On Qualitative Analysis of Fault Trees Using Structurally Persistent Nets

Ricardo J. Rodríguez
rj.rodriguez@unileon.es

Research Institute of Applied Sciences in Cybersecurity
University of León, Spain

June 10, 2015
XXIII Jornadas de Concurrencia y Sistemas Distribuidos
Málaga (Spain)

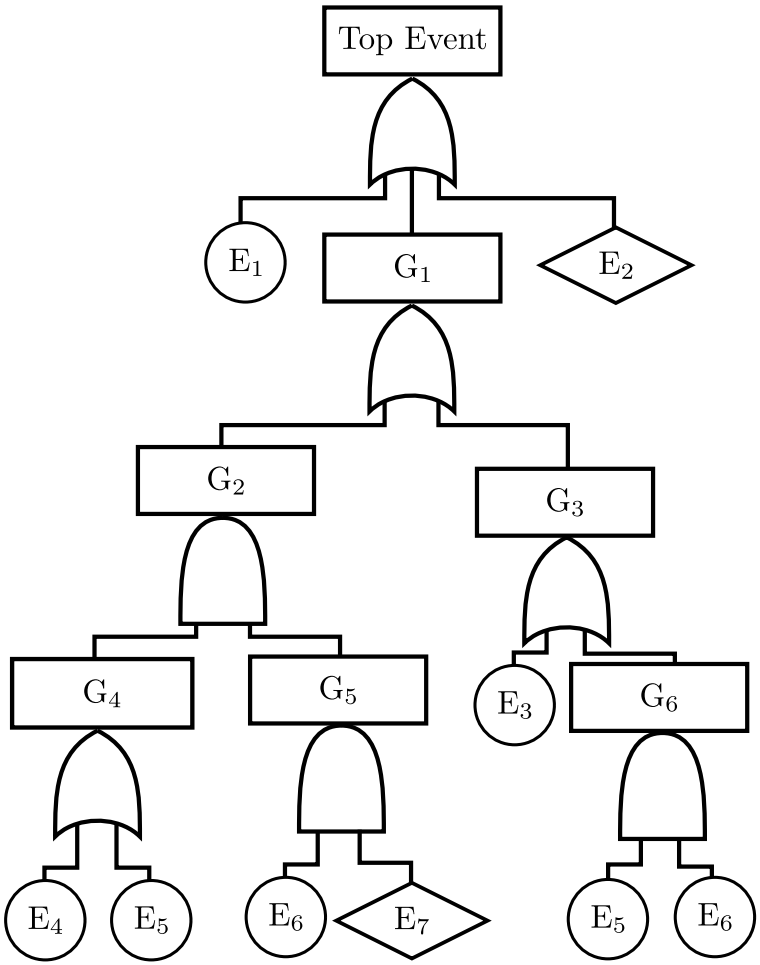
To appear in IEEE Trans. on Systems, Man, and Cybernetics: Systems
doi: 10.1109/TSMC.2015.2437360

Agenda

Agenda

Introduction (I)

Definition of Fault Tree

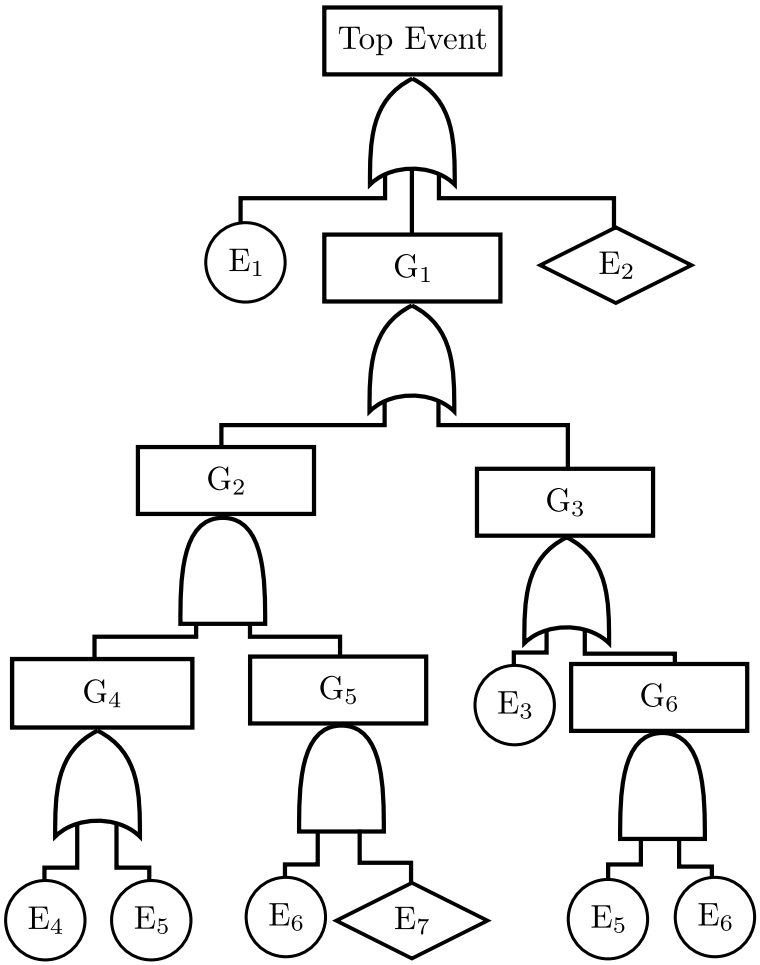


Fault Tree

- Event-driven failure logic
- Top Event: undesired state (@ the root)
- Gates: describe logic that relates events
- Event: different kind (next slide)

Introduction (I)

Definition of Fault Tree

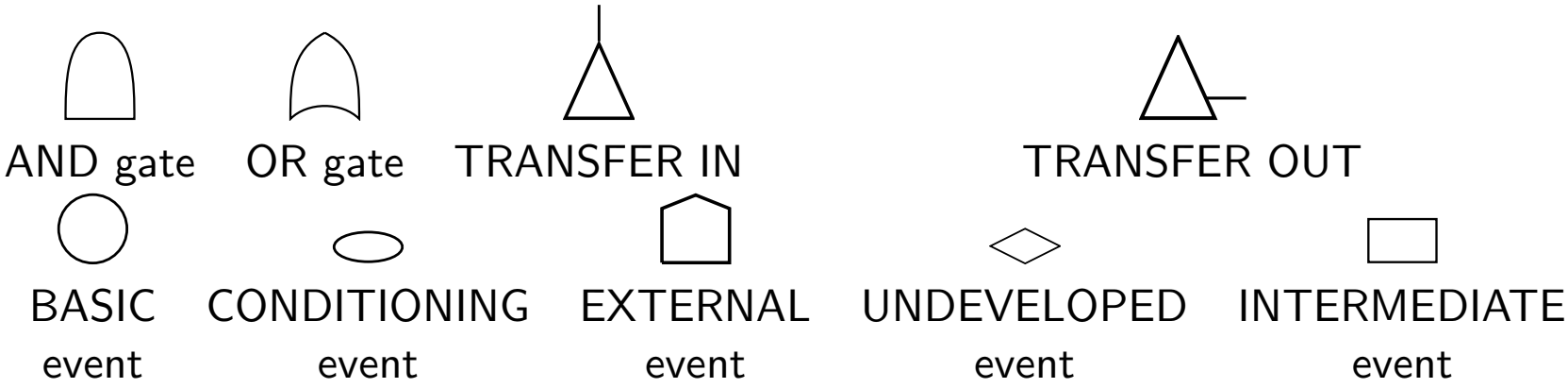


Fault Tree

- Event-driven failure logic
- Top Event: undesired state (@ the root)
- Gates: describe logic that relates events
- Event: different kind (next slide)
- Coherent Fault Tree: logic restricted to AND/OR formulae

Introduction (II)

A bit more of Fault Trees...



Graphical symbols

- AND / OR gates
- Event type:
  - Basic: component/human fault; failure & repair data available
  - Conditioning: gate triggered by an event
  - External (or house): normally expected to occur
  - Undeveloped: no further developed (e.g., no consequence, lack of data)
  - Intermediate: middle/top event, generated by combination of others
- Transfer: to divide large FTs into smaller ones, or reduce duplication

Introduction (III)

Fault Tree Analysis

- Find event combinations out that leads to an undesired state
- Top-down deductive analysis technique, from the early 60s
- Used in safety and reliability engineering

Introduction (III)

Fault Tree Analysis

- Find event combinations out that leads to an undesired state
- Top-down deductive analysis technique, from the early 60s
- Used in safety and reliability engineering

(Minimal) Cut Sets

Set of basic events whose occurrence causes a system to fail

Minimal Cut Set: it cannot be further reduced, and still leads to an undesired state

Introduction (III)

Fault Tree Analysis

- Find event combinations out that leads to an undesired state
- Top-down deductive analysis technique, from the early 60s
- Used in safety and reliability engineering

(Minimal) Cut Sets

Set of basic events whose occurrence causes a system to fail
Minimal Cut Set: it cannot be further reduced, and still leads to an undesired state

(Minimal) Path Sets

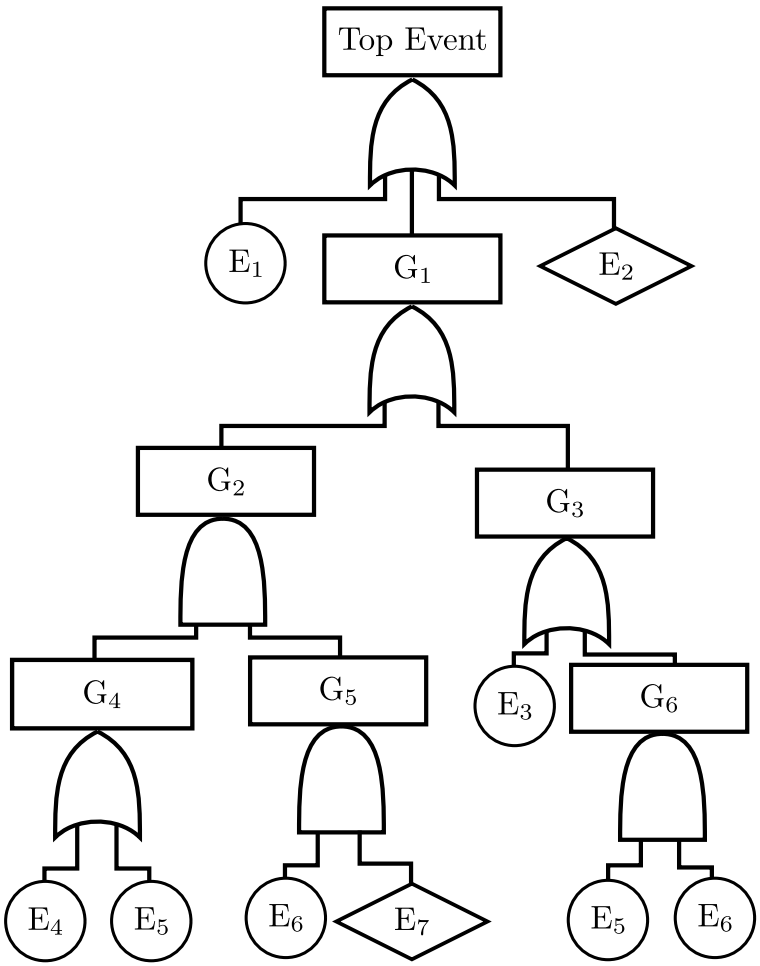
Set of basic events whose nonoccurrence assures the nonoccurrence of TE
Minimal Path Set: it cannot be further reduced, and still leads to an undesired state

MPS are a dual set of MCS

RIASC

Introduction (IV)

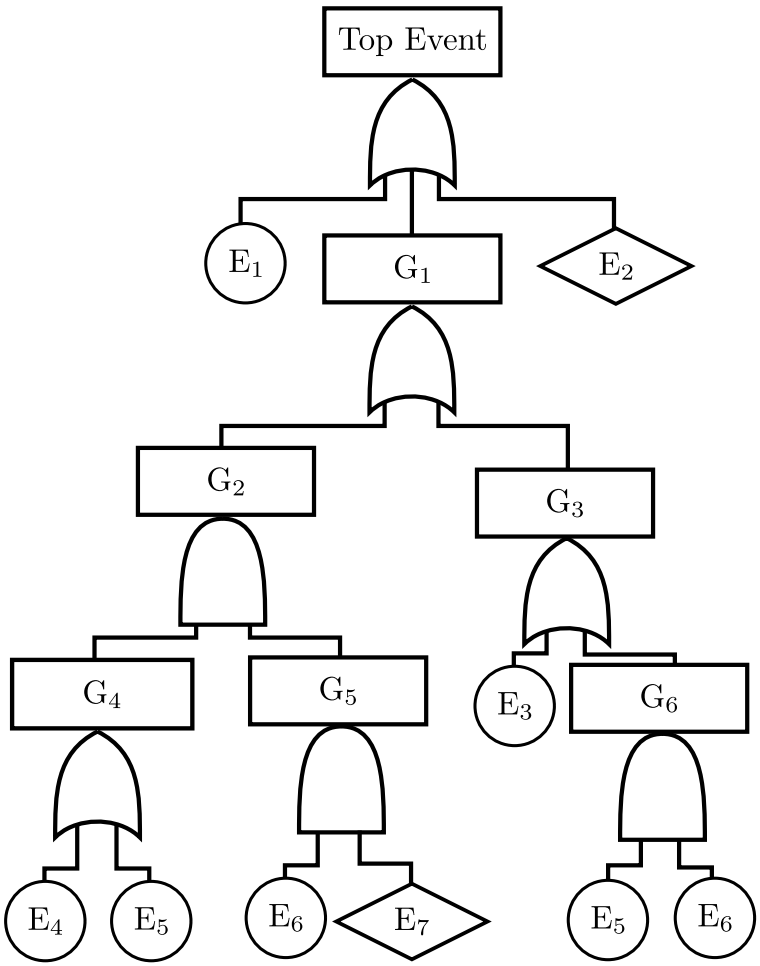
Recall the example...



- Six path sets:
  - $\mathcal{PS}_1 = \{E_1, E_2, E_3, E_4, E_5\}$
  - $\mathcal{PS}_2 = \{E_1, E_2, E_3, E_5, E_6\}$
  - $\mathcal{PS}_3 = \{E_1, E_2, E_3, E_5, E_7\}$
  - $\mathcal{PS}_4 = \{E_1, E_2, E_3, E_4, E_5, E_6\}$
  - $\mathcal{PS}_5 = \{E_1, E_2, E_3, E_6\}$
  - $\mathcal{PS}_6 = \{E_1, E_2, E_3, E_6, E_7\}$

Introduction (IV)

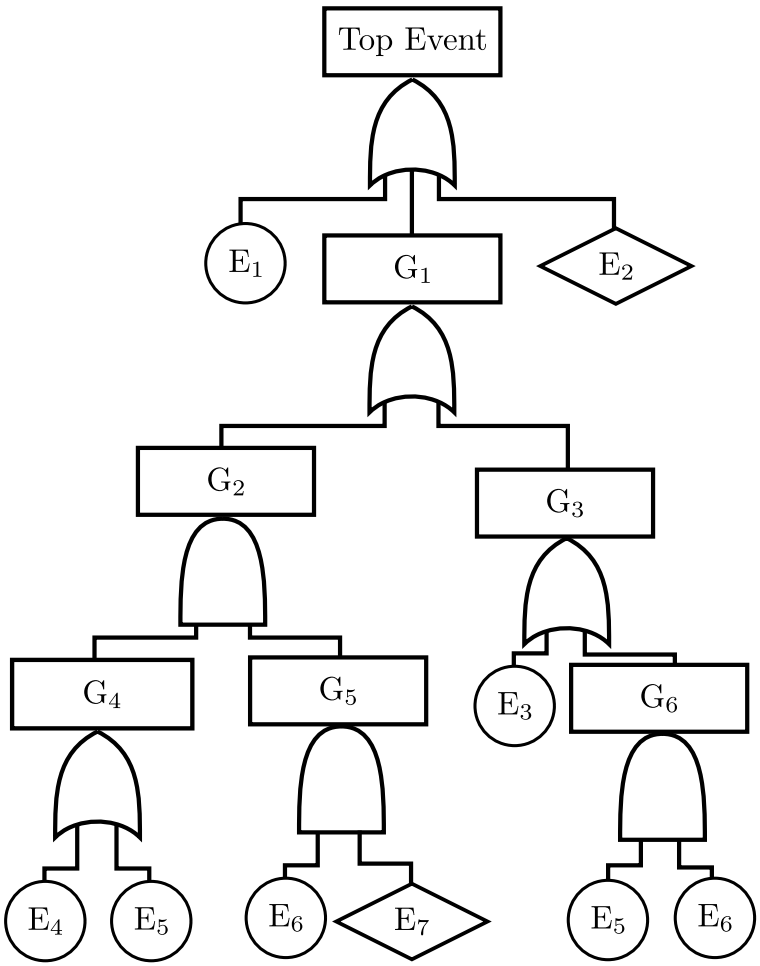
Recall the example...



- Six path sets:
  - $\mathcal{PS}_1 = \{E_1, E_2, E_3, E_4, E_5\}$
  - $\mathcal{PS}_2 = \{E_1, E_2, E_3, E_5, E_6\}$
  - $\mathcal{PS}_3 = \{E_1, E_2, E_3, E_5, E_7\}$
  - $\mathcal{PS}_4 = \{E_1, E_2, E_3, E_4, E_5, E_6\}$
  - $\mathcal{PS}_5 = \{E_1, E_2, E_3, E_6\}$
  - $\mathcal{PS}_6 = \{E_1, E_2, E_3, E_6, E_7\}$
- Not minimal!
  - $\mathcal{PS}_2 \supset \mathcal{PS}_5, \mathcal{PS}_4 \supset \mathcal{PS}_5$ (or $\mathcal{PS}_4 \supset \mathcal{PS}_1$), $\mathcal{PS}_6 \supset \mathcal{PS}_5$

Introduction (IV)

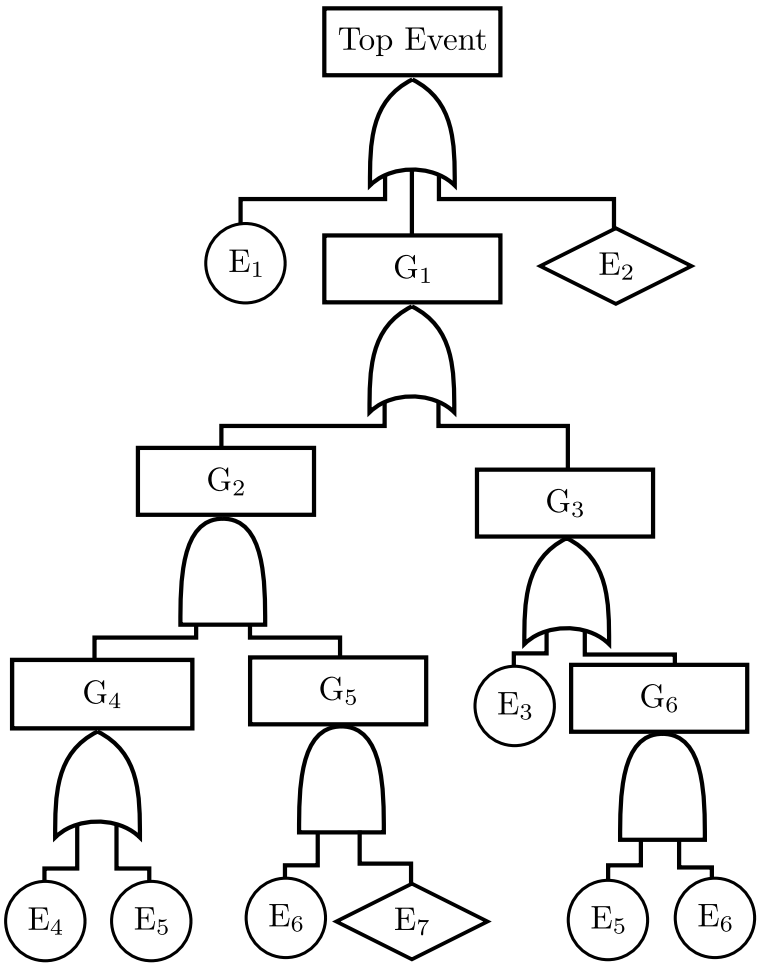
Recall the example...



- Six path sets:
  - $\mathcal{PS}_1 = \{E_1, E_2, E_3, E_4, E_5\}$
  - $\mathcal{PS}_2 = \{E_1, E_2, E_3, E_5, E_6\}$
  - $\mathcal{PS}_3 = \{E_1, E_2, E_3, E_5, E_7\}$
  - $\mathcal{PS}_4 = \{E_1, E_2, E_3, E_4, E_5, E_6\}$
  - $\mathcal{PS}_5 = \{E_1, E_2, E_3, E_6\}$
  - $\mathcal{PS}_6 = \{E_1, E_2, E_3, E_6, E_7\}$
- Not minimal!
  - $\mathcal{PS}_2 \supset \mathcal{PS}_5, \mathcal{PS}_4 \supset \mathcal{PS}_5$ (or $\mathcal{PS}_4 \supset \mathcal{PS}_1), \mathcal{PS}_6 \supset \mathcal{PS}_5$
- MPS: $\mathcal{PS}_1, \mathcal{PS}_3$, and $\mathcal{PS}_5$

Introduction (IV)

Recall the example...



- Six path sets:
  - $\mathcal{PS}_1 = \{E_1, E_2, E_3, E_4, E_5\}$
  - $\mathcal{PS}_2 = \{E_1, E_2, E_3, E_5, E_6\}$
  - $\mathcal{PS}_3 = \{E_1, E_2, E_3, E_5, E_7\}$
  - $\mathcal{PS}_4 = \{E_1, E_2, E_3, E_4, E_5, E_6\}$
  - $\mathcal{PS}_5 = \{E_1, E_2, E_3, E_6\}$
  - $\mathcal{PS}_6 = \{E_1, E_2, E_3, E_6, E_7\}$
- Not minimal!
  - $\mathcal{PS}_2 \supset \mathcal{PS}_5, \mathcal{PS}_4 \supset \mathcal{PS}_5$ (or $\mathcal{PS}_4 \supset \mathcal{PS}_1), \mathcal{PS}_6 \supset \mathcal{PS}_5$
- MPS: $\mathcal{PS}_1, \mathcal{PS}_3$, and $\mathcal{PS}_5$
- Five MCS:
  - $\mathcal{MCS}_1 = \{E_1\}, \mathcal{MCS}_2 = \{E_2\}$
  - $\mathcal{MCS}_3 = \{E_3\}, \mathcal{MCS}_4 = \{E_5, E_6\}$
  - $\mathcal{MCS}_5 = \{E_4, E_6, E_7\}$

Introduction (V)

Fault Tree Assessment

- Qualitative analysis: extraction of MCS/MPS
  - Enables to characterize a TE by a logic formula
- Quantitative analysis: for given data values, compute occurrence probability of the TE

Introduction (V)

Fault Tree Assessment

- Qualitative analysis: extraction of MCS/MPS
  - Enables to characterize a TE by a logic formula
- Quantitative analysis: for given data values, compute occurrence probability of the TE

Contributions

- Computation of MCS/MPS of a FT is equal to compute minimal p-semiflows of a Petri net, obtained by model transformation
- Minimal p-semiflows are computable in polynomial time (for the subclass of PN obtained)

Agenda

Definitions (I)

Formally defining a coherent Fault Tree

Coherent fault tree

$\mathcal{F} = \langle \mathcal{E}, \mathcal{G}, \mathcal{G}^+, \mathcal{G}^*, \mathcal{T} \rangle$, where:

- $\mathcal{E}, |\mathcal{E}| \geq 1$: set of *basic, undeveloped, or external events*;
- $\mathcal{G}, |\mathcal{G}| \geq 1, \mathcal{G} \cap \mathcal{E} = \emptyset$: set of *intermediate events*;
- $\mathcal{G}^+ : \mathcal{G} \times (\mathcal{E} \cup \mathcal{G}) \to \{0, 1\}$: OR relationship between events
- $\mathcal{G}^* : \mathcal{G} \times (\mathcal{E} \cup \mathcal{G}) \to \{0, 1\}$: AND relationship between events
- $\mathcal{T} = \{g\}, g \in \mathcal{G}$: *top event*

Definitions (I)

Formally defining a coherent Fault Tree

Coherent fault tree

$\mathcal{F} = \langle \mathcal{E}, \mathcal{G}, \mathcal{G}^+, \mathcal{G}^*, \mathcal{T} \rangle$, where:

- $\mathcal{E}, |\mathcal{E}| \geq 1$: set of *basic, undeveloped, or external events*;
- $\mathcal{G}, |\mathcal{G}| \geq 1, \mathcal{G} \cap \mathcal{E} = \emptyset$: set of *intermediate events*;
- $\mathcal{G}^+ : \mathcal{G} \times (\mathcal{E} \cup \mathcal{G}) \to \{0, 1\}$: OR relationship between events
- $\mathcal{G}^* : \mathcal{G} \times (\mathcal{E} \cup \mathcal{G}) \to \{0, 1\}$: AND relationship between events
- $\mathcal{T} = \{g\}, g \in \mathcal{G}$: *top event*

Some notes...

- We denote $\mathcal{G}^+, \mathcal{G}^*$, in matrix form, i.e., $\mathcal{G}^+, \mathcal{G}^* \in \{0, 1\}^{|\mathcal{G}| \times (|\mathcal{E}| + |\mathcal{G}|)}$
- An event $g \in \mathcal{G}$ has only non-null components in either $\mathcal{G}^+$ or $\mathcal{G}^*$, and not both
- Self-feedback is not allowed in intermediate events

Definitions (II)

On Petri nets

Petri nets

A Petri net (PN) is a 4–tuple $\mathcal{N} = \langle P, T, \mathbf{Pre}, \mathbf{Post} \rangle$, where:

- P and T are disjoint non-empty sets of *places* and *transitions*; and
- **Pre** (**Post**) are the pre–(post–)incidence non-negative integer matrices of size $|P| \times |T|$

Definitions (II)

On Petri nets

Petri nets

A Petri net (PN) is a 4–tuple $\mathcal{N} = \langle P, T, \mathbf{Pre}, \mathbf{Post} \rangle$, where:

- P and T are disjoint non-empty sets of *places* and *transitions*; and
- **Pre** (**Post**) are the pre–(post–)incidence non-negative integer matrices of size $|P| \times |T|$
- A Petri net system $\mathcal{S} = \langle \mathcal{N}, \mathbf{m_0} \rangle$ is a Petri net $\mathcal{N}$ with an initial marking $\mathbf{m_0}$

Reachability Set and Boundedness

- $RS(\mathcal{N}, \mathbf{m_0})$: set of markings *reachable* from $\mathbf{m_0}$ in $\mathcal{N}$
- A place $p \in P$ is $k - bounded$ if $\forall \mathbf{m} \in RS(\mathcal{N}, \mathbf{m_0}), \mathbf{m}(p) \leq k$
  - A net system $\mathcal{S}$ is *k-bounded* if each place is k-bounded
  - A net system is *bounded* if $\exists$ some k for which it is k-bounded

Definitions (IV)

Identical and series places

- A place p is *identical* to a place $p' \neq p$ if $\mathbf{m_0}(p) = \mathbf{m_0}(p'), \mathbf{Pre}(p, \cdot) = \mathbf{Pre}(p', \cdot)$, and $\mathbf{Post}(p, \cdot) = \mathbf{Post}(p', \cdot)$
- Places $p, p' \neq p$, are *series* places if $\mathbf{Pre}(p, \cdot) = \mathbf{Post}(p', \cdot)$

P-Semiflows

- $\mathbf{y} \geq \mathbf{0}$ such that $\mathbf{y}^\top \cdot \mathbf{C} = \mathbf{0}$
- Token conservation law independent of any firing of transitions
- Minimal p-semiflow: $\|\mathbf{y}\| = \{i | \mathbf{y}(i) \neq 0\}$, is not a proper superset of the support of any other p-semiflow, and the greatest common divisor of its elements is one
- Conservativeness: all places are covered by a p-semiflow

RIASC

Definitions (V)

Transition conflicts

- *Structural conflict*: ${}^\bullet t \cap {}^\bullet t' \neq \emptyset$
- *Effective conflict for a marking* $\mathbf{m}$: t, t' in structural conflict and both enabled at $\mathbf{m}$

Persistent net

For any reachable marking $\mathbf{m}$ and for all transitions $t_i, t_j, t_i \neq t_j$, enabled in $\mathbf{m}$, the sequence t_i, t_j is firable from $\mathbf{m}$

Structurally persistent net (SPN)

When $\langle \mathcal{N}, \mathbf{m_0} \rangle$ is persistent for all finite initial markings $\mathbf{m_0}$

- SPN are totally conflict-free, i.e., no pair of transitions is in structural or effective conflict. That is, $\forall p \in P, |p^\bullet| \leq 1$

RIASC

Agenda

Model Transformation: from a $\mathcal{FT}$ to a SPN $\langle \mathcal{N}, \mathbf{m_0} \rangle$

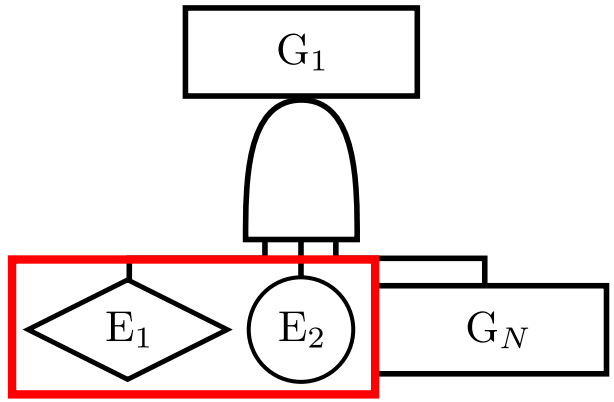
- P in $\mathcal{N}$ is divided into three disjoint sets P_E, P_G, P_{EG}

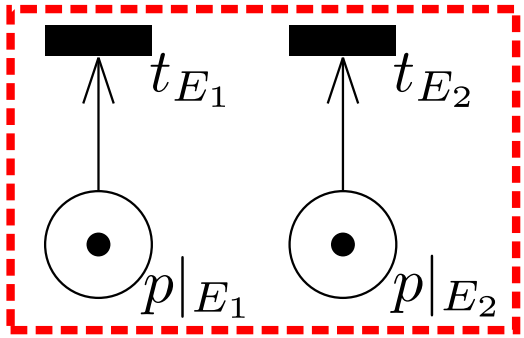
Model Transformation: from a $\mathcal{FT}$ to a SPN $\langle \mathcal{N}, \mathbf{m_0} \rangle$

- P in $\mathcal{N}$ is divided into three disjoint sets P_E, P_G, P_{EG}

Steps

1. Transform every event $e \in \mathcal{E}$



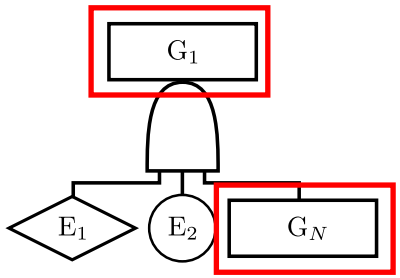


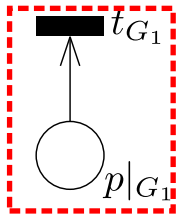
Model Transformation: from a $\mathcal{FT}$ to a SPN $\langle \mathcal{N}, \mathbf{m_0} \rangle$

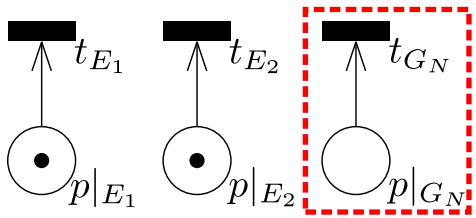
- P in $\mathcal{N}$ is divided into three disjoint sets P_E, P_G, P_{EG}

Steps

1. Transform every event $e \in \mathcal{E}$
2. Transform every event $g \in \mathcal{G}$





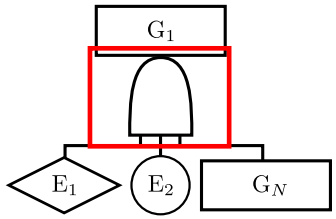


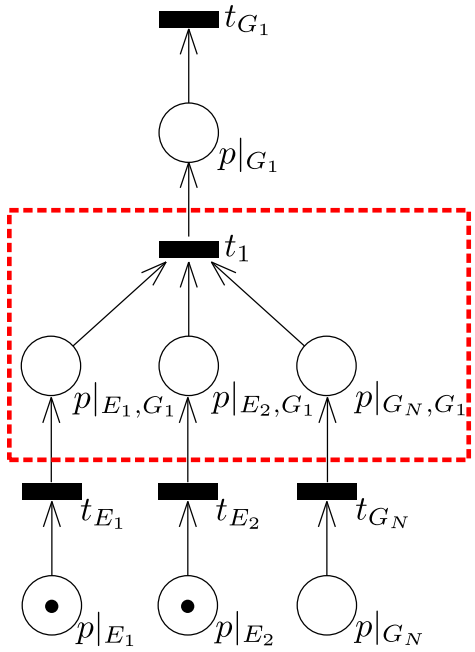
Model Transformation: from a $\mathcal{FT}$ to a SPN $\langle \mathcal{N}, \mathbf{m_0} \rangle$

- P in $\mathcal{N}$ is divided into three disjoint sets P_E, P_G, P_{EG}

Steps

1. Transform every event $e \in \mathcal{E}$
2. Transform every event $g \in \mathcal{G}$
3. Transform gate connections
   - AND gate



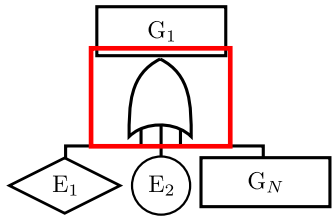


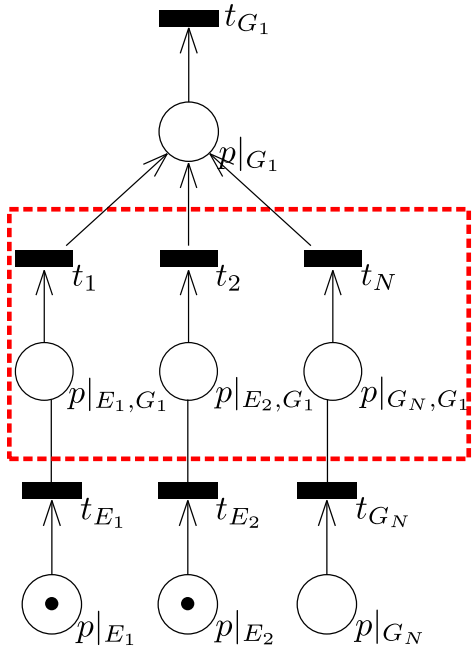
Model Transformation: from a $\mathcal{FT}$ to a SPN $\langle \mathcal{N}, \mathbf{m_0} \rangle$

- P in $\mathcal{N}$ is divided into three disjoint sets P_E, P_G, P_{EG}

Steps

1. Transform every event $e \in \mathcal{E}$
2. Transform every event $g \in \mathcal{G}$
3. Transform gate connections
   - AND gate
   - OR gate



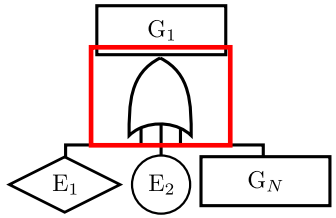


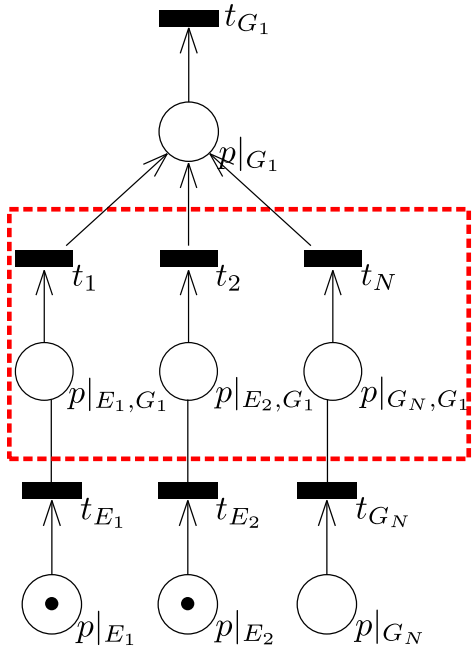
Model Transformation: from a $\mathcal{FT}$ to a SPN $\langle \mathcal{N}, \mathbf{m_0} \rangle$

- P in $\mathcal{N}$ is divided into three disjoint sets P_E, P_G, P_{EG}

Steps

1. Transform every event $e \in \mathcal{E}$
2. Transform every event $g \in \mathcal{G}$
3. Transform gate connections
   - AND gate
   - OR gate
4. Remove t_g of place $p_g, g = \mathcal{T}$



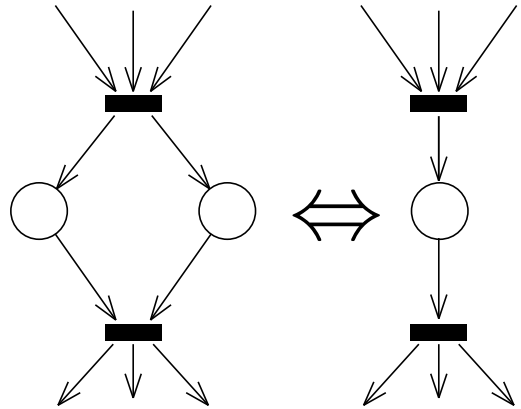


Model Transformation: from a $\mathcal{FT}$ to a SPN $\langle \mathcal{N}, \mathbf{m_0} \rangle$

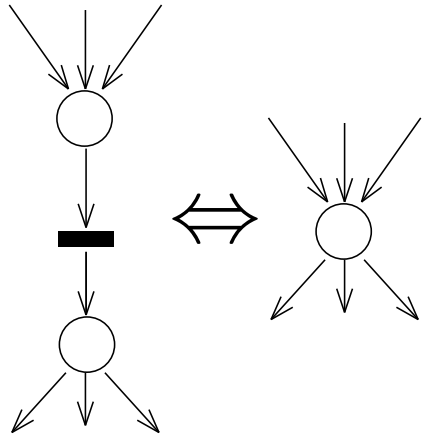
- P in $\mathcal{N}$ is divided into three disjoint sets P_E, P_G, P_{EG}

Steps

1. Transform every event $e \in \mathcal{E}$
2. Transform every event $g \in \mathcal{G}$
3. Transform gate connections
   - AND gate
   - OR gate
4. Remove t_g of place $p_g, g = \mathcal{T}$
5. Petri net reductions rules applied
   - Elimination of identical places
   - Fusion of series places



(a) Elimination of identical places



(b) Fusion of series places

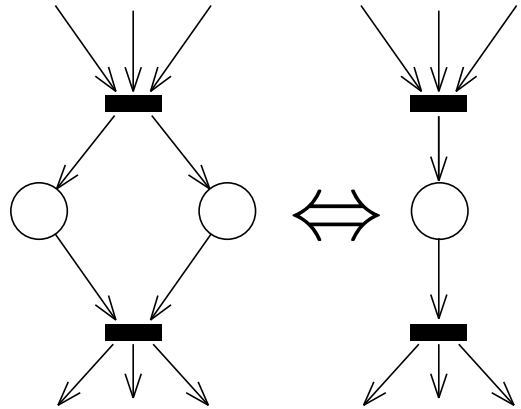
Model Transformation: from a $\mathcal{FT}$ to a SPN $\langle \mathcal{N}, \mathbf{m_0} \rangle$

- P in $\mathcal{N}$ is divided into three disjoint sets P_E, P_G, P_{EG}

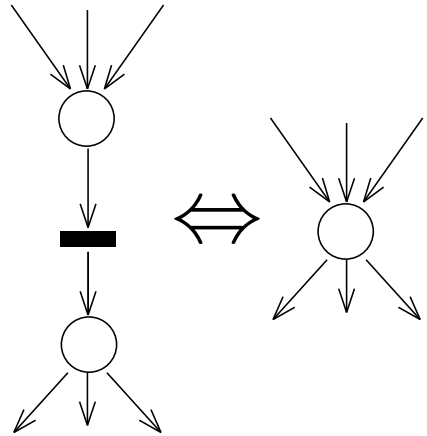
Steps

1. Transform every event $e \in \mathcal{E}$
2. Transform every event $g \in \mathcal{G}$
3. Transform gate connections
   - AND gate
   - OR gate
4. Remove t_g of place $p_g, g = \mathcal{T}$
5. Petri net reductions rules applied
   - Elimination of identical places
   - Fusion of series places

- Acyclic
- Bounded ($\forall t \in T, |{}^\bullet t| \geq 1$)



(a) Elimination of identical places



(b) Fusion of series places

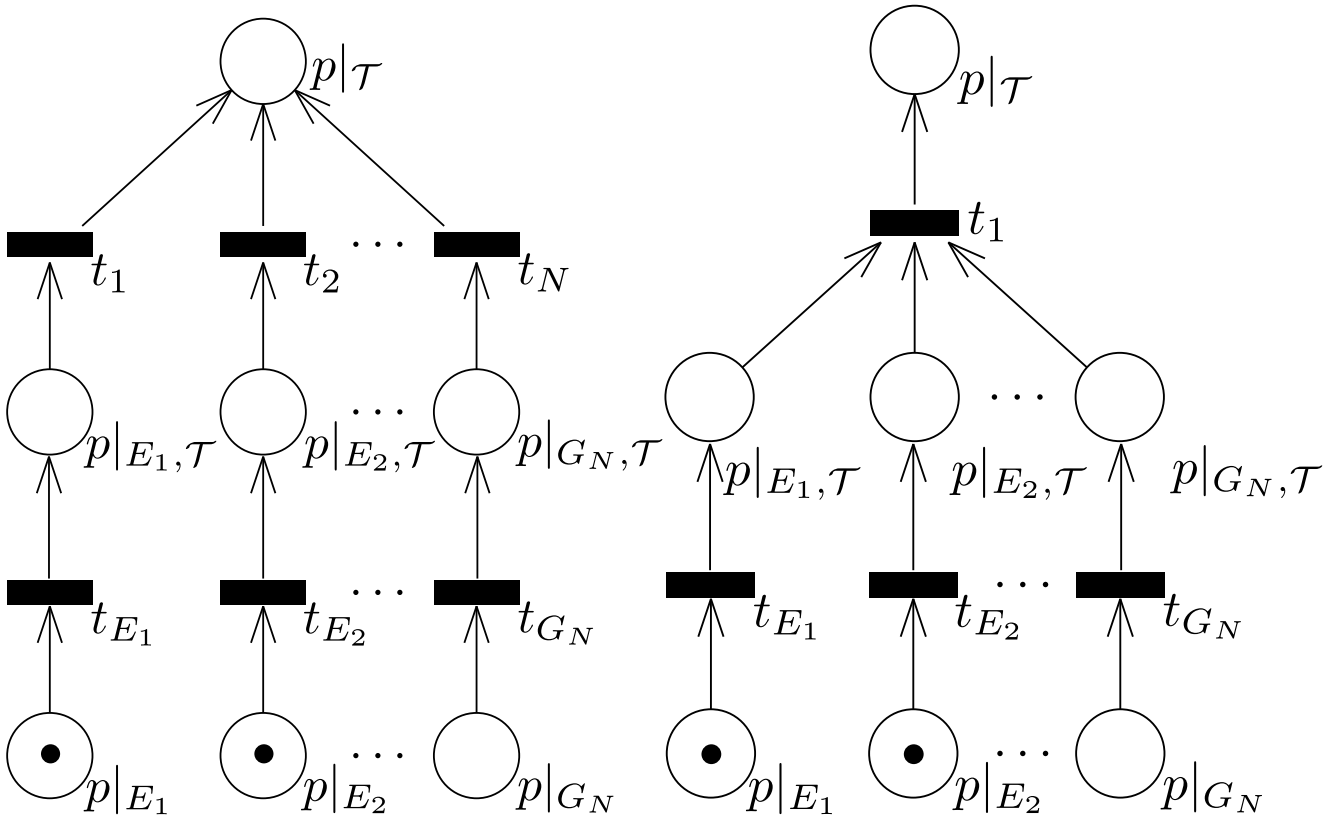
Agenda

Fault Tree Analysis using P-Semiflows (I)

FT-SPN $\mathcal{S}_{\mathcal{F}} = \langle \mathcal{N}, \mathcal{R}, \mathbf{m_0} \rangle$ obtained by transformation

Theorem

An FT-SPN is conservative



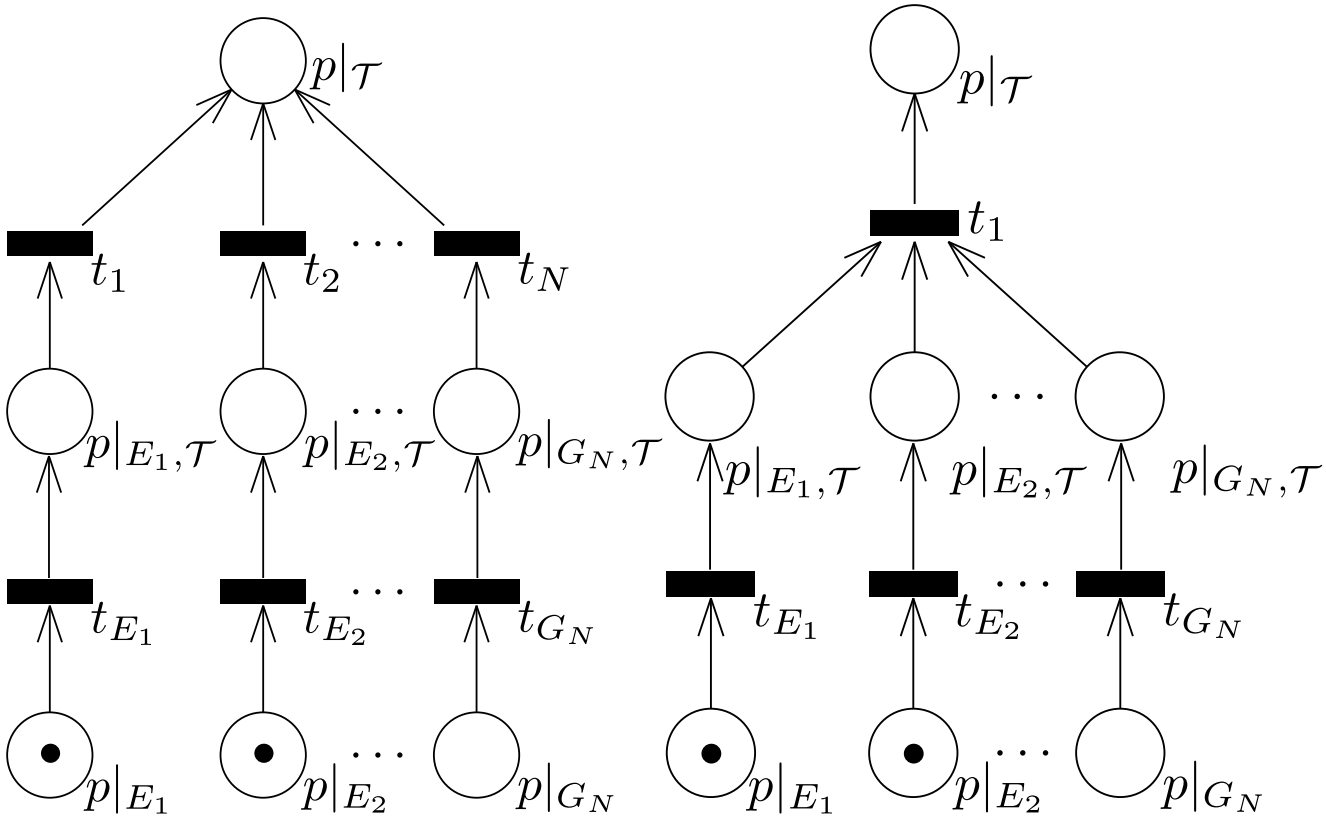
Starting at the top event, we can reach the basic events recursively. . .

Fault Tree Analysis using P-Semiflows (II)

$\mathcal{S}_{\mathcal{F}} = \langle \mathcal{N}, \mathcal{R}, \mathbf{m_0} \rangle$ obtained by transformation of $\mathcal{F} = \langle \mathcal{E}, \mathcal{G}, \mathcal{G}^+, \mathcal{G}^*, \mathcal{T} \rangle$

Theorem

The set of places $p \in P_E$ contained in the support of a minimal p-semiflow of $\mathcal{N}$ representing events $e \in \mathcal{E}$ defines a path set of $\mathcal{F}$



Fault Tree Analysis using P-Semiflows (III)

Theorem

A minimal p-semiflow $\mathbf{y}$ *of a FT-SPN, after applying reduction rules, that includes* $p \in P_E$ *in its support, i.e.,* $p \in \|\mathbf{y}\|$*, can be computed by the following Linear Programming problem:*

$$
\begin{aligned}
&\textit{maximize } \mathbf{y}(p) \\
&\textit{subject to} \\
&\mathbf{y}^\top \cdot \mathbf{C} = \mathbf{0} \\
&\mathbf{y}^\top \cdot \mathbf{m_0} = 1 \\
&\mathbf{y} \geq \mathbf{0}
\end{aligned}
$$

Proof.

- Suppose that $\mathbf{y} = \sum_{i=1}^{n} \alpha_i \cdot \mathbf{y}_i, \alpha_i > 0$

Fault Tree Analysis using P-Semiflows (III)

Theorem

A minimal p-semiflow $\mathbf{y}$ *of a FT-SPN, after applying reduction rules, that includes* $p \in P_E$ *in its support, i.e.,* $p \in \|\mathbf{y}\|$*, can be computed by the following Linear Programming problem:*

$$\begin{aligned} &\textit{maximize } \mathbf{y}(p) \\ &\textit{subject to} \\ &\mathbf{y}^\top \cdot \mathbf{C} = \mathbf{0} \\ &\mathbf{y}^\top \cdot \mathbf{m_0} = 1 \\ &\mathbf{y} \geq \mathbf{0} \end{aligned}$$

Proof.

- Suppose that $\mathbf{y} = \sum_{i=1}^{n} \alpha_i \cdot \mathbf{y}_i, \alpha_i > 0$
- $\mathbf{y} \cdot \mathbf{m_0} = 1 \rightarrow \sum_{i=1}^{n} \alpha_i \cdot \mathbf{y}_i \cdot \mathbf{m_0} = \alpha_1 \cdot \mathbf{y}_1 \cdot \mathbf{m_0} + \alpha_2 \cdot \mathbf{y}_2 \cdot \mathbf{m_0} + \cdots + \alpha_n \cdot \mathbf{y}_n \cdot \mathbf{m_0} = 1, \alpha_i > 0$

Fault Tree Analysis using P-Semiflows (III)

Theorem

A minimal p-semiflow $\mathbf{y}$ *of a FT-SPN, after applying reduction rules, that includes* $p \in P_E$ *in its support, i.e.,* $p \in \|\mathbf{y}\|$*, can be computed by the following Linear Programming problem:*

$$\begin{aligned} &\textit{maximize } \mathbf{y}(p) \\ &\textit{subject to} \\ &\mathbf{y}^\top \cdot \mathbf{C} = \mathbf{0} \\ &\mathbf{y}^\top \cdot \mathbf{m_0} = 1 \\ &\mathbf{y} \geq \mathbf{0} \end{aligned}$$

Proof.

- Suppose that $\mathbf{y} = \sum_{i=1}^{n} \alpha_i \cdot \mathbf{y}_i, \alpha_i > 0$
- $\mathbf{y} \cdot \mathbf{m_0} = 1 \rightarrow \sum_{i=1}^{n} \alpha_i \cdot \mathbf{y}_i \cdot \mathbf{m_0} = \alpha_1 \cdot \mathbf{y}_1 \cdot \mathbf{m_0} + \alpha_2 \cdot \mathbf{y}_2 \cdot \mathbf{m_0} + \cdots + \alpha_n \cdot \mathbf{y}_n \cdot \mathbf{m_0} = 1, \alpha_i > 0$
- $\mathbf{m_0}(p) = 1, \forall p \in P_E, \mathbf{m_0}(p') = 0, \forall p' \in P \setminus P_E \rightarrow \mathbf{y}_i \cdot \mathbf{m_0} = \sum \mathbf{y}_i(p), p \in P_E, p \in \|\mathbf{y}_i\|$
- $\alpha_1 \cdot \sum \mathbf{y}_1(p) + \alpha_2 \cdot \sum \mathbf{y}_2(p) + \cdots + \alpha_n \cdot \sum \mathbf{y}_n(p) = 1, \alpha_i > 0$, where $p \in P_E, p \in \|\mathbf{y}_i\|, i = 1 \ldots n$
- $|||\mathbf{y}||| > |||\mathbf{y}_i|||$, $\mathbf{y}(p)$ for a given $p \in P_E$, the value of $\mathbf{y}(p)$ is not maximum

Fault Tree Analysis using P-Semiflows (IV)

Corollary

The computation of the minimal cut sets and minimal path sets of a coherent Fault Tree are solvable in polynomial time.

Fault Tree Analysis using P-Semiflows (IV)

Corollary

The computation of the minimal cut sets and minimal path sets of a coherent Fault Tree are solvable in polynomial time.

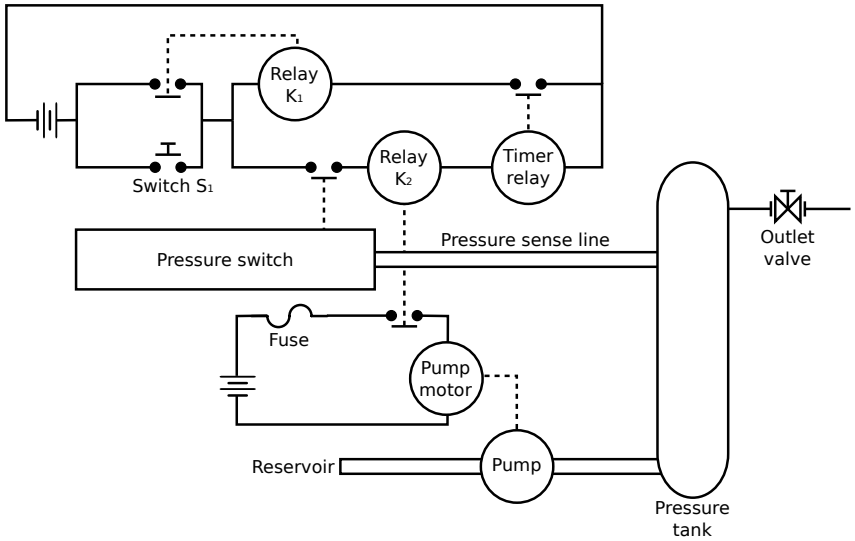
Input: $\mathcal{F} = \langle \mathcal{E}, \mathcal{G}, \mathcal{G}^+, \mathcal{G}^*, \mathcal{T} \rangle$
Output: The minimal cut sets $\mathcal{M}$ and the minimal path sets $\mathcal{M}'$ of $\mathcal{F}$

1 $\mathcal{M} = \mathcal{M}' = \emptyset$
2 Transform $\mathcal{F}$ into an FT-SPN $\mathcal{S}_\mathcal{F} = \langle \mathcal{N}_\mathcal{F}, \mathcal{R}, \mathbf{m_0} \rangle$ (see Section IV)
3 Build P_f, folding step over $\mathcal{S}_\mathcal{F}$
4 **foreach** $p \in P_E$ **do**
5 | Compute the minimal p-semiflow $\mathbf{y}$ that contains p in its support, using LPP (1)
6 | Unfolding step over $\mathbf{y}$ taking P_f into account, obtaining a set of minimal p-semiflows $\mathcal{Z}$
7 | $\forall \mathbf{z} \in \mathcal{Z}, \mathcal{M}' = \mathcal{M}' \bigcup \{e | \langle p, e \rangle \in \mathcal{R}, p \in \|\mathbf{z}\| \bigcap \|\mathbf{m_0}\|\}$
8 **end**
9 Remove the supersets from $\mathcal{M}$
10 Transform $\mathcal{F}$ into its dual $\mathcal{F}'$
11 Transform $\mathcal{F}'$ into an FT-SPN $\mathcal{S}_{\mathcal{F}'} = \langle \mathcal{N}_{\mathcal{F}'}, \mathcal{R}, \mathbf{m_0} \rangle$ (see Section IV)
12 Build P_f, folding step over $\mathcal{S}_{\mathcal{F}'}$
13 **foreach** $p \in P_E$ **do**
14 | Compute the minimal p-semiflow $\mathbf{y}$ that contains p in its support, using LPP (1)
15 | Unfolding step over $\mathbf{y}$ taking P_f into account, obtaining a set of minimal p-semiflows $\mathcal{Z}$
16 | $\forall \mathbf{z} \in \mathcal{Z}, \mathcal{M} = \mathcal{M} \bigcup \{e | \langle p, e \rangle \in \mathcal{R}, p \in \|\mathbf{z}\| \bigcap \|\mathbf{m_0}\|\}$
17 **end**
18 Remove the supersets from $\mathcal{M}$

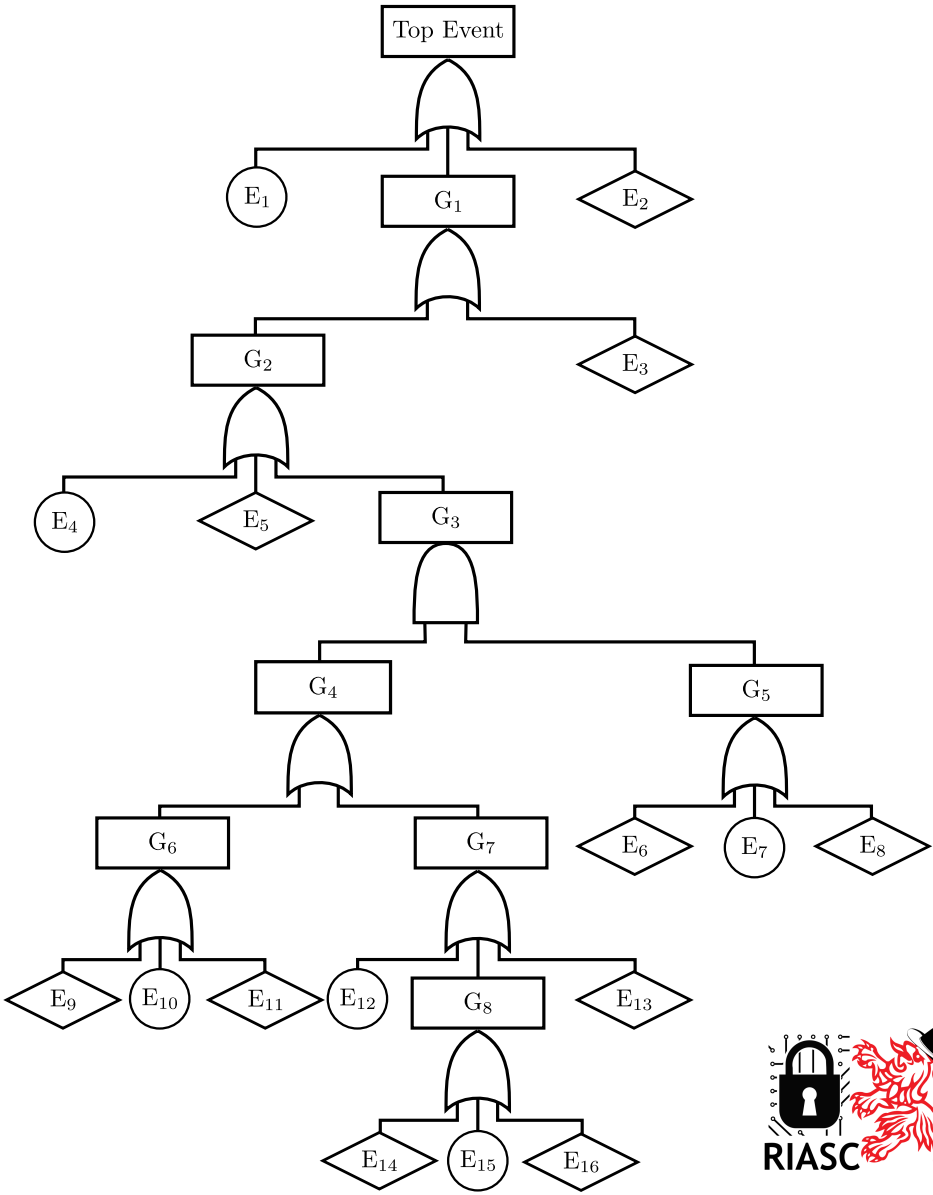
RIASC

Agenda

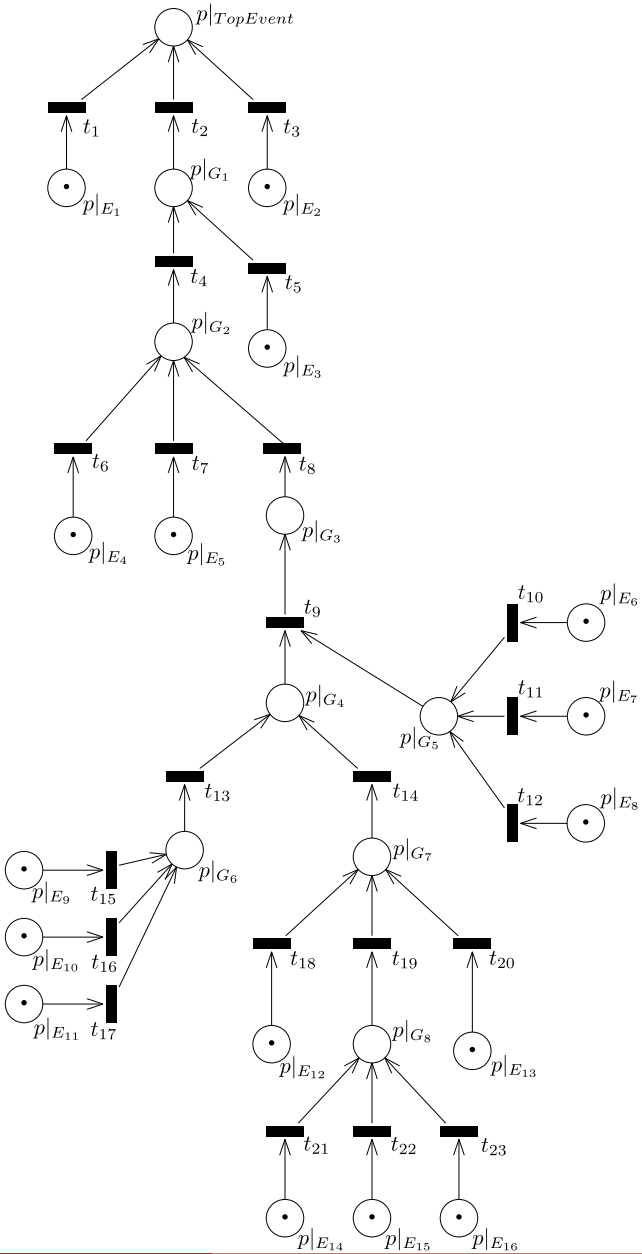
Case Study: A Pressure Tank System (I)

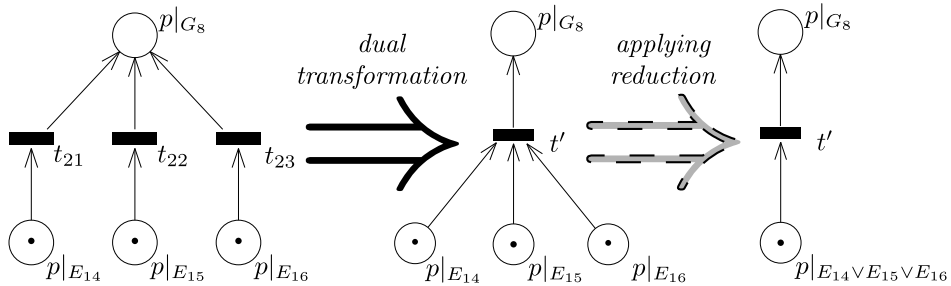


Event	Description
Top Event	Pressure tank rupture.
E_1	Pressure tank ruptures under load.
E_2	Tank ruptures due to improper installation.
G_1	Secondary failure of ruptured pressure tank.
E_3	Secondary failure of tank from some other out of tolerance conditions (e.g., mechanical, thermal).
G_2	K_2 relay contacts remain closed for a time $T > 60$ seconds.
E_4	K_2 relay contacts fail to open.
E_5	K_2 relay secondary failure.
G_3	EMF to K_2 relay coil for a time $T > 60$ seconds.
G_4	EMF remains on pressure switch (P/S) contacts when P/S contacts closed for a time $T > 60$ seconds.
G_5	P/S contacts closed, $T > 60$ seconds.
G_6	EMF through S_1 switch contacts when P/S contacts closed, $T > 60$ seconds.
G_7	EMF through K_1 relay contacts when P/S contacts closed, $T > 60$ seconds.
E_6	Pressure switch secondary failure.
E_7	Pressure switch contacts fail to open.
E_8	Excess pressure not sensed by pressure-activated switch.
E_9	S_1 switch secondary failure.
E_{10}	S_1 switch contacts fail to open.
E_{11}	External reset activation force remains on switch S_1.
E_{12}	K_1 relay contacts fail to open.
E_{13}	K_1 relay secondary failure.
G_8	Timer relay contacts fail to open when P/S contacts closed, $T > 60$ seconds.
E_{14}	Timer does not timeout due to improper setting installation.
E_{15}	Timer relay contacts fail to open.
E_{16}	Timer relay secondary failure.

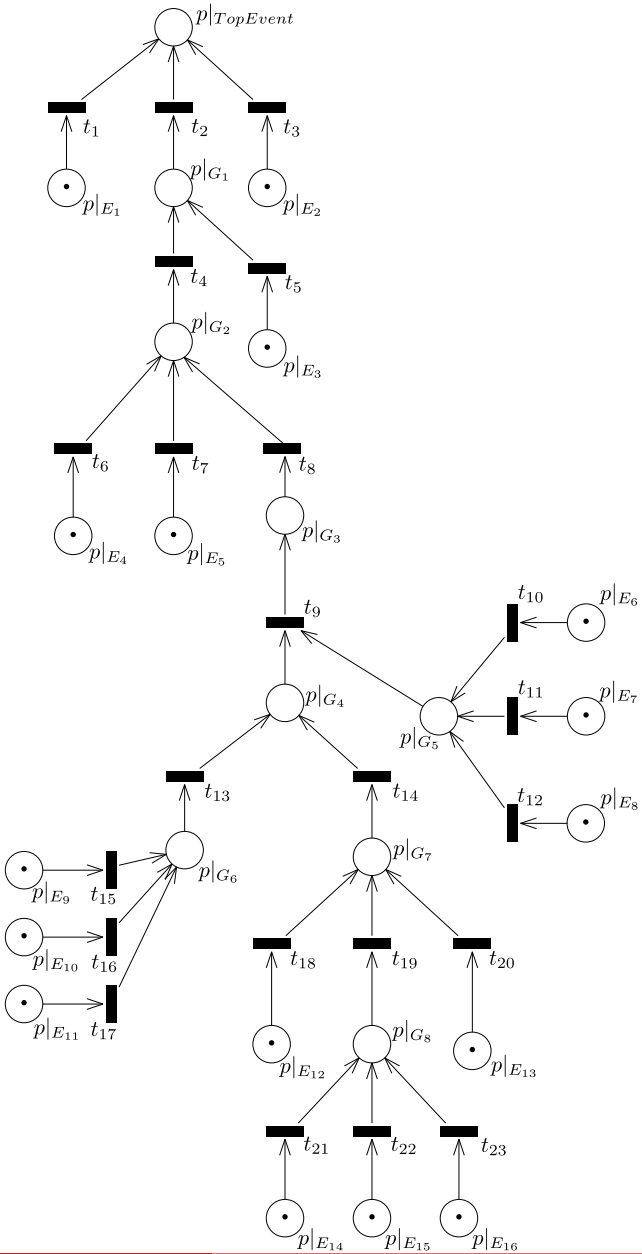


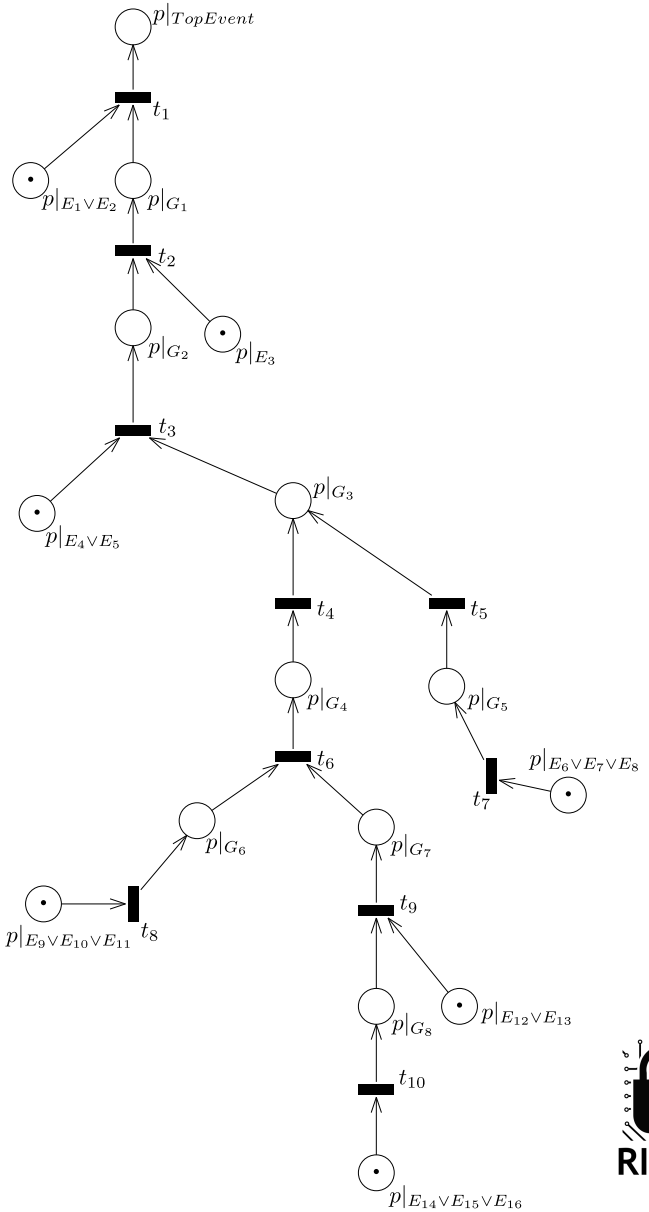
Case Study: A Pressure Tank System (II)





Case Study: A Pressure Tank System (II)





Case Study: A Pressure Tank System (II)

Place p	Minimal p-semiflow	MCS
$p\|_{E_1}$	$\mathbf{y}_1 = \{p\|_{TopEvent}, p\|_{E_1}\}$	$\{E_1\}$
$p\|_{E_2}$	$\mathbf{y}_2 = \{p\|_{TopEvent}, p\|_{E_2}\}$	$\{E_2\}$
$p\|_{E_3}$	$\mathbf{y}_3 = \{p\|_{TopEvent}, p\|_{G_1}, p\|_{E_3}\}$	$\{E_3\}$
$p\|_{E_4}$	$\mathbf{y}_4 = \{p\|_{TopEvent}, p\|_{G_1}, p\|_{G_2}, p\|_{E_4}\}$	$\{E_4\}$
$p\|_{E_5}$	$\mathbf{y}_5 = \{p\|_{TopEvent}, p\|_{G_1}, p\|_{G_2}, p\|_{E_5}\}$	$\{E_5\}$
$p\|_{E_6}$	$\mathbf{y}_6 = \{p\|_{TopEvent}, p\|_{G_1}, p\|_{G_2}, p\|_{G_3}, p\|_{G_4}, p\|_{G_5}, p\|_{E_6}, p\|_{G_6}, p\|_{E_9}\}$	$\{E_6, E_9\}$
$p\|_{E_7}$	$\mathbf{y}_7 = \{p\|_{TopEvent}, p\|_{G_1}, p\|_{G_2}, p\|_{G_3}, p\|_{G_4}, p\|_{G_5}, p\|_{E_7}, p\|_{G_6}, p\|_{E_9}\}$	$\{E_7, E_9\}$
$p\|_{E_8}$	$\mathbf{y}_8 = \{p\|_{TopEvent}, p\|_{G_1}, p\|_{G_2}, p\|_{G_3}, p\|_{G_4}, p\|_{G_5}, p\|_{E_8}, p\|_{G_6}, p\|_{E_9}\}$	$\{E_8, E_9\}$
$p\|_{E_9}$	$\mathbf{y}_9 = \mathbf{y}_6$	$\{E_6, E_9\}$
$p\|_{E_{10}}$	$\mathbf{y}_{10} = \{p\|_{TopEvent}, p\|_{G_1}, p\|_{G_2}, p\|_{G_3}, p\|_{G_4}, p\|_{G_5}, p\|_{E_6}, p\|_{G_6}, p\|_{E_{10}}\}$	$\{E_6, E_{10}\}$
$p\|_{E_{11}}$	$\mathbf{y}_{11} = \{p\|_{TopEvent}, p\|_{G_1}, p\|_{G_2}, p\|_{G_3}, p\|_{G_4}, p\|_{G_5}, p\|_{E_6}, p\|_{G_6}, p\|_{E_{11}}\}$	$\{E_6, E_{11}\}$
$p\|_{E_{12}}$	$\mathbf{y}_{12} = \{p\|_{TopEvent}, p\|_{G_1}, p\|_{G_2}, p\|_{G_3}, p\|_{G_4}, p\|_{G_5}, p\|_{E_6}, p\|_{G_7}, p\|_{E_{12}}\}$	$\{E_6, E_{12}\}$
$p\|_{E_{13}}$	$\mathbf{y}_{13} = \{p\|_{TopEvent}, p\|_{G_1}, p\|_{G_2}, p\|_{G_3}, p\|_{G_4}, p\|_{G_5}, p\|_{E_6}, p\|_{G_7}, p\|_{E_{13}}\}$	$\{E_6, E_{13}\}$
$p\|_{E_{14}}$	$\mathbf{y}_{14} = \{p\|_{TopEvent}, p\|_{G_1}, p\|_{G_2}, p\|_{G_3}, p\|_{G_4}, p\|_{G_5}, p\|_{E_6}, p\|_{G_7}, p\|_{E_{14}}\}$	$\{E_6, E_{14}\}$
$p\|_{E_{14}}$	$\mathbf{y}_{15} = \{p\|_{TopEvent}, p\|_{G_1}, p\|_{G_2}, p\|_{G_3}, p\|_{G_4}, p\|_{G_5}, p\|_{E_6}, p\|_{G_7}, p\|_{E_{15}}\}$	$\{E_6, E_{15}\}$
$p\|_{E_{16}}$	$\mathbf{y}_{16} = \{p\|_{TopEvent}, p\|_{G_1}, p\|_{G_2}, p\|_{G_3}, p\|_{G_4}, p\|_{G_5}, p\|_{E_6}, p\|_{G_7}, p\|_{E_{16}}\}$	$\{E_6, E_{16}\}$

Case Study: A Pressure Tank System (II)

Place p	Minimal p-semiflow	MCS
$p\|_{E_1}$	$\mathbf{y}_1 = \{p\|_{TopEvent}, p\|_{E_1}\}$	$\{E_1\}$
$p\|_{E_2}$	$\mathbf{y}_2 = \{p\|_{TopEvent}, p\|_{E_2}\}$	$\{E_2\}$
$p\|_{E_3}$	$\mathbf{y}_3 = \{p\|_{TopEvent}, p\|_{G_1}, p\|_{E_3}\}$	$\{E_3\}$
$p\|_{E_4}$	$\mathbf{y}_4 = \{p\|_{TopEvent}, p\|_{G_1}, p\|_{G_2}, p\|_{E_4}\}$	$\{E_4\}$
$p\|_{E_5}$	$\mathbf{y}_5 = \{p\|_{TopEvent}, p\|_{G_1}, p\|_{G_2}, p\|_{E_5}\}$	$\{E_5\}$
$p\|_{E_6}$	$\mathbf{y}_6 = \{p\|_{TopEvent}, p\|_{G_1}, p\|_{G_2}, p\|_{G_3}, p\|_{G_4}, p\|_{G_5}, p\|_{E_6}, p\|_{G_6}, p\|_{E_9}\}$	$\{E_6, E_9\}$
$p\|_{E_7}$	$\mathbf{y}_7 = \{p\|_{TopEvent}, p\|_{G_1}, p\|_{G_2}, p\|_{G_3}, p\|_{G_4}, p\|_{G_5}, p\|_{E_7}, p\|_{G_6}, p\|_{E_9}\}$	$\{E_7, E_9\}$
$p\|_{E_8}$	$\mathbf{y}_8 = \{p\|_{TopEvent}, p\|_{G_1}, p\|_{G_2}, p\|_{G_3}, p\|_{G_4}, p\|_{G_5}, p\|_{E_8}, p\|_{G_6}, p\|_{E_9}\}$	$\{E_8, E_9\}$
$p\|_{E_9}$	$\mathbf{y}_9 = \mathbf{y}_6$	$\{E_6, E_9\}$
$p\|_{E_{10}}$	$\mathbf{y}_{10} = \{p\|_{TopEvent}, p\|_{G_1}, p\|_{G_2}, p\|_{G_3}, p\|_{G_4}, p\|_{G_5}, p\|_{E_6}, p\|_{G_6}, p\|_{E_{10}}\}$	$\{E_6, E_{10}\}$
$p\|_{E_{11}}$	$\mathbf{y}_{11} = \{p\|_{TopEvent}, p\|_{G_1}, p\|_{G_2}, p\|_{G_3}, p\|_{G_4}, p\|_{G_5}, p\|_{E_6}, p\|_{G_6}, p\|_{E_{11}}\}$	$\{E_6, E_{11}\}$
$p\|_{E_{12}}$	$\mathbf{y}_{12} = \{p\|_{TopEvent}, p\|_{G_1}, p\|_{G_2}, p\|_{G_3}, p\|_{G_4}, p\|_{G_5}, p\|_{E_6}, p\|_{G_7}, p\|_{E_{12}}\}$	$\{E_6, E_{12}\}$
$p\|_{E_{13}}$	$\mathbf{y}_{13} = \{p\|_{TopEvent}, p\|_{G_1}, p\|_{G_2}, p\|_{G_3}, p\|_{G_4}, p\|_{G_5}, p\|_{E_6}, p\|_{G_7}, p\|_{E_{13}}\}$	$\{E_6, E_{13}\}$
$p\|_{E_{14}}$	$\mathbf{y}_{14} = \{p\|_{TopEvent}, p\|_{G_1}, p\|_{G_2}, p\|_{G_3}, p\|_{G_4}, p\|_{G_5}, p\|_{E_6}, p\|_{G_7}, p\|_{E_{14}}\}$	$\{E_6, E_{14}\}$
$p\|_{E_{14}}$	$\mathbf{y}_{15} = \{p\|_{TopEvent}, p\|_{G_1}, p\|_{G_2}, p\|_{G_3}, p\|_{G_4}, p\|_{G_5}, p\|_{E_6}, p\|_{G_7}, p\|_{E_{15}}\}$	$\{E_6, E_{15}\}$
$p\|_{E_{16}}$	$\mathbf{y}_{16} = \{p\|_{TopEvent}, p\|_{G_1}, p\|_{G_2}, p\|_{G_3}, p\|_{G_4}, p\|_{G_5}, p\|_{E_6}, p\|_{G_7}, p\|_{E_{16}}\}$	$\{E_6, E_{16}\}$

$$\mathbf{y}_1 = \{p|_{TopEvent}, p|_{G_1}, p|_{G_2}, p|_{G_3}, p|_{G_4}, p|_{G_5}, p|_{E_6 \vee E_7 \vee E_8}, p|_{G_6}, p|_{E_9 \vee E_{10} \vee E_{11}}\}$$

$$\mathbf{y}_2 = \{p|_{TopEvent}, p|_{G_1}, p|_{G_2}, p|_{G_3}, p|_{G_4}, p|_{G_5}, p|_{E_6 \vee E_7 \vee E_8}, p|_{G_7}, p|_{G_8}, p|_{E_{14} \vee E_{15} \vee E_{16}}\}$$

$$\mathbf{y}_3 = \{p|_{TopEvent}, p|_{G_1}, p|_{G_2}, p|_{G_3}, p|_{G_4}, p|_{G_5}, p|_{E_6 \vee E_7 \vee E_8}, p|_{G_7}, p|_{E_{12} \vee E_{13}}\}$$

$$\mathbf{y}_4 = \{p|_{TopEvent}, p|_{G_1}, p|_{G_2}, p|_{E_4 \vee E_5}\}$$

$$\mathbf{y}_5 = \{p|_{TopEvent}, p|_{G_1}, p|_{E_3}\}$$

$$\mathbf{y}_6 = \{p|_{TopEvent}, p|_{E_1 \vee E_2}\}$$

$\mathcal{MCS}_1 = \{E_6, E_9\}$	$\mathcal{MCS}_{11} = \{E_6, E_{15}\}$	$\mathcal{MCS}_{21} = \{E_7, E_{12}\}$
$\mathcal{MCS}_2 = \{E_6, E_{10}\}$	$\mathcal{MCS}_{12} = \{E_6, E_{16}\}$	$\mathcal{MCS}_{22} = \{E_7, E_{13}\}$
$\mathcal{MCS}_3 = \{E_6, E_{11}\}$	$\mathcal{MCS}_{13} = \{E_7, E_{14}\}$	$\mathcal{MCS}_{23} = \{E_8, E_{12}\}$
$\mathcal{MCS}_4 = \{E_7, E_9\}$	$\mathcal{MCS}_{14} = \{E_7, E_{15}\}$	$\mathcal{MCS}_{24} = \{E_8, E_{13}\}$
$\mathcal{MCS}_5 = \{E_7, E_{10}\}$	$\mathcal{MCS}_{15} = \{E_7, E_{16}\}$	$\mathcal{MCS}_{25} = \{E_4\}$
$\mathcal{MCS}_6 = \{E_7, E_{11}\}$	$\mathcal{MCS}_{16} = \{E_8, E_{14}\}$	$\mathcal{MCS}_{26} = \{E_5\}$
$\mathcal{MCS}_7 = \{E_8, E_9\}$	$\mathcal{MCS}_{17} = \{E_8, E_{15}\}$	$\mathcal{MCS}_{27} = \{E_3\}$
$\mathcal{MCS}_8 = \{E_8, E_{10}\}$	$\mathcal{MCS}_{18} = \{E_8, E_{16}\}$	$\mathcal{MCS}_{28} = \{E_1\}$
$\mathcal{MCS}_9 = \{E_8, E_{11}\}$	$\mathcal{MCS}_{19} = \{E_6, E_{12}\}$	$\mathcal{MCS}_{29} = \{E_2\}$
$\mathcal{MCS}_{10} = \{E_6, E_{14}\}$	$\mathcal{MCS}_{20} = \{E_6, E_{13}\}$	

TE occurrence formula:

$$\bigvee_{i=1}^{29} \mathcal{MCS}_i$$

Agenda

Related Work

Computation of MCS/MPS is an NP-hard problem (in general)

- Two main approaches, depending on how the FT is analyzed
  - Top-down
  - Bottom-up
- MOCUS, CARA, DICOMICS, FATRAM, MICSUP...

Other model transformation

- To Coloured PNs, or Reverse PNs: Reachability graph, reachability markings
  - NP-hard problem, with exponential space requirements
- To Reliability Block Diagrams
- To BDDs
  - Its computation may fail and does not avoid the exponential problem

Agenda

Conclusions

- Computation of MCS/MPS of a coherent Fault Tree performed in linear time, by model transformation into a Petri net
- Constraints applied:
  - Logic restricted to AND/OR formulae
  - Only basic, undeveloped, external, and intermediate events considered

Conclusions

- Computation of MCS/MPS of a coherent Fault Tree performed in linear time, by model transformation into a Petri net
- Constraints applied:
  - Logic restricted to AND/OR formulae
  - Only basic, undeveloped, external, and intermediate events considered

Future work

- Implemented as module of PeabraiN tool (done!)
- Better characterize coherent FT whose MCS/MPS are solvable in polynomial time
- Compare to existing approaches
- Do the maths to avoid model transformation

RIASC

On Qualitative Analysis of Fault Trees Using Structurally Persistent Nets

Ricardo J. Rodríguez
rj.rodriguez@unileon.es

Research Institute of Applied Sciences in Cybersecurity
University of León, Spain

June 10, 2015
XXIII Jornadas de Concurrencia y Sistemas Distribuidos
Málaga (Spain)

To appear in IEEE Trans. on Systems, Man, and Cybernetics: Systems
doi: 10.1109/TSMC.2015.2437360