph encoding a formula φ. BDD contains :

- decision nodes labelled by a proposition v own exactly two sons, the low son (resp high son) accessed through "0"(resp "1") edge is the root of the BDD encoding $\varphi|_{v=\mathbf{F}}$ (resp. $\varphi|_{v=\mathbf{T}}$)
- terminal 1 (resp. 0) encoding the formula $\mathbf{T}$ (resp. $\mathbf{F}$)

Binary Decision Diagram (BDD)

$$\varphi = disty.f \textbf{ or } (EDPy.f \textbf{ or } eng2.f \textbf{ or } rsvgy.f)$$

$$\Downarrow Shannon\ decomposition$$

$\textbf{ite}(disty.f, \textbf{T},$ $)$

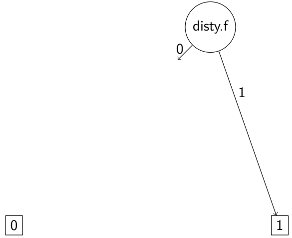


FIGURE – BDD of the loss of yellow system

Binary Decision Diagram (BDD)

$$\varphi = disty.f \textbf{ or } (EDPy.f \textbf{ or } eng2.f \textbf{ or } rsvgy.f)$$

$$\Downarrow Shannon\ decomposition$$

$$\textbf{ite}(disty.f, \mathbf{T}, \textbf{ite}(EDPy.f, \mathbf{T}, \qquad\qquad))$$

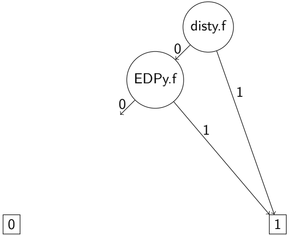


FIGURE – BDD of the loss of yellow system

Binary Decision Diagram (BDD)

$$\varphi = disty.f \textbf{ or } (EDPy.f \textbf{ or } eng2.f \textbf{ or } rsvgy.f)$$

$$\Downarrow Shannon\ decomposition$$

$$\textbf{ite}(disty.f, \mathbf{T}, \textbf{ite}(EDPy.f, \mathbf{T}, \textbf{ite}(rsvy.f, \mathbf{T}, \qquad)))$$

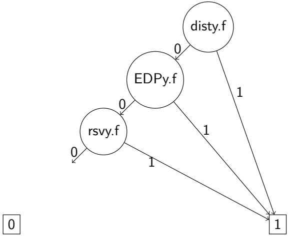


FIGURE – BDD of the loss of yellow system

Binary Decision Diagram (BDD)

$$\varphi = disty.f \textbf{ or } (EDPy.f \textbf{ or } eng2.f \textbf{ or } rsvgy.f)$$

$$\Downarrow Shannon\ decomposition$$

$$\textbf{ite}(disty.f, \textbf{T}, \textbf{ite}(EDPy.f, \textbf{T}, \textbf{ite}(rsvy.f, \textbf{T}, eng2.f)))$$

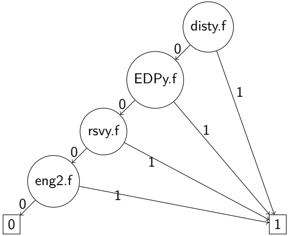


FIGURE – BDD of the loss of yellow system

Binary Decision Diagram (BDD)

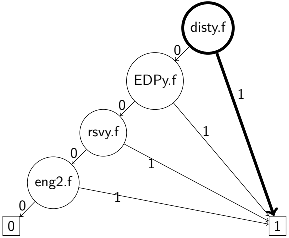


FIGURE – BDD of the loss of yellow system

Paths from root to 1 terminal ⇒ implicants

Implicant

Product $\{disty.f\}$ is an implicant of φ

Binary Decision Diagram (BDD)

Why introducing BDD ?

- compact representation of formulae based on Shannon decomposition
- used to compute prime implicant and probabilities
- to play with BDD `formal.cs.utah.edu:8080/pbl/BDD.php`

Prime Implicant Computation

Morreale Decomposition Theorem

Let $\varphi = \textbf{ite}(v, \varphi|_{v=\mathbf{T}}, \varphi|_{v=\mathbf{F}})$ then

$$PI(\varphi) = PI_{-} \cup PI_{\mathbf{T}} \cup PI_{\mathbf{F}}$$

where

$$\begin{aligned}
PI_{-} &= PI(\varphi|_{v=\mathbf{T}} \textbf{ and } \varphi|_{v=\mathbf{F}}) \\
PI_{\mathbf{T}} &= \{\{v\} \cup X | X \in PI(\varphi|_{v=\mathbf{T}}) \textbf{ and } X \notin PI_{-}\} \\
PI_{\mathbf{F}} &= \{\{\textbf{not } v\} \cup X | X \in PI(\varphi|_{v=\mathbf{F}}) \textbf{ and } X \notin PI_{-}\} \\
PI(\mathbf{F}) &= \emptyset \\
PI(\mathbf{T}) &= \{\emptyset\}
\end{aligned}$$

Prime implicant computation

Compute PI of $\varphi = (a\ \textbf{and}\ b)\ \textbf{or}\ (\textbf{not}\ a\ \textbf{and}\ c)$:

Prime Implicant Computation

Prime implicant computation

Compute PI of $\varphi = (a \textbf{ and } b) \textbf{ or } (\textbf{not } a \textbf{ and } c)$:

1. $\varphi = \textbf{ite}(a, b, c)$
2. $PI(\varphi|_{a=\mathsf{T}}) = PI(b) = \{\{b\}\}$
3. $PI(\varphi|_{a=\mathsf{F}}) = PI(c) = \{\{c\}\}$
4. $PI_{-} = PI(\varphi|_{a=\mathsf{T}} \textbf{ and } \varphi|_{a=\mathsf{F}}) = PI(b \textbf{ and } c) = \{\{b, c\}\}$
5. $PI(\varphi|_{a=\mathsf{T}}) \cap PI_{-} = \emptyset$ so $PI_{\mathsf{T}} = \{\{a, b\}\}$
6. $PI(\varphi|_{a=\mathsf{F}}) \cap PI_{-} = \emptyset$ so $PI_{\mathsf{F}} = \{\{\textbf{not } a, c\}\}$
7. $PI(\varphi) = \{\{a, b\}, \{\textbf{not } a, c\}, \{b, c\}\}$

What does $\{$**not** $a, c\}$ implicant mean ?

Minimal cutsets computation

Negative literals in prime implicants

$\Downarrow$

Some components must "work" to trigger the failure condition

$\Downarrow$

No miracle rule : Considering that component failure can mitigate the failure condition should be avoided

$\Downarrow$ Pessimistic approach (safe)

Minimal cutsets = Positive part of prime implicants

Minimal cutsets computation

Cut sets computation

Let $\varphi = \textbf{ite}(v, \varphi|_{v=\mathbf{T}}, \varphi|_{v=\mathbf{F}})$ then

$$MCS(\varphi) = MCS_{\mathbf{F}} \cup MCS_{\mathbf{T}}$$

where

$$\begin{aligned}
MCS_{\mathbf{F}} &= \{X | X \in MCS(\varphi|_{v=\mathbf{F}})\} \\
MCS_{\mathbf{T}} &= \{\{v\} \cup X | X \in MCS(\varphi|_{v=\mathbf{T}}) \textbf{ and } X \notin MCS_{\mathbf{F}}\} \\
MCS(\mathbf{F}) &= \emptyset \\
MCS(\mathbf{T}) &= \{\emptyset\}
\end{aligned}$$

Minimal cutsets computation

Compute MCS of $\varphi = (a$ **and** $b)$ **or** (**not** a **and** $c)$:

Minimal cutsets computation

Minimal cutsets computation

Compute MCS of $\varphi = (a$ **and** $b)$ **or** $($**not** a **and** $c)$:

1. $\varphi = \textbf{ite}(a, b, c)$
2. $MCS(\varphi|_{a=\mathsf{T}}) = MCS(b) = \{\{b\}\}$
3. $MCS(\varphi|_{a=\mathsf{F}}) = MCS(c) = \{\{c\}\}$
4. $MCS_{\mathsf{F}} = MCS(\varphi|_{a=\mathsf{F}}) = \{\{c\}\}$
5. $MCS(\varphi|_{a=\mathsf{T}}) \cap MCS_{\mathsf{F}} = \emptyset$ so $MCS_{\mathsf{T}} = \{\{a, b\}\}$
6. $MCS(\varphi) = \{\{a, b\}, \{c\}\}$

$$PI(\varphi) = \{\{a, b\}, \{\textbf{not}\ a, c\}, \{b, c\}\}$$

$$\Downarrow \text{Pessimism}$$

$$MCS(\varphi) = \{\{a, b\}, \{c\}\}$$

Probability computation

Option 1 : Approximate computation MCS : minimal cutsets for FC, and $p(event)$ probability of failure for primary events :

$$p(FC) = \sum_{cut \in MCS} \prod_{event \in cut} p(event)$$

Approximate computation

Let MCS=$\{\{a,b\},\{c\}\}$ be the minimal cutsets for FC :

$$p_{approx}(FC) = p(a)p(b) + p(c)$$

Probability computation

Option 2 : Exact computation Shannon decomposition :

$$\begin{aligned} p(\textbf{ite}(v, \varphi|_{v=\mathbf{T}}, \varphi|_{v=\mathbf{F}})) &= p(v)p(\varphi|_{v=\mathbf{T}}) + (1 - p(v))p(\varphi|_{v=\mathbf{F}}) \\ p(\mathbf{T}) &= 1 \\ p(\mathbf{F}) &= 0 \end{aligned}$$

Exact computation

Let $\varphi = \textbf{ite}(a, b, c)$ be the Shannon decomposition for FC :

$$p(FC) = p(a)p(b) + (1 - p(a))p(c)$$

Pessimism introduced by approximation $\left(p(x) = 10^{-3}\right)$:

$$\frac{p_{approx}(FC) - p(FC)}{p(FC)} = \frac{p(a)p(c)}{p(a)p(b) + (1 - p(a))p(c)} \simeq .1\%$$

OK but is the hydraulic system is safe or not ?

Safety objectives (Reminder)

criticality	qualitative requirement	quantitative requirement
Catastrophic	order ≥ 2	$\overline{\Lambda} \leq 10^{-9} / flight\ hour$
Hazardous	order ≥ 1	$\overline{\Lambda} \leq 10^{-7} / flight\ hour$
Major	order ≥ 1	$\overline{\Lambda} \leq 10^{-5} / flight\ hour$
Minor	order ≥ 1	$\overline{\Lambda} \leq 10^{-3} / flight\ hour$

TABLE – Acceptability matrix

Order and Mean failure rate

Order

The order is the minimal cardinality of MCS

Order

The order of $MCS = \{\{a,b\},\{c\}\}$ is 1

Mean failure rate

Mean failure rate is $\overline{\Lambda}(T) \underset{0}{\sim} \frac{\overline{R(T)}}{T}$

Mean failure rate

The mean failure rate of $MCS = \{\{a,b\},\{c\}\}$ at T is $\overline{\Lambda}(T) \underset{0}{\sim} \frac{p(a)p(b)+p(c)}{T}$

Requirements verification

⚠ We assume that primary events are independent

1. Determine the failure conditions and their criticality (from FHA)
2. Build the fault trees for each failure condition
3. Compute the minimal cutsets
4. Qualitative verification : Compute the order and compare it to the required bound
5. Quantitative verification : Compute the probability and compare it to the required bound

Requirements verification

Check the requirements for yellow system

1. our failure condition "loss of yellow system" is Minor
   $\Rightarrow$ order ≥ 1 and $p(FC) \leq 10^{-3}$
2. fault tree (cf slide 77)
3. the minimal cutsets are $MCS = \{\{disty.f\}, \{eng2.f\}, \{EDPy.f\}, \{rsvy.f\}\}$
4. the order is 1 $\Rightarrow$ qualitative requirement OK
5. let assume that $p(event) = 10^{-4}/FH$ for all events then :

$$\begin{aligned} p_{approx}(FC) &= p(disty.f) + p(EDPy.f) + p(eng2.f) + p(rsvy.f) \\ &= 4.10^{-4} \Rightarrow \text{quantitative requirement OK} \end{aligned}$$

Check the hydraulic system considering Loss of the green system is Minor

Solution

1. our failure condition "loss of green system" is Minor
   $\Rightarrow$ order ≥ 1 and $p(FC) \leq 10^{-3}$
2. fault tree (cf slide 79)
3. the minimal cutsets are :

$$MCS = \left\{ \begin{array}{ll} \{distg.f\}, & \{rsvg.f\}, \\ \{EMPg.f, EDPg.f\}, & \{EMPg.f, eng1.f\}, \\ \{elec.f, EDPg.f\}, & \{elec.f, eng1.f\} \end{array} \right\}$$

4. the order is 1 $\Rightarrow$ qualitative requirement OK
5. let assume that $p(event) = 10^{-4}/FH$ for all events then :

$$\begin{array}{rcl} p_{approx}(FC) & = & 2.10^{-4} + 4.10^{-8} \\ & \simeq & 2.10^{-4} \Rightarrow \text{quantitative requirement OK} \end{array}$$

Now a Recap

Today's lesson in 30"

Safety assessment process

1. Identify the failure conditions
2. Find the safety objectives (slide 112)
3. If the system is static build the fault tree (slide 75)
4. Compute the order of the cutsets (slide 112)
5. Compute the probability out of minimal cutsets (slide 109)
6. Compare it to the objectives

You understand highlighted terms
$\Rightarrow$ congratulations you've got the idea
Otherwise check out the slides !

Bibliography I

[ALRL04] Algirdas Avizienis, J-C Laprie, Brian Randell, and Carl Landwehr.
Basic concepts and taxonomy of dependable and secure computing.
IEEE transactions on dependable and secure computing, 1(1) :11–33, 2004.

[BDS11] Pierre Bieber, Rémi Delmas, and Christel Seguin.
Dalculus–theory and tool for development assurance level allocation.
In *Computer Safety, Reliability, and Security*, pages 43–56. Springer, 2011.

[ISO10] ISO.
ISO-26262 -Road vehicles – Functional safety, 2010.

[SAE96] SAE.
Aerospace Recommended Practices 4761 - guidelines and methods for conducting the safety assessment process on civil airborne systems and equipment, 1996.

[SAE10] SAE.
Aerospace Recommended Practices 4754a - Development of Civil Aircraft and Systems, 2010.

Thank you

ONERA

THE FRENCH AEROSPACE LAB

www.onera.fr