Towards a (De-)Compositional Strategy for SAML

Michael Lipaczewski
Otto-von-Guericke Universität Magdeburg
- Intermediate Language for traditional and probabilistic analysis of formal models

![Diagram showing SAML's role in bridging between different tools and verification engines]

- Easy to understand and use
- Verification-engine independent
- Precise semantic
control unit, contains: sensor validator (SV), redundant crash detectors (CD), detection monitor (DM)
Ideas

Brute Force

Sparse Matrix

Composition
Software
• Especially in SAML: model is divided into functional behavior and failure pattern
  - Functional behavior: n states
  - Failure pattern: k states

• functional behavior already includes failure behavior
  - → include failure pattern into functional behavior
    - → will save k-1 states
• Target: making large models quantitative checkable

• Problems:
  • Approximation necessary
  • Functional behavior is lost

• Next Steps:
  • How big is the approximation error?
  • How big is the impact of local used states to the computation complexity?
Thanks for your attention

Questions?