Fault Tree Linking versus Event Tree Linking Approaches: A Mathematical and Algorithmic Reconciliation

Dr. O. Nusbaumer, Switzerland
Prof. Dr. A. Rauzy, France

- Fault Tree Linking (FTL) vs. Event Tree Linking (ETL)
- Quantification issues
- Strong and weak equivalence between models
- Reconciliation between FTL and ETL
  - For coherent (and truncated) models
  - For non-coherent models
- Conclusions and perspectives

Fault Tree Linking vs. Event Tree Linking

- **In PSA, two different quantification techniques have evolved**

Fault Tree Linking (FTL)	Event Tree Linking (ETL)
RiskSpectrum	RiskMan
FTREX / CAFTA	…
FinPSA	
…	

Minimal Cutsets (MCS)

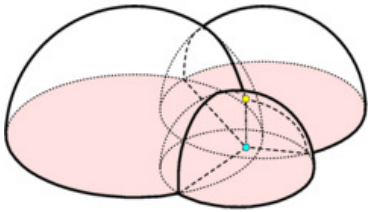
Sum of Disjoint Products (SDP)

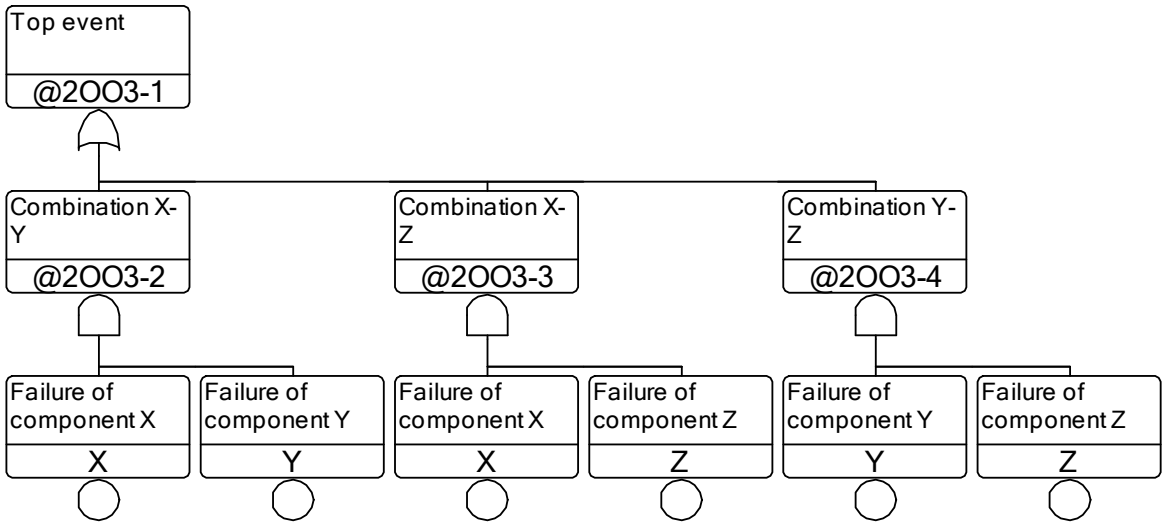
Fault Tree Linking vs. Event Tree Linking

- **Fault Tree Linking (FTL)**
  - Uses large Fault Trees to model defense barriers
  - Uses small Event Trees to model the accident progression
  - Widely used worlwide
  - Performs a coherent (monotone) approximation of the model
- **Event Tree Linking (ETL)**
  - Uses relatively large Event Trees to represent system states
  - Aims to make Function Events independent one another
  - Sequences are summed up in Fault Trees

Quantification issues

- **Assume a 2-out-of-3 system (in FTL)**





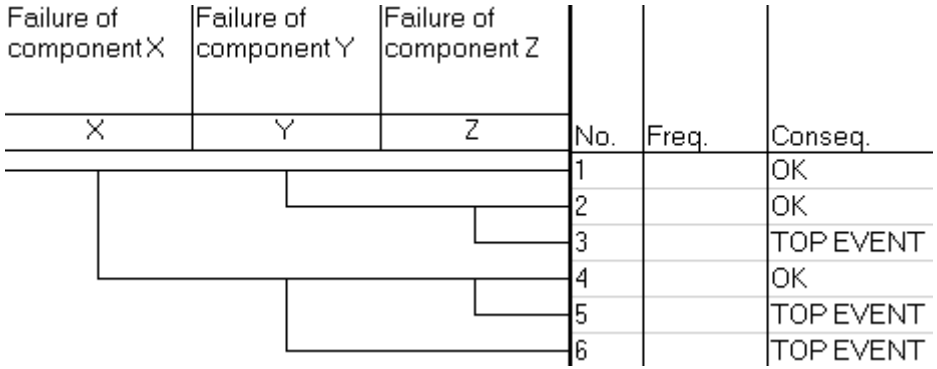
→ ~~$P(top) = 3 \cdot p^2$~~ → $P(top) = 3 \cdot p^2 - 2 \cdot p^3$

First order („rare event“) approximation

$$p(F) = \sum_{1 \le i \le n} p(\pi_i) - \sum_{1 \le i_1 \le i_2 \le n} p(\pi_{i_1} \cdot \pi_{i_2}) + \ldots + (-1)^p \cdot \sum_{1 \le i_1 \le \ldots \le i_p \le n} p(\pi_{i_1} \cdot \ldots \cdot \pi_{i_p}) + \ldots + (-1)^n \cdot p(\pi_{i_1} \cdot \cdots \cdot \pi_{i_n})$$

Quantification issues

- **Now assume a 2-out-of-3 system (in ETL)**

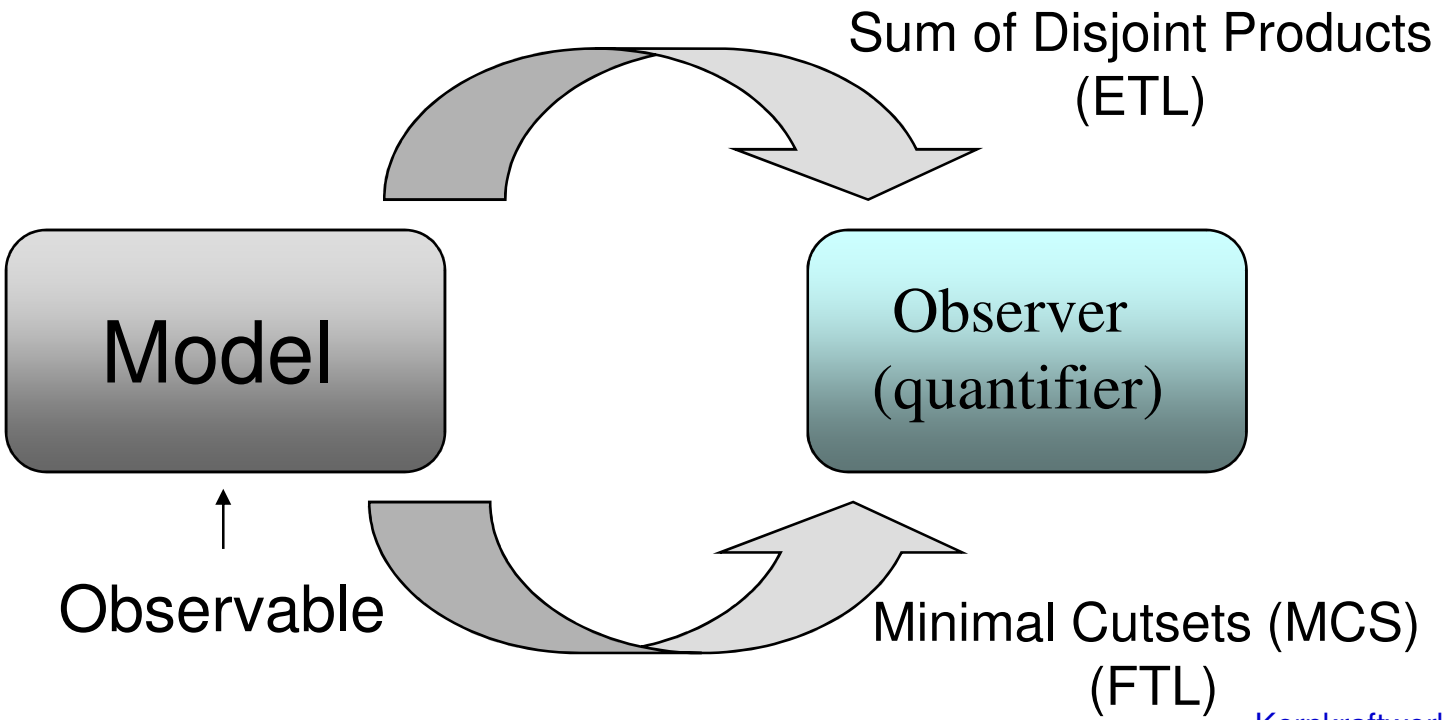


→ $P(top) = (1-p)\cdot p^2 + p\cdot(1-p)\cdot p + p^2 = 3\cdot p^2 - 2\cdot p^3$

Observation: Rare Event Approximation in FTL ←→ "1-p(x)" in ETL !

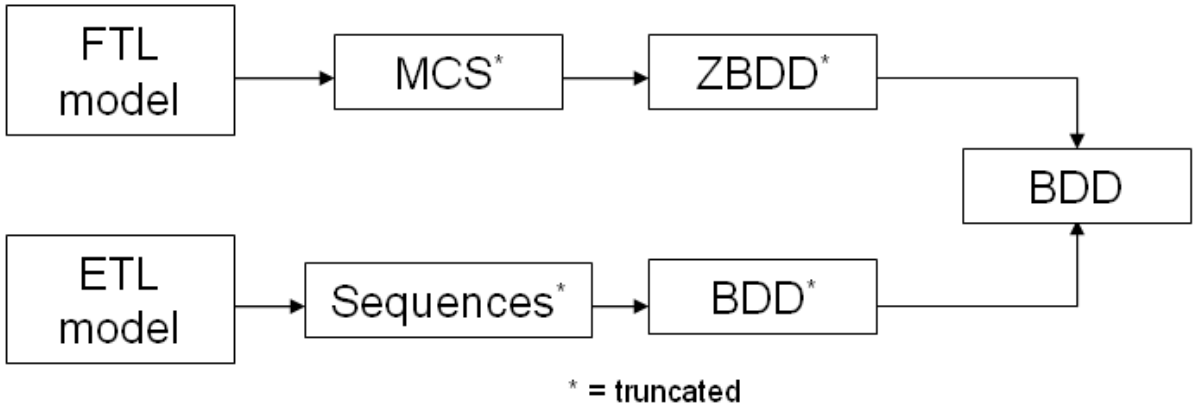
Strong and weak equivalence between models

- **Two objects under study can be considered as equivalent if they cannot be distinguished with the observation means at hand !**



Reconciliation between FTL and ELT

- **In the 90‘s, Rauzy introduced the Binary Decision Diagrams (BDD) in the reliability field**
- **Immediately implemented for the quantification of the small, independent Fault Tree in ETL models**
- **Later on, implemented on small to medium size FTL models (still very difficult on large models)**



Strong and weak equivalence between models

- **Strong equivalence**
  - ☼ Two models are strongly equivalent if they agree on states whose probability is bigger than the given cutoff, i.e. if they cannot be distinguished by means of a Sum of Disjoint Products algorithm
- **Weak equivalence**
  - ☼ Two models are weakly equivalent if they agree on Minimal Cutsets whose probability is bigger than the given cutoff, i.e. if they cannot be distinguished by means of a Minimal Cutsets algorithm

Strong and weak equivalence between models

Let F and G be two Boolean formulas and χ be a cutoff value:

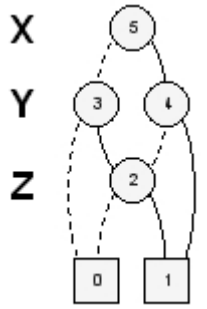
- F **strongly** entails G at precision χ if for any minterm π such that $p(|\pi|) \geq \chi$, if $\pi \in F$ then $\pi \in G$. F and G are strongly equivalent at precision χ, if both F strongly entails G and G strongly entails F at precision χ.
- F **weakly** entails G at precision χ if for any minterm π such that $p(|\pi|) \geq \chi$, if $\pi \in G$ then there exists a minterm $\rho \leq \pi$ such that $p(|\rho|) \geq \chi$ and $\rho \in F$. F and G are weakly equivalent at precision χ, if both F weakly entails G and G weakly entails F at precision χ.

Reconciliation for Coherent Models

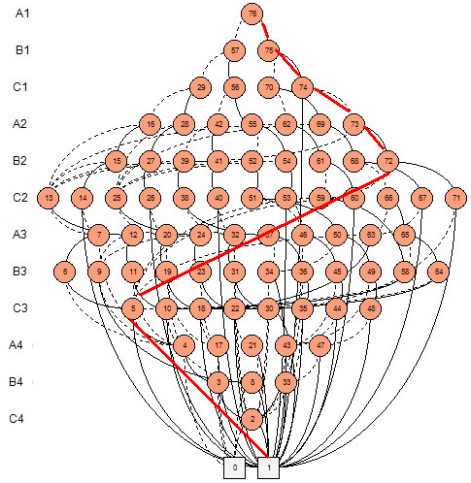
- **Since *F = MCS(F)* [Rauzy 92], can we solve coherent models exactly from their Minimal Cutsets (MCS) ?**
  - ☼ YES
- **Why ?**
  - ☼ Since *F = MCS(F)* for any coherent Function *F*, it follows that *BDD(MCS(F)) = SDP(F) = p(F)*
  - ☼ Example for the 2-out-of-3 system:

    $MCS(F) = X \cdot Y + X \cdot Z + Y \cdot Z$

    $BDD(F) = X \cdot Y + X \cdot (1-Y) \cdot Z + (1-X) \cdot Y \cdot Z \rightarrow p(top) = 3p^2 - 2p^3$!



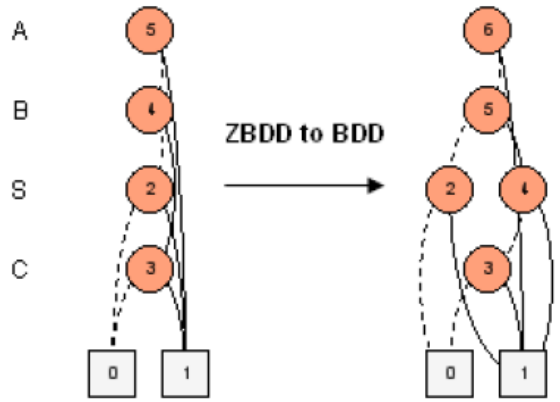
Reconciliation for Coherent Models



- **What if the MCS are not complete ?**
  - ☼ Assume truncation level c
  - ☼ Intuitively, missing MCS won't "distort" the remaining path of the BDD
  - ☼ The quantification will be exact up to a precision c
  - ☼ Moreover, the following logical equivalence holds:
    - $MCS_{\geq\chi}(F) \equiv MCS(\ SDP_{\geq\chi}(F)\ )$
    - It asserts that truncated MCS and truncated SDP agree on failure scenarios
  - ☼ Large probability values (e.g. seismic PSA) is not a problem

Reconciliation for Coherent Models

- **On coherent models, the exact probability can be obtained from the calculated MCS**
  - ☼ By either applying the full Sylvester-Poincaré Development or calculating a BDD straight from the model or
  - ☼ By calculating a BDD from the MCS (can be CPU costly)




```
zbdd2bdd(F: ZBDD): BDD
  if F=0
      return 0
  else if F=1
      return 1
  else F=ZBDD(x, F1, F0)
      R0 = zbdd2bdd(F0)
      R1 = bdd-or(R0, zbdd2bdd(F1))
      return create-bdd-node(x, R1, R0)
```

Reconciliation for Non-Coherent Models

- **Most models use “Negative Logic” (NOT-logic) making models non-coherent**
- **NOT-logic uses non-coherent cognitive**
  - ☼ e.g. NOR, XOR, NAND, ¬X (NOT)
- **A fault tree is non-coherent when both failure and success can cause the top event to occur**
- **Indicates how the lack of an event's occurrence can cause the top event to occur**
  - ☼ If the NOT-logic can be eliminated from the fault tree, the fault tree is coherent, if not, it is not.

Reconciliation for Non-Coherent Models

- **Typical uses of NOT-logic in PSA**
  - Exclude unwanted or impossible fault combinations (e.g. maintenance rules)
  - Taking credit of failures
  - “IF-THEN-ELSE” (ITE) operations
  - Taking credit of success branches in Event Trees (1-p(x))
  - Conditional adaptation of success criteria
  - Exchanging basic events (specific to CAFTA)
  - … and more weird things !

Reconciliation for Non-Coherent Models

- **We decided to start an international survey on the uses of NOT-logic in order to categorize them**
- **Basically, two questions were asked:**
  - ☼ (Why) do you use negation ?
  - ☼ How specific issues are modelled using negation ?
- **Participating countries included Sweden, Finland, France, Germany, Switzerland, Spain, USA**

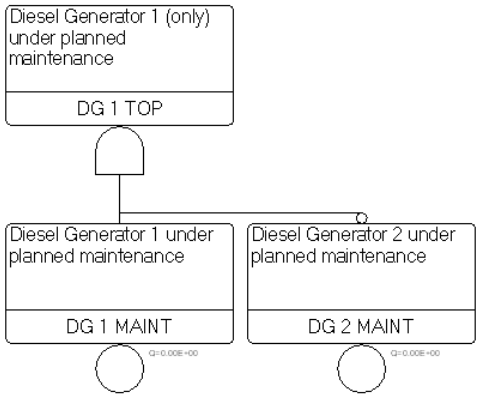
Reconciliation for Non-Coherent Models

- **The following 3 categories were identified according to their mathematical characteristics and treatment by quantification engines**

  1. Exclusion of forbidden or impossible configuration
  2. Conditional adaptation of success criteria (ITE operation)
  3. Delete terms

Reconciliation for Non-Coherent Models

- **Case 1: Exclusion of forbidden or impossible configuration**
  - ☼ Assume 2 systems X_1 and X_2
    - Failure probabilities $p(X_1)$, resp. $p(x_2)$
    - Unavailabilities $u(x_1)$, resp. $u(x_2)$
  - ☼ Then the exact mean unavailability yields:

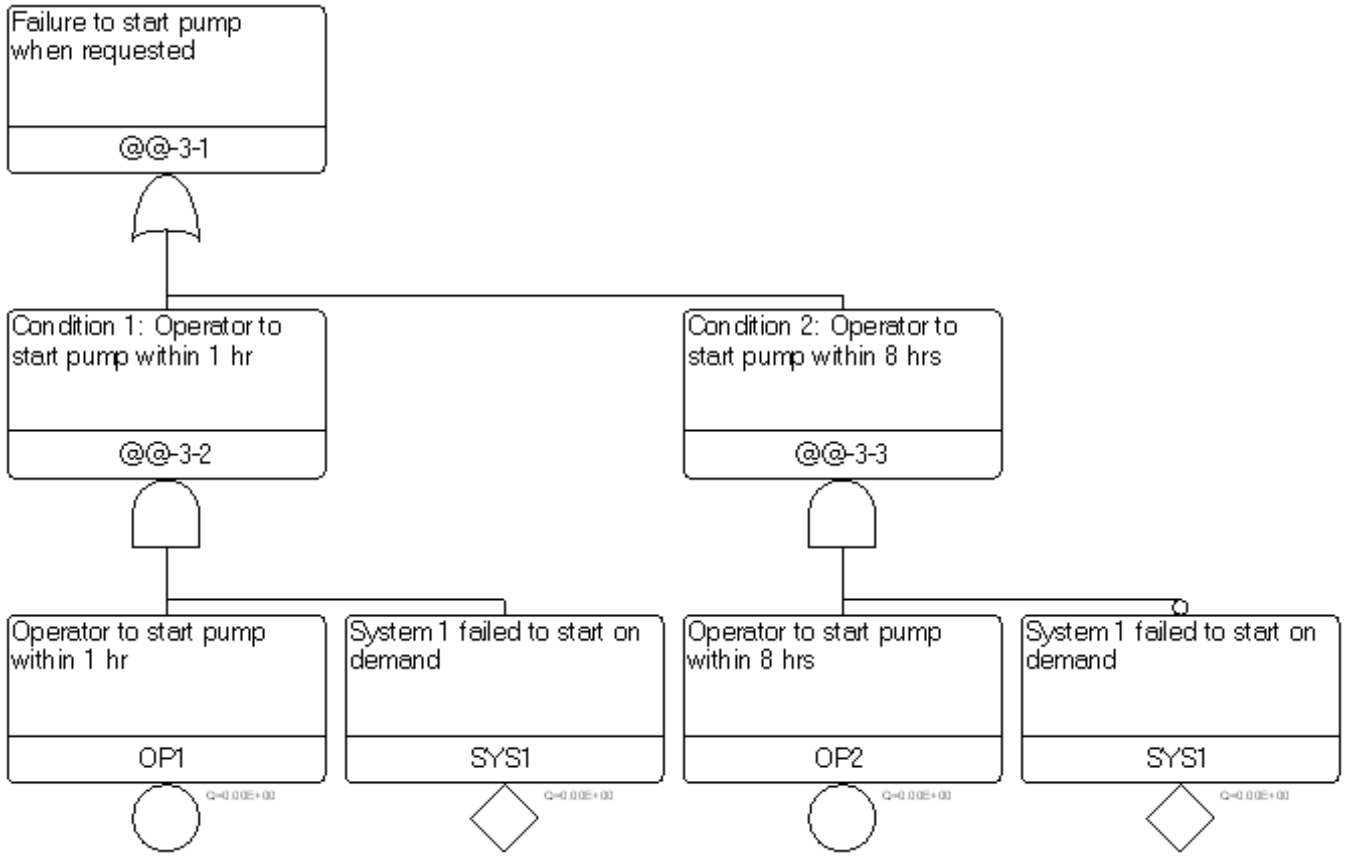


$$u(x_1)\cdot p(x_2)+u(x_2)\cdot p(x_1)+\left[1-u(x_1)-u(x_2)\right]\cdot p(x_1)\cdot p(x_2)$$

  - ☼ The result is not a Boolean expression
  - ☼ Both FTL and ETL to be solves with „configuration management“, i.e. add discrete model states (“configurations”) together

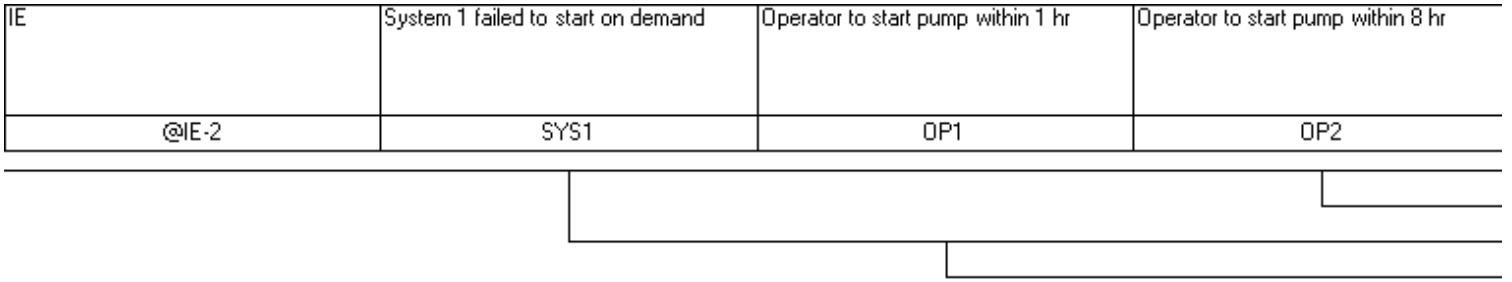
Reconciliation for Non-Coherent Models

- **Case 2: Conditional Adaptation of Success Criteria (ITE)**



Reconciliation for Non-Coherent Models

- **Case 2: Conditional Adaptation of Success Criteria (ITE)**
  - ☼ Yields non-coherent equations (unfortunately) and cannot be solved by *BDD(ZBDD(MCS))*
  - ☼ Equivalent to asking the question upwardly in the Event Tree



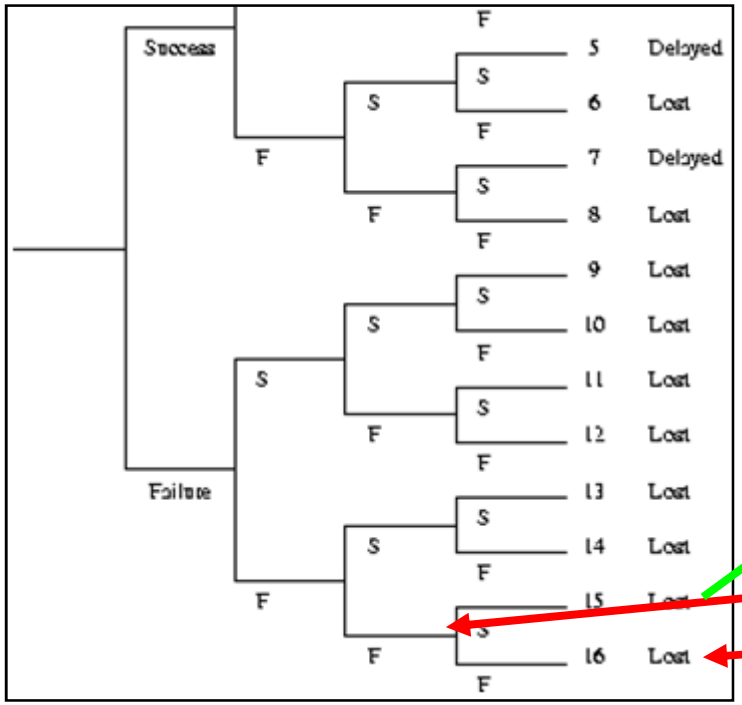
  - ☼ Use to “retrieve” non-queried conditions in the Event Tree
  - ☼ → Treatment of success branches in Event Trees

Reconciliation for Non-Coherent Models

- **Case 3: Delete Term**
  - By “delete term” we mean here removing one or many MCS or basic events from the overall result
  - A typical example is when the modeller wants to get rid of specific basic events in Fault Tree
  - Similar to truncation on a FTL framework, deleting basic events or MCS does not yield incoherent results.
  - The resulting model is, despite appearances, coherent, and the previous results hold.

Reconciliation for Non-Coherent Models

- **ETL practitioners often claim that MCS-based algorithm cannot quantify success branches in Event Trees (non-coherent)**
  - ☼ This is partially true (even if no current quantifier has implemented it)



This sequence is non-coherent, but can be analytically calculated from the two other ones !

This sequence is coherent !

This sequence is coherent !

Reconciliation for Non-Coherent Models

- **Algorithm to quantify success branches in Event Trees:**

AssessSequence(*S*: sequence) : real
if S contains only positive events
then calculate *MCS(S)*, evaluate *p(MCS(S))* and return it
else $S = -Q.S'$
$p_1 =$ AssessSequence(S')
$p_2 =$ AssessSequence($Q.S'$)
return $p_1 - p_2$

Reconciliation for Non-Coherent Models

- **Excellent ! We can numerically solve Success Paths using MCS exactly**
- **… Even better, Case 2 (“Conditional Adaptation of Success Criteria”) can also be solved exactly, by**
  - ☼ Detecting the ITE structure in the Fault Trees
  - ☼ Rewriting the Event Tree to “ask” non-queried conditions
  - ☼ Bringing the Fault Tree to a coherent structure

Conclusion and perspectives

- **What we have learnt**
  - **Equivalence relations** between models have been **formally defined**
  - **Coherent models can be solved exactly** using MCS at precision c
    - Large probability values (e.g. seismic PSA) is not a problem
  - Incoherent models can be brought to a coherent form
  - **Success paths in Event Trees can be calculated exactly**
  - FTL and ETL are "**weakly equivalent**"
  - **The discrepancies observed for years between FTL and ETL models are probably due to the use and treatment of negative logic**

Conclusion and perspectives

- **The theoretical and algorithmic ingredients are now available to develop the <u>“bridging” software</u>**