The Quest Towards Analytical Solutions of Linked Fault Tree Models using Binary Decision Diagrams(*)

Olivier Nusbaumer

- Motivation and issues with current PSA softwares
- Binary Decision Diagrams (BDD) as an alternative
- Research and development at KKL and ETH Zurich
- Results, insights and outlook
- Presentation of the *NeuralSpectrum* Software

(*) From the PhD thesis: “Analytical Solutions of Linked Fault Tree Probabilistic Risk Assessments using Binary Decision Diagrams with Emphasis on Nuclear Safety Applications”

Motivation and issues with current PSA softwares

- **The (Swiss) NPP have to submit best-estimate, plant-specific PSA models to the regulatory authority**
  - Level 1 PSA: Calculation of the Core Damage Frequency (CDF)
  - Level 2 PSA: Calculation of the Plant Damage State (PDS) frequencies and associated radiological consequences
  - For internal events, area events and external events
  - For full power and shutdown conditions

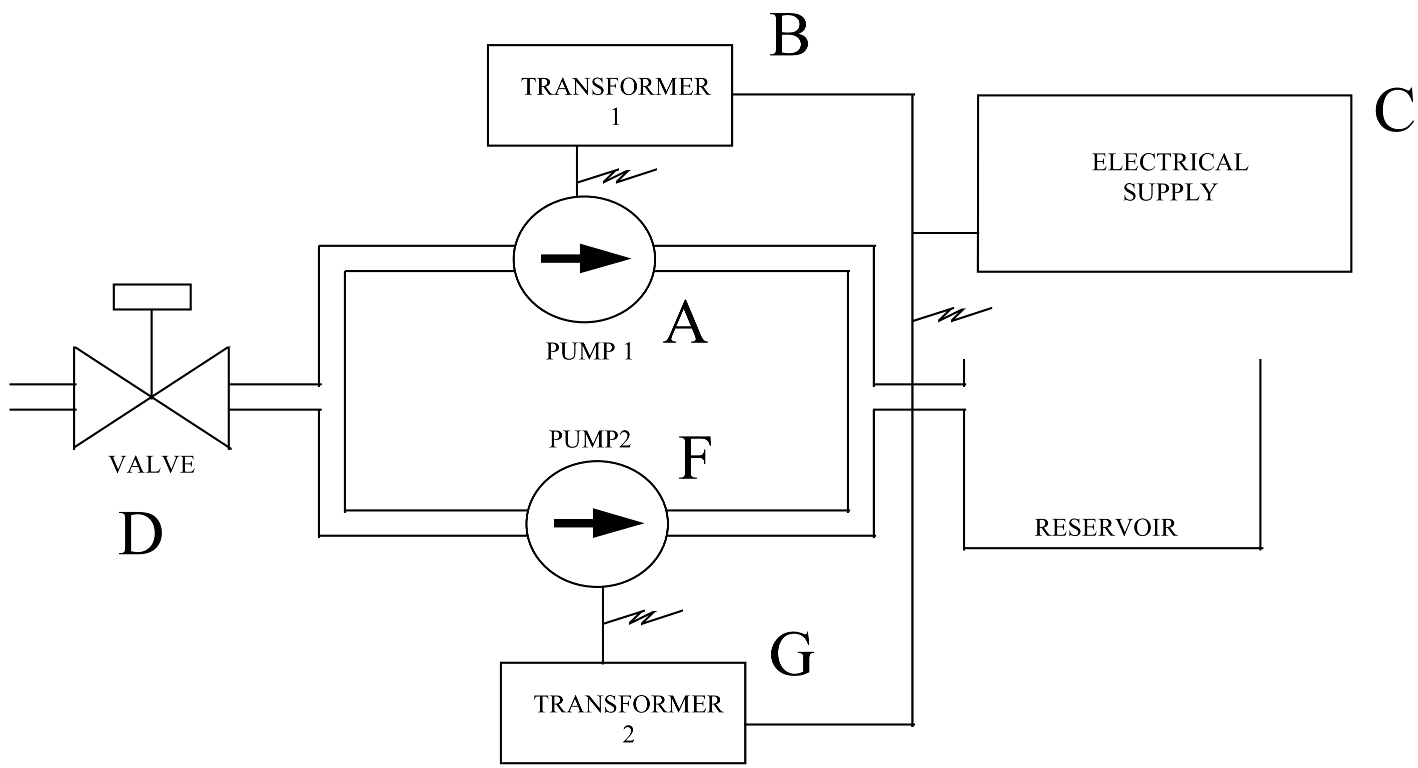
Motivation and issues with current PSA softwares

- **Worldwide Fault Tree Analyses (FTA) are performed with codes that produce "questionable" results**
  - Rare event approximation…
  - … but not only !
- **Resulting risk importance measures are even more questionable (conservative or optimistic, no one can know)**

$$RIF_{x_i} = \frac{CDF(x_i = 1)}{CDF}$$

Motivation and issues with current PSA softwares

- **What is the mean unreliability of the system's function, based on individual Basic Events probabilities ?**

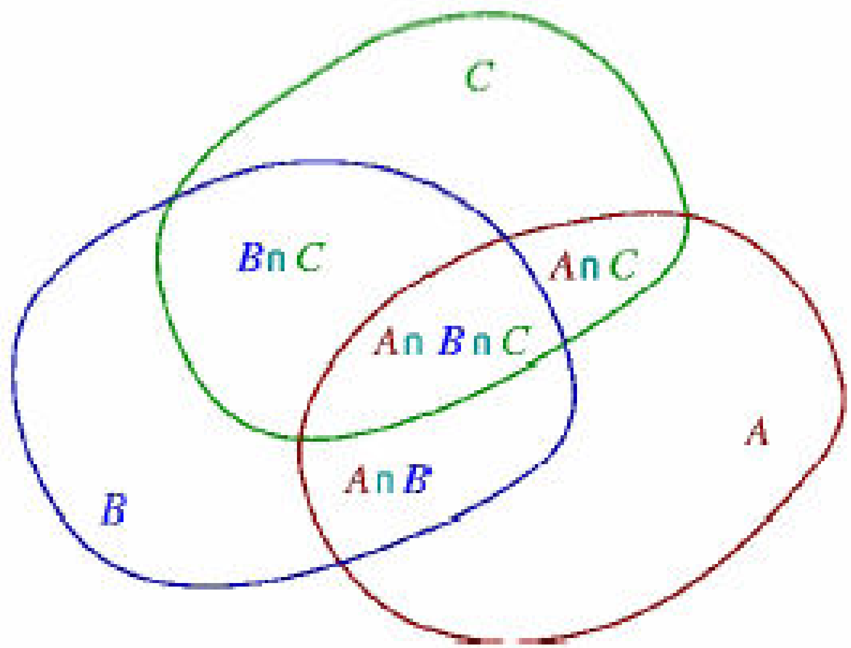


$$P_{top} = D + C + AF + AG + BF + BG$$

Motivation and issues with current PSA softwares

- **The rare event approximation (Moivre‘s equation)**

$$|A_1 \cup \ldots \cup A_p| = \sum_{1 \le i \le p} |A_i| - \sum_{1 \le i_1 < i_2 \le p} |A_{i_1} \cap A_{i_2}| + \ldots + (-1)^{p-1} |A_1 \cap \ldots \cap A_p|$$

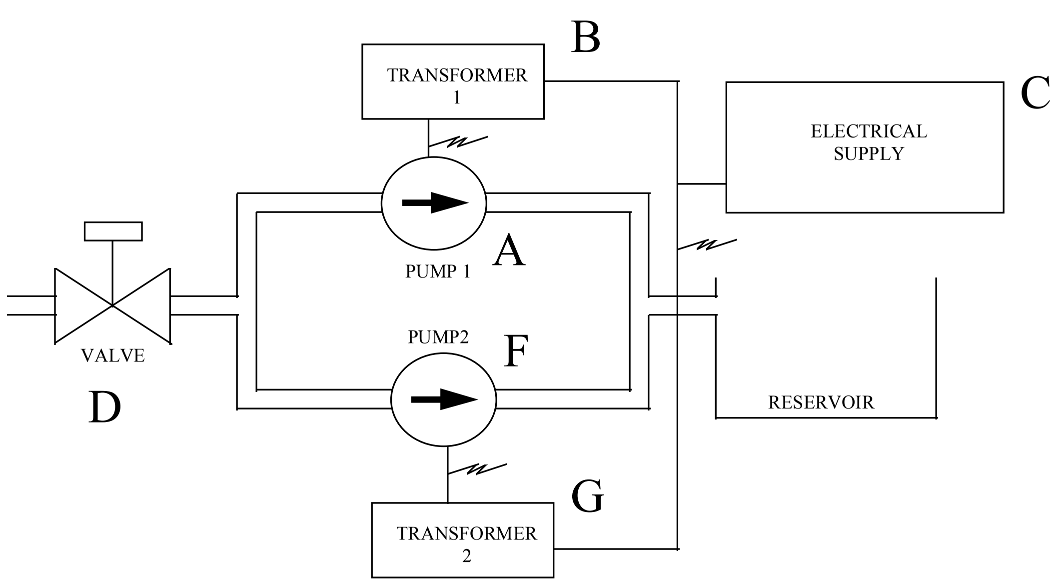


Motivation and issues with current PSA softwares

- **Analytically correct result yields:**

$$P_{top} = (d + f + g + c - d f - d g - d c - f g - f c - g c + d f g + d f c + d g c + f g c - d f g c) (a + b + c + d - a b - a c - a d - b c - b d - c d + a b c + a b d + a c d + b c d - a b c d)$$

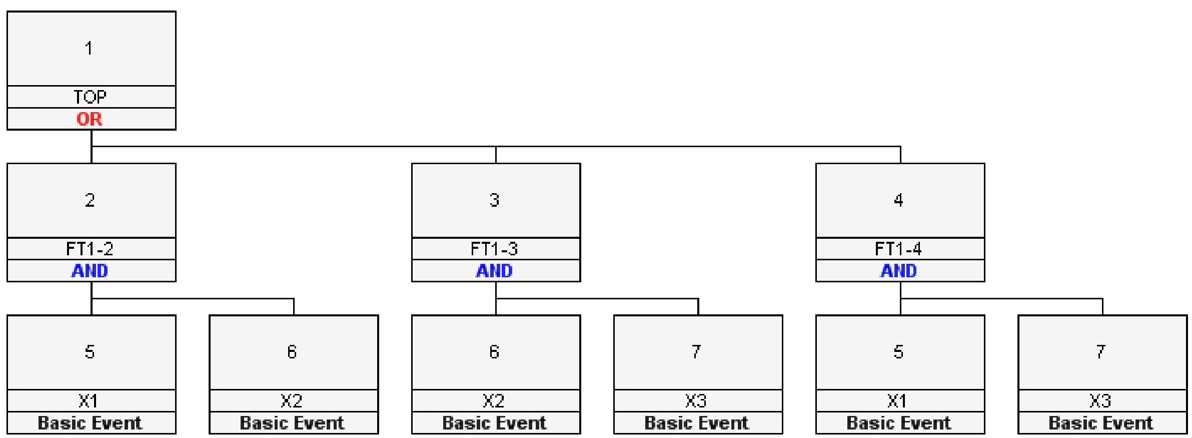
$$= [c - a (-1 + b) (-1 + c) (-1 + d) + b (-1 + c) (-1 + d) + d - c d] \cdot [f - c (-1 + d) (-1 + f) (-1 + g) + d (-1 + f) (-1 + g) + g - f g]$$



Motivation and issues with current PSA softwares

- **Consider a « 2 out of 3 system » given by the Boolean equation:**

$$T = (x_1 \wedge x_2) \vee (x_1 \wedge x_3) \vee (x_2 \wedge x_3) \qquad p(x_1) = p(x_2) = p(x_3) := q$$



$$p(T) \neq p(x_1)p(x_2) + p(x_1)p(x_3) + p(x_2)p(x_3) = 3 \cdot q^2$$

$$p(T) = p(x_1)p(x_2) + p(x_1)\underbrace{(1 - p(x_2))}_{Success}p(x_3) + \underbrace{(1 - p(x_1))}_{Success}p(x_2)p(x_3) = 3q^2 - 2q^3$$

Motivation and issues with current PSA softwares

- **Other issues include:**
  - Wrong treatment of negative logic
    (e.g. forbidden maintenance unavailabilities according to TechSpec)
  - Quantification cutoff (typically 1E-12 to 1E-14)
  - Interpretation of risk importance measures of components, systems and safety divisions (RIF, FV, etc.)
  - Treatment of exchange events
- **Advanced PSA models include HRA, CCF, seismic and phenomenological events, where failure probabilities approach 1**
- **It is accepted that current quantification tools have reached their limits [Rauzy, 2001]**

Motivation and issues with current PSA softwares

- **Develop a new PSA quantification methodology that:**
  - Overcomes the deficiencies of the rare approximation, i.e. credit success branches, calculate the rare event up to **infinite order**
  - Yields a correct evaluation of Risk Importance Factors (RIFs)
  - Support the treatment of negative logic
  - Do not apply cutoff !
  - Improve calculation speed and result consistency

Binary Decision Diagrams (BDD) as an alternative

- **Shannon expansion**

$$x \rightarrow y_0, y_1 := (x \wedge y_0) \vee (\overline{x} \wedge y_1) := ite(x, y_0, y_1)$$

- **Shannon expansion of *t* with respect to *x***

$$t = x \rightarrow t[1/x], t[0/x] \Rightarrow t = (x \wedge t[1/x]) \vee (\overline{x} \wedge t[0/x])$$

  - *t[0/x]* and *t[1/x]* both contain one less variable than the expression *t*
  - We can <u>recursively</u> find ITEs up to the basic elements *0* (false) and *1* (true)

Binary Decision Diagrams (BDD) as an alternative

- **Example for the „2 out of 3“ system *t* = *AB* + *BC* + *AC***

$t = A \rightarrow (t_0, t_1)$

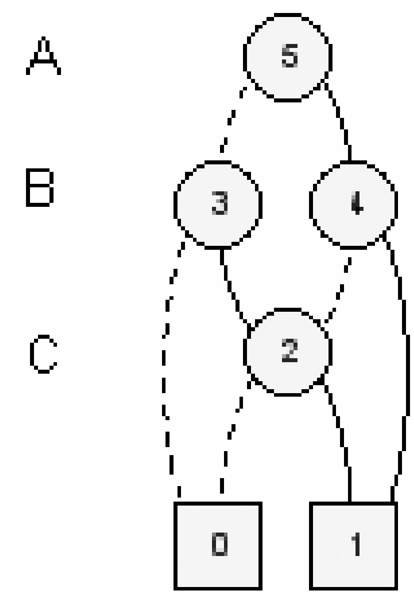
☼ $t_0 = B \rightarrow (0, t_{01})$

☼ $t_1 = B \rightarrow (1, t_{10})$

☼ $t_{01} = C \rightarrow (1, 0)$

☼ $t_{10} = C \rightarrow (1, 0)$

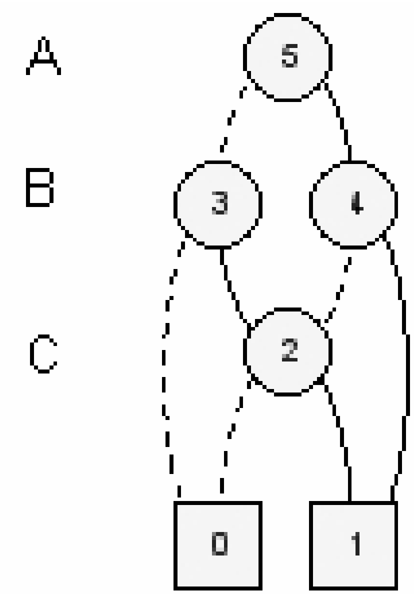
=> $t = A \rightarrow (B \rightarrow (0, C \rightarrow (1, 0)), B \rightarrow (1, C \rightarrow (1, 0)))$



Binary Decision Diagrams (BDD) as an alternative

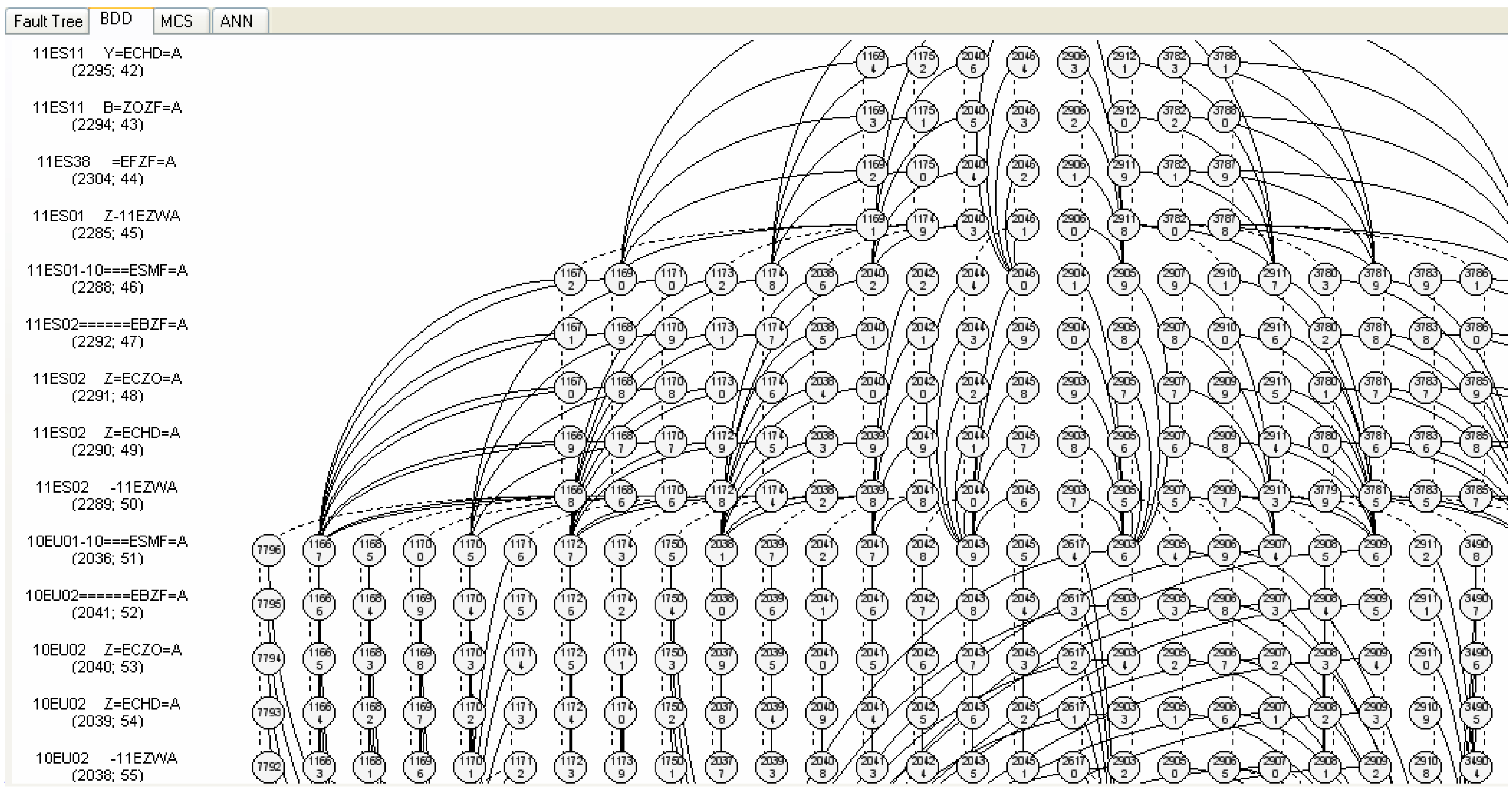
- **<u>Canonical formulation of Boolean equations !</u>**

- **Example:** $P_{t=true} = AB + A\ (1-B)\ C + (1-A)\ B\ C$



Binary Decision Diagrams (BDD) as an alternative

- **BDD complexity is not related to the number of prime implicants of the encoded formula**
- **This small BDD (37’620 nodes) encodes a total of 10^9 cutsets**

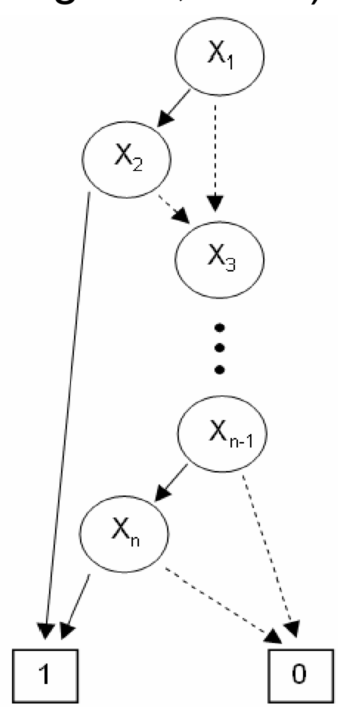


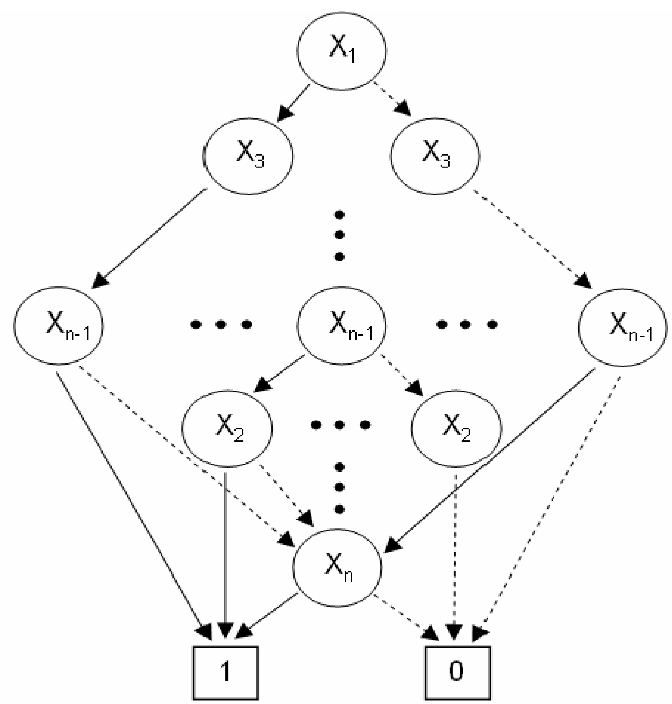
Binary Decision Diagrams (BDD) as an alternative

- **Let's have a closer look at it !**
  - ☼ HPCS System of the Leibstadt NPP

Binary Decision Diagrams (BDD) as an alternative

- **Impact of variable order on BDD size**
  - ☼ From linear ☺ to exponential ☹
  - ☼ ☹ Finding the best order is of **NP-Complete complexity** (Bollig / Wegener, 1996)

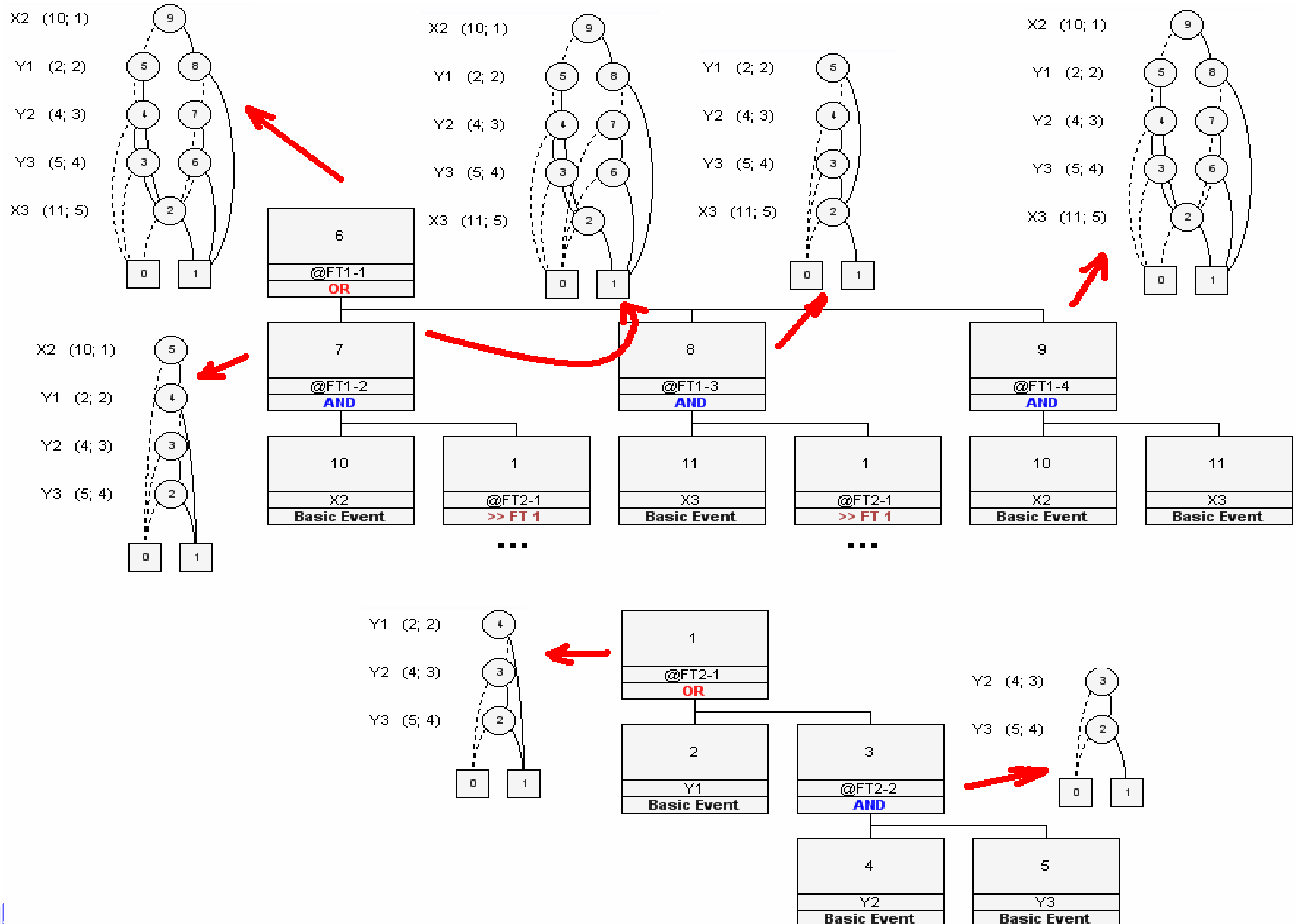




Binary Decision Diagrams (BDD) as an alternative

- **Previous Work**
  - BDD have been implemented in the early 90's for Integrated Circuits (IC) checking and CPU optimization (16 and 32 bits)
  - Some attempts to convert small to medium size models (typically with a few hundreds Basic Events) have succeeded
  - All attempts with more Basic Events (>>1000) have failed due to the exponential growth in complexity ("BDD blow up")

Binary Decision Diagrams (BDD) as an alternative

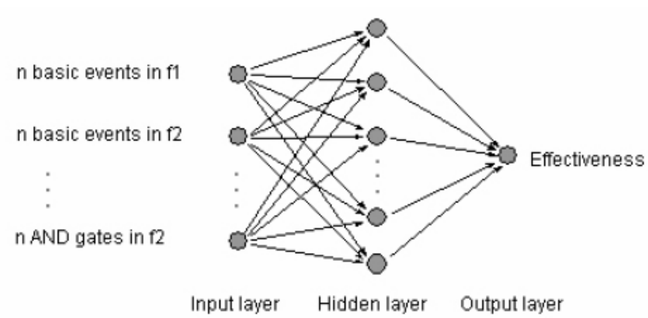


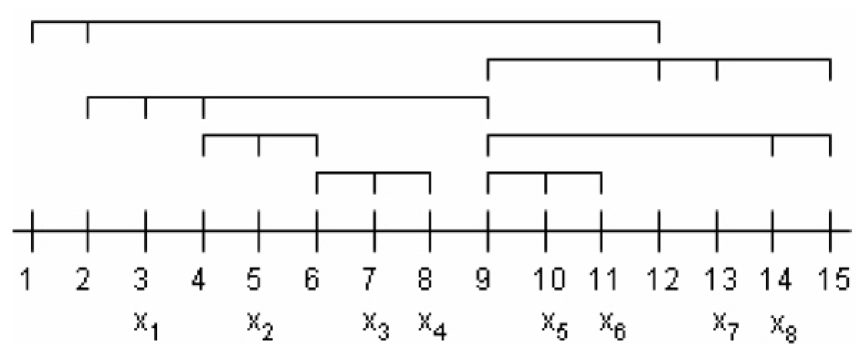
Research and development at KKL and ETH Zurich

- **Development of Fault Tree to BDD conversion engine**
- **Development of statistical measures**
- **Analysis of Fault Tree model pre-processing (rewriting) techniques**
  - Basic Event occurrence based ordering
  - Weights fan-out pre-processing →

$$W(v) = \begin{cases} 1 & \text{for basic events} \\ \sum_i W(v_i) & \text{for gates} \end{cases}$$

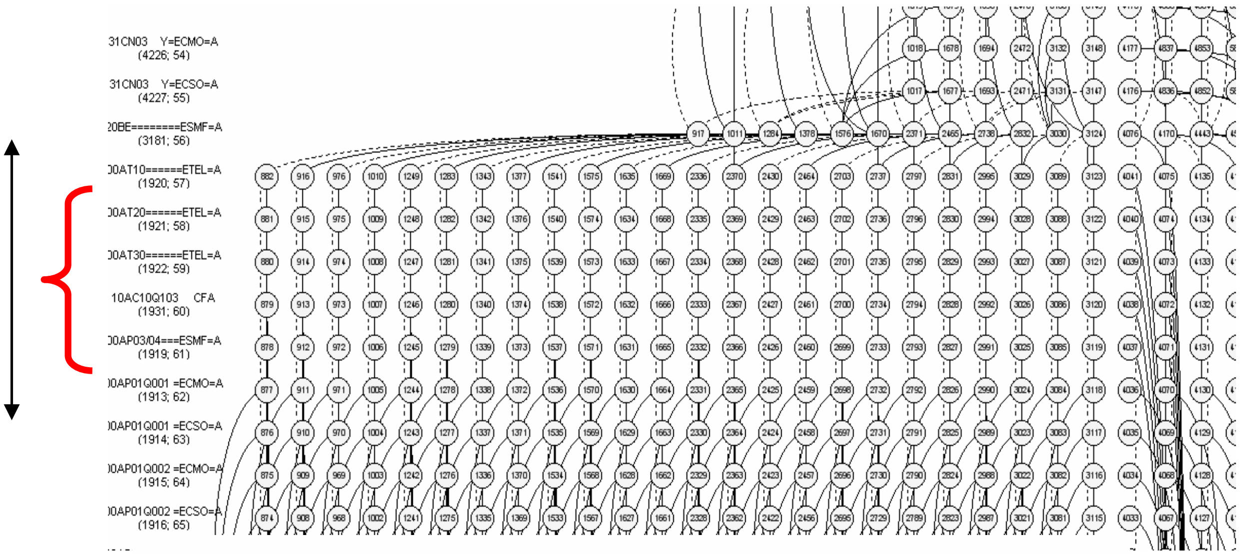
  - Hypergraph optimization techniques
  - Artificial Neural Network





Research and development at KKL and ETH Zurich

- **Development of dynamic optimization techniques (e.g. Sifting, p-cut variable arrangement)**
- **Development of Group-Sifting for FTA**

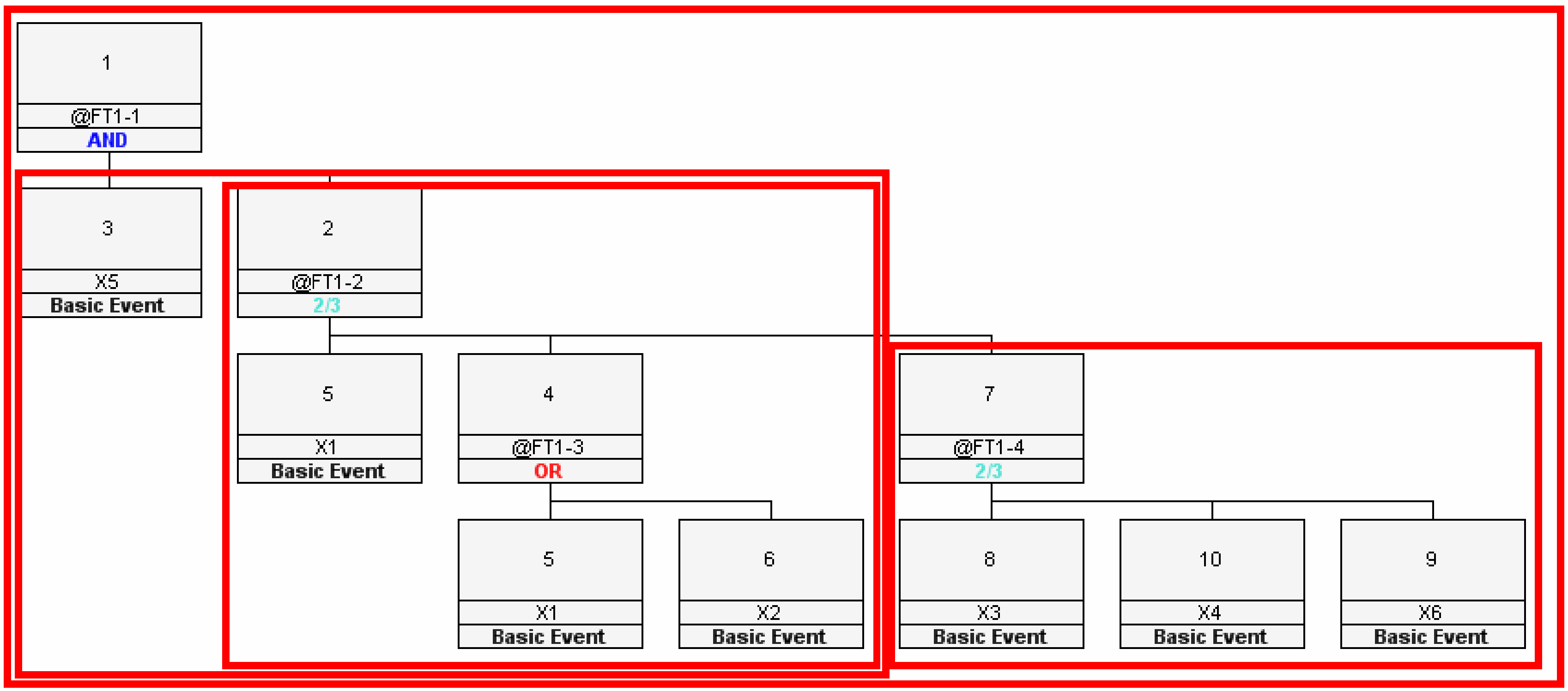


	DFLM	Regular sifting	Group-sifting
HPCS	6545	3204	497
LPCS	206'503	40'656	7763
RHR/A	306'339	99'945	11'948

(*) number of nodes in BDD

Research and development at KKL and ETH Zurich

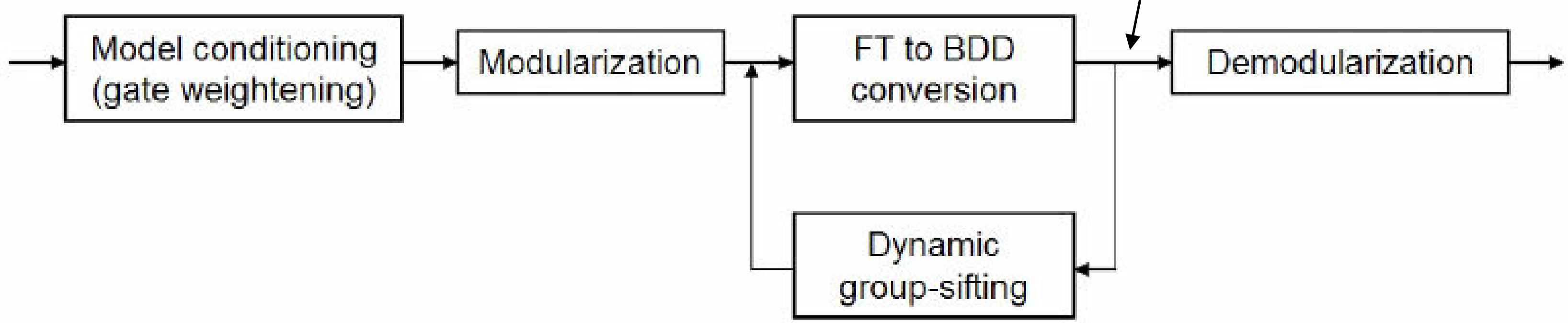
- **Development and analysis of modularization techniques**
  - Occurrence vectors and detection criteria



Research and development at KKL and ETH Zurich

- **Dynamic Group-Sifting using modularization (*<u>DGSM</u>*)**

$$size_{BDD_R} > \left(\left(\frac{size_{BDD_R}}{N} \right)^{\alpha} + 1 \right) \times \left(size_{BDD1} + size_{BDD_2} \right)$$

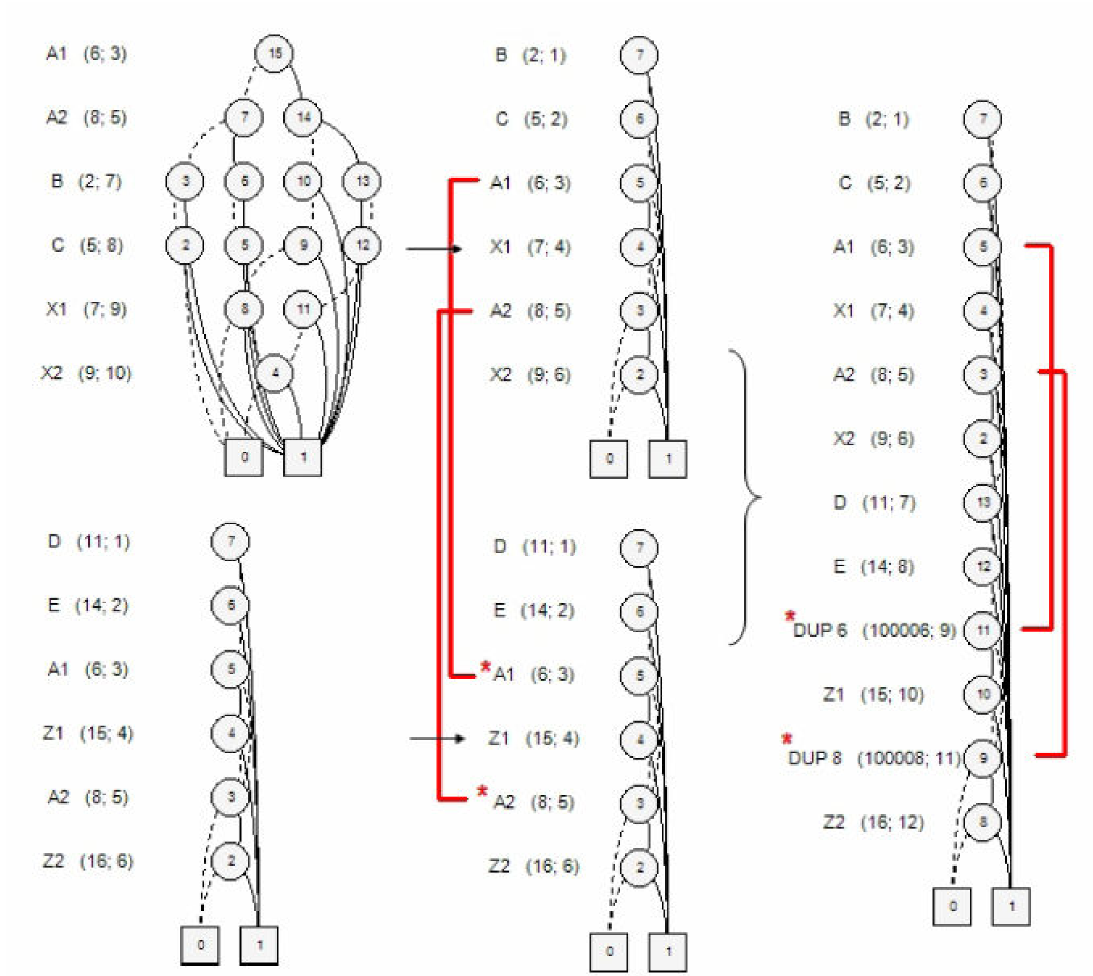


Systems	DFLM	Dynamic group-sifting	Dynamic group-sifting using modularization
HPCS	6545	761	497
LPCS	206'503	3050	3053
RHR/A	306'339	3117	3117
RHR/B	impossible	52'447	21'177
ECCS	impossible	impossible	1'781'100 (CPU= 11 hours)

(*) number of nodes in BDD

Research and development at KKL and ETH Zurich

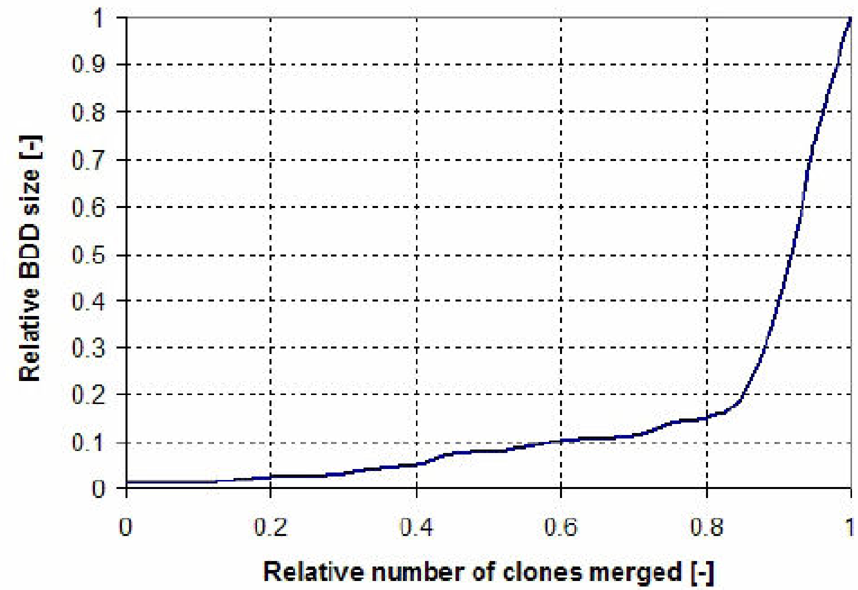
- **Algorithm *FUSION* (BDD as objects)**

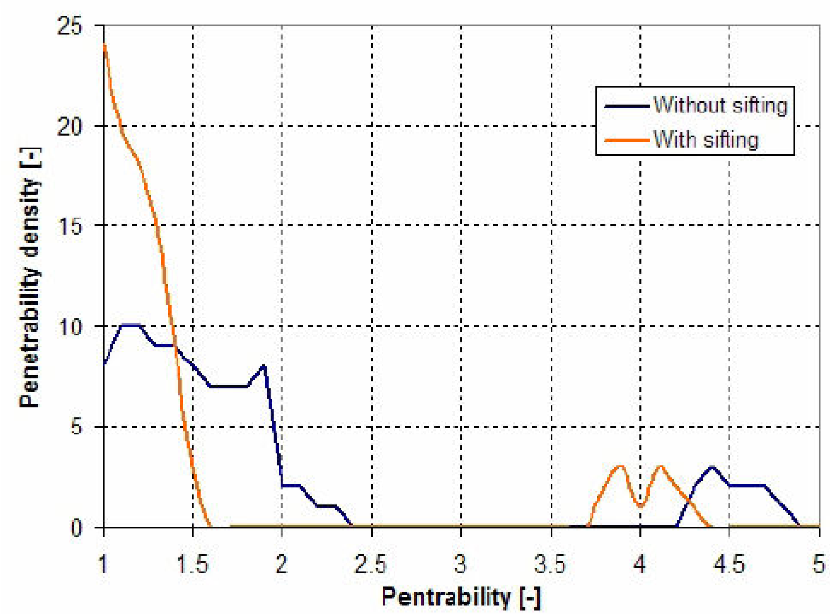


Research and development at KKL and ETH Zurich

- **Insights from *FUSION***
  - ☼ About 90% of the variables can be merged without major impact on BDD size
  - ☼ Penetrability *p*: effective identification of “hot spots”

$$p(x) = \frac{Size(Opt(BDD_{x\ merged}))}{Size(BDD)}$$





Research and development at KKL and ETH Zurich

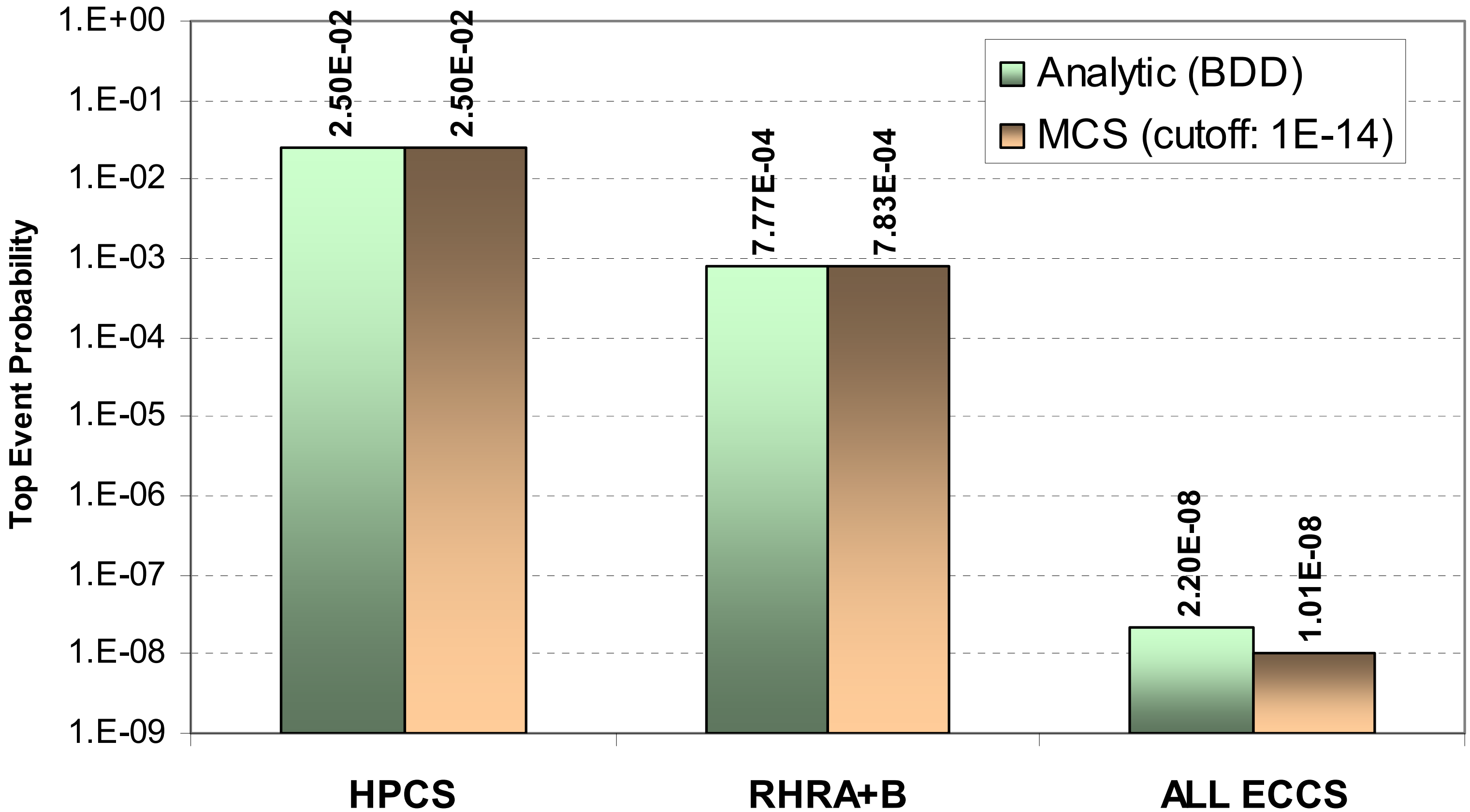
- **Improved Dynamic Group-Sifting Using Modularization (*IDGSM*)**
  - ☼ Further limitation of global perturbation when optimizing locally
  - ☼ Online identification and treatment of “hot spots” using penetrability spectrum
  - ☼ Improvement in the Group-Sifting algorithm
  - ☼ Use of genetic optimization algorithms
  - ☼ Generation and treatment of “clones” (Algorithm ***FUSION***)

Insights and outlook

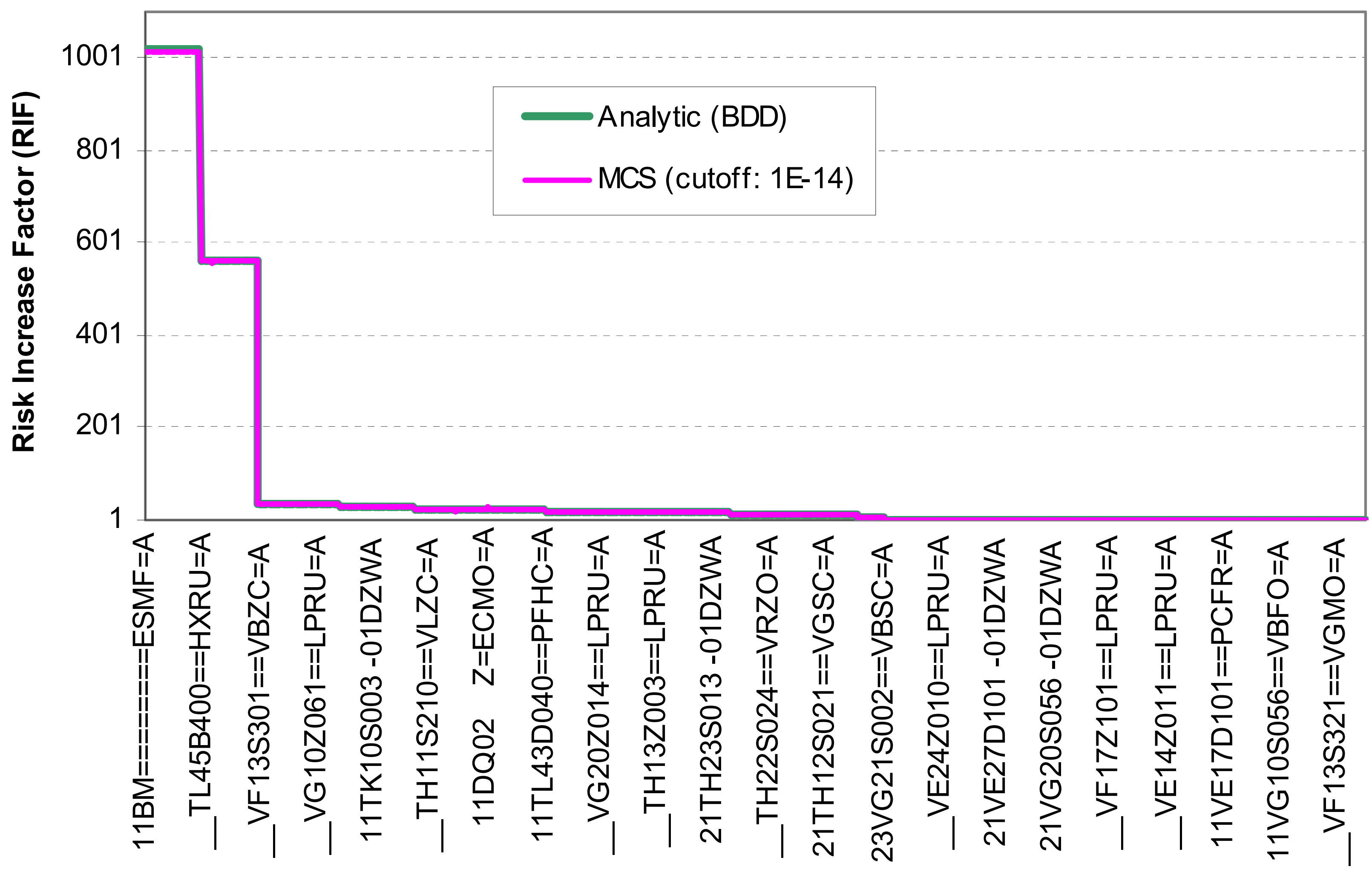
- **Results and insights:**
  - FTA quantification using BDD requires complex algorithms and programming techniques
  - The combination of global, static, dynamic and BDD objects techniques proved to be effective when dealing with large models
  - We succeeded in converting the Leibstadt PSA model to a BDD form of more than 1’500’000 with 30 clones, for a total of about 3500 basic events.

    The representative sequence includes reactor shutdown and reactor cooling with the eight Emergency Core Cooling Systems of the Leibstadt Nuclear Power Plant (including all support systems)

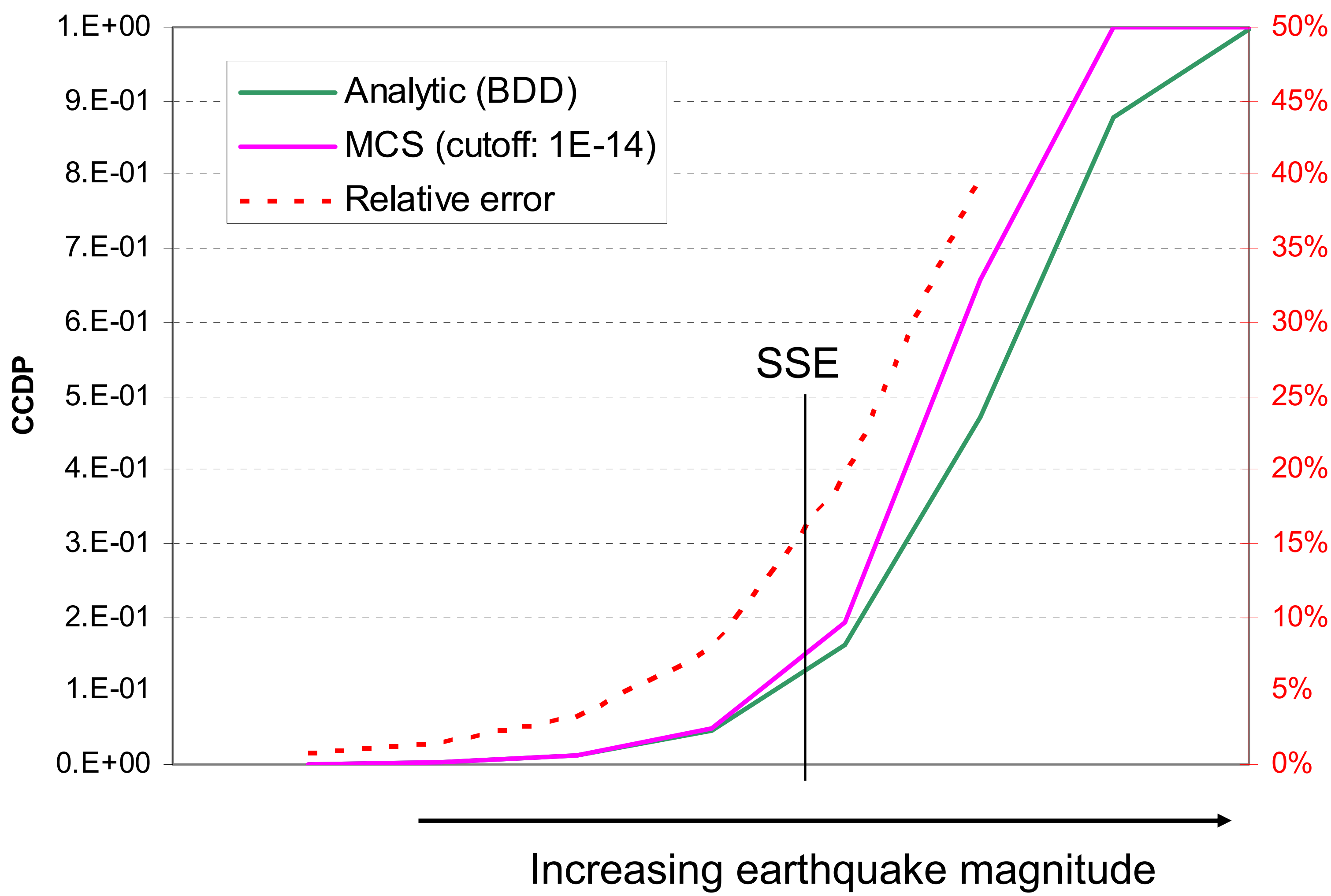
Results: Internal Events Comparison



Results: Importance Analysis Comparison (RHR A+B)



Results: Seismic PSA Comparison



Insights and outlook

- **The nuclear industry is facing a major issue (and is not yet fully aware of it):**
  - ☼ Worldwide probabilistic analyses are performed with codes that produce questionable results (conservative or optimistic, no one can know)
  - ☼ New IAEA requirements are difficult to address with existing FTA quantifiers (e.g. seismic assessments, Level 2 PSA)
  - ☼ Utilities and authorities *still* trust the results of existing FTA quantifiers, applying approximations where they should not be applied
    (→ blind-trust in numbers)
  - ☼ The techniques developed in this study raises the BDD approach to a mature technology for PSA model solving