# New PSA engines and Binary Decision Diagram

*Cristina Ibáñez Llano*

Next Generation PSA Software, Declarative Modeling, and Model Representation Standards Workshop

Kernkraftwerk Gösgen - Daeniken – Switzerland – 12 June 2007

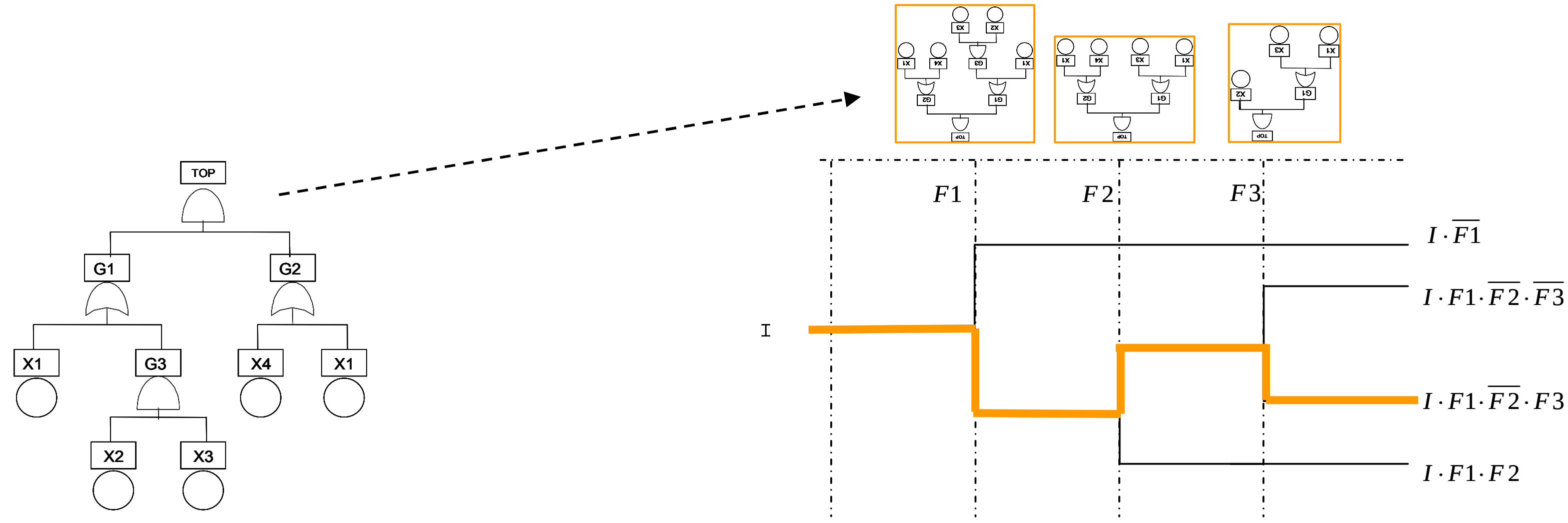
# Introduction

- **Workshop Objectives**
  1. Next generation PSA software → new calculation engines techniques
  2. Declarative modelling → better user interfaces. Completeness of models
  3. Model Representation Standards → software-independent format. Standard language

- **PSA software: calculation engines**
  - ✓ Classical methodology:
    - Calculation of MCS: direct manipulation of the Boolean formulae
    - Approximations & assumptions (truncations, rare event approximation)
  - ✓ BDD technology:
    - Encoding Boolean model in BDD: more compact data structure
    - Analysis relies on BDD: improvement of accuracy and efficiency
    - Main drawback: sensitivity to initial variable ordering (memory blow-up)
    - Mature technology:
      - Most work centered on Fault Tree models
      - Next step: linked Fault Trees and Event Trees

# PSA models and BDD

- **PSA models. Event Tree & Fault Trees**




- **BDD methodology for PSA models**
  - ✓ Sequences of non-disjoints or linked Fault Trees (variables in common)
  - ✓ Current treatment: sequences reduced to an AND large gate
  - ✓ Need to **extend previous strategies** to deal with groups of **non-disjoints FTs**
  - ✓ Key issue: treatment of shared variables (domains intersection):
    1. Conversion Strategies
    2. Ordering Schemes

# Strategies for conversion (I)

- **Conversion procedures for linked Fault Trees**
  - ✓ Non-disjoints FT within the same sequence: common variables
  - ✓ To apply BDD approach: convert whole sequence to BDD before analysis

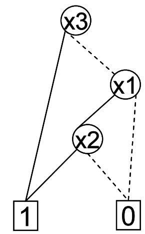
- **Collapsed Method**
  - ✓ Sequence delineation completely known, reduced to a higher AND gate (large FT)
  - ✓ Compilation and pre-processing of full sequence before tackling BDD conversion
  - ✓ BDD obtained in one collapsed step

- **Accumulative method**
  - ✓ Partially compose the BDD of the linked FTs
  - ✓ Compilation, pre-processing & conversion considered for each FT
  - ✓ BDD obtained incrementally:
    - Compose previous partial sequence BDD with the Fault Tree BDD

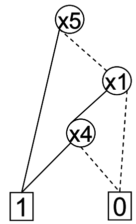
# Strategies for conversion (II)

- **Accumulative procedure. Example** $\boxed{F_1 \bullet F_2}$

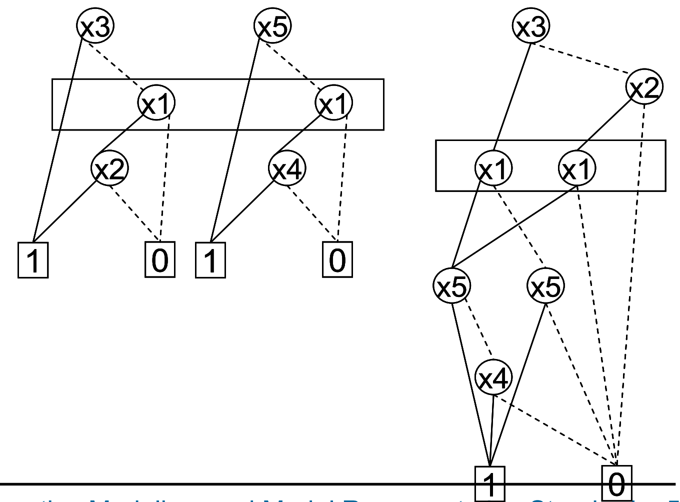
$$F_1 = x_1 \cdot x_2 + x_3 \xrightarrow{\sigma_1} \boldsymbol{BDD_1} \xrightarrow{\sigma_{seq}} BDD_{seq} = BDD_1$$




$$F_2 = x_4 \cdot x_1 + x_5 \xrightarrow{\sigma_2} \boldsymbol{BDD_2} \xrightarrow{\sigma_{seq}} BDD_{seq} = BDD_{seq} \bullet BDD_2 =$$




$$\boldsymbol{BDD_1} \bullet \boldsymbol{BDD_2} = \boldsymbol{BDD_{1,2}}$$



# Strategies for conversion (III)

- **Advantages for the incremental approach**
  - ✓ **Flexibility** to apply the methodology by defining more **local** strategies
  - ✓ **Reutilization** of previous calculations (shared sub-branches)
  - ✓ **Adaptable** for dynamics approaches
  - ✓ Study of **direct effect** of each subsystem on the final result

- **Research efforts need to be done on:**
  - ✓ Strategies for variable ordering and for conversion
  - ✓ Shared variables ordering in the overall sequence ordering

# Variable Ordering Schemes (I) - Review

- Problem studied exclusively in the context of FT (sequence → large AND gate)

**Variable Ordering Schemes**

**Static methods (SVO)**
Initial ordering prior to conversion
Topological considerations

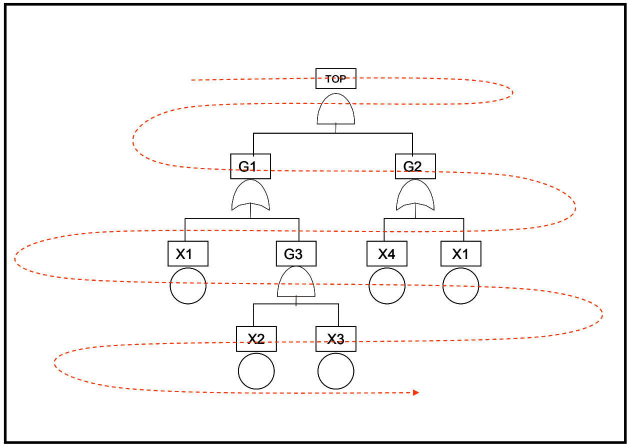
**Dynamic methods (SVO)**
change ordering during conversion
Swapping and sifting variables

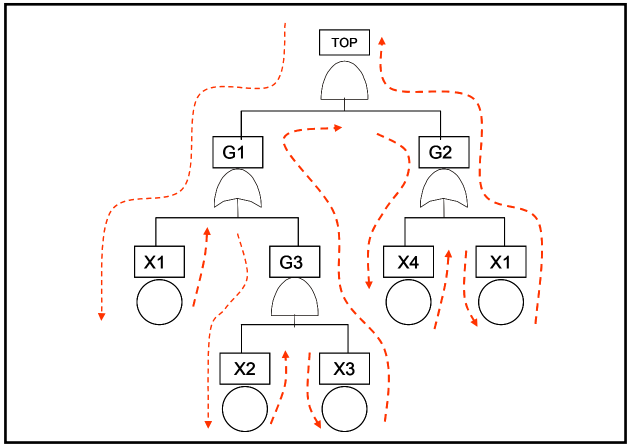
1. **Rule-based schemes:**
   - ✓ general or universal ordering scheme
   - ✓ straightforward rule-based strategy
   - ✓ Simple to implement and easy to compute
   - ✓ Categories: structural & weighted methods
2. **Adaptable schemes approach**
   - ✓ No universal method: select the best basic scheme for each case

# Variable Ordering Schemes (II) - Review

- **Static Rule-based Schemes:**
  - ✓ **Structural**
    - organized traversal of the tree
    - Two basic: breadth-first (BFS) & depth-first (DFS)
    - Variants:
      - – reordering arguments (formula rewriting)
      - – Different criteria to modify priority of exploration
    - More used: preserve neighborhoods & respect modules
  - ✓ **Weigthed**
    - Complete rearrange: different measures to variables

BFS



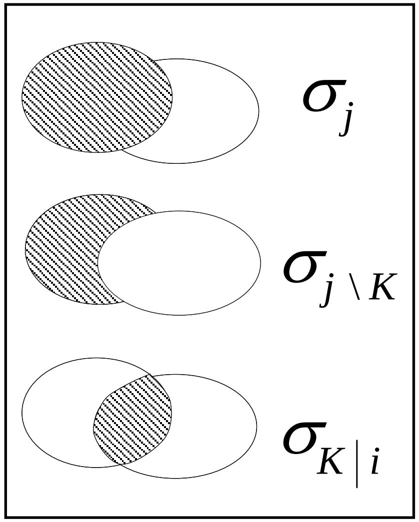




DFS

# Variable Ordering Schemes (III) - Extensions

- **Motivation**
  - ✓ Good orderings for each FT, **but …**
  - ✓ Strategies to **combine** individual orderings & study **domain intersection ordering**

- **Questions**
  - ✓ Keep common variables together?
  - ✓ Position of common variables?
  - ✓ Apply strategy locally or globally?




- **Notation**
  - ✓ $\sigma_j$ : variables ordering for tree $F_j$
  - ✓ $\sigma_{j \setminus K}$ : variables ordering of tree $F_j$ without variables of $Dom(K)$
  - ✓ $\sigma_{K | i}$ : variables ordering of $Dom(K)$ induced by ordering of $F_i$

# Variable Ordering Schemes (IV) - Extensions

- **Extensions**
  - ✓ **EH1**: Do not keep together domains intersection. Order variables when encountered
  - ✓ **EH2**: Consecutive tree domain intersection variables together, placed in the middle
  - ✓ **EH3**: All trees common variables together, placed at the beginning of global ordering
  - ➢ Individual orderings obtained with different basic scheme. Individual study is required

- **Preliminary studies and results**
  - ✓ Accumulative method gives good results with some ordering extensions, showing the potential of this approach
  - ✓ Concerning the orderings extensions, it appears beneficial but opposite effects:
    - keeping common variables together vs. preserving neighbourhoods

- **Future work**
  - ✓ Compromise between both effects
  - ✓ Bigger benchmark with more configurations of linked FTs
  - ✓ Problem to obtain a good benchmark. Real models are very large

# Conclusions

- BDD approach offers great improvements for PSA models analysis & assessment
- Methodology has to be extended for linked Fault Trees
- The incremental approach provides flexibility for both PSA applications and dynamic extensions
- But the key issue of the common variables ordering has to be studied carefully
- An appropriate benchmark is needed to develop this ideas

# END PRESENTATION