Towards a (De-)Compositional Strategy for SAML

Michael Lipaczewski
Otto-von-Guericke Universität Magdeburg
Security Assertion Markup Language

System Analysis Modeling Language
• Intermediate Language for traditional and probabilistic analysis of formal models
What is SAML?
Case Study

control unit, contains:
sensor validator (SV),
redundant crash detectors (CD),
detection monitor (DM)

<table>
<thead>
<tr>
<th></th>
<th>Airbag Model</th>
<th>Extended Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>time for model construction</td>
<td>0.703 seconds</td>
<td>7.72 seconds</td>
</tr>
<tr>
<td>states</td>
<td>528363</td>
<td>499456001</td>
</tr>
<tr>
<td>transitions</td>
<td>4527444</td>
<td>16590774822</td>
</tr>
<tr>
<td>choices</td>
<td>0</td>
<td>998912002</td>
</tr>
<tr>
<td>nodes</td>
<td>60705</td>
<td>345934</td>
</tr>
</tbody>
</table>

Michael Lipaczewski

[Kloos/Hussain/Eschbach08]
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
    • \(\rightarrow\) include failure pattern into functional behavior
    • \(\rightarrow\) 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 computational complexity?
Thanks for your attention

Questions?