The What, Why, and How of Probabilistic Verification
Part 3: Towards Verifying Gigantic Markov Models

Joost-Pieter Katoen

UNIVERSITY OF TWENTE.

CAV Invited Tutorial 2015, San Francisco
Overview

Treating Gigantic Markov Models
Treating Gigantic Markov Models
Payload is mission-specific, e.g.:

- telecom transponders,
- navigation signals,
- earth observation telemetry
Payload is mission-specific, e.g.:
- telecom transponders,
- navigation signals,
- earth observation telemetry

Platform keeps the satellite in space:
- attitude and orbital control
- power distribution
- data handling
- communication
- thermal regulation
### AADL Model of Satellite Platform

<table>
<thead>
<tr>
<th>Scope</th>
<th>Metric</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Components</td>
<td></td>
<td>86</td>
</tr>
<tr>
<td>Ports</td>
<td></td>
<td>937</td>
</tr>
<tr>
<td>Modes</td>
<td></td>
<td>244</td>
</tr>
<tr>
<td>Error models</td>
<td></td>
<td>20</td>
</tr>
<tr>
<td>Recoveries</td>
<td></td>
<td>16</td>
</tr>
<tr>
<td><strong>Nominal state space</strong></td>
<td></td>
<td><strong>48421100</strong></td>
</tr>
<tr>
<td>LOC (without comments)</td>
<td></td>
<td>3831</td>
</tr>
<tr>
<td><strong>Model</strong></td>
<td><strong>Propositional</strong></td>
<td>25</td>
</tr>
<tr>
<td></td>
<td><strong>Absence</strong></td>
<td>2</td>
</tr>
<tr>
<td><strong>Requirements</strong></td>
<td><strong>Universality</strong></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td><strong>Response</strong></td>
<td>14</td>
</tr>
<tr>
<td></td>
<td><strong>Probabilistic Invariance</strong></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td><strong>Probabilistic Existence</strong></td>
<td>1</td>
</tr>
</tbody>
</table>
State Space Growth by Fault Injection

- Single earth sensor signal failure: 1 injection, 3 multiplications
- Double earth sensors signal failure: 2 injections, 11 multiplications
- Propulsion failure: 5 injections, 35 multiplications
- AOCS equipments failure: 7 injections, 213563 multiplications
- All reactionwheel failures: 8 injections, 16 multiplications
- Single reactionwheel failure: 8 injections, 10 multiplications
- Processor module failures: 10 injections, 1372 multiplications
- Complete earth sensor failure: 11 injections, 14 multiplications
- Reactionwheel + earth sensor failures: 22 injections, 172 multiplications
Conquering the State Space Explosion Problem

1. Symbolic approaches using (MT)BDDs

2. Bisimulation minimisation

3. Aggressive abstraction beyond bisimulation

4. Compositional abstraction

5. Confluence reduction (aka: partial-order reduction)

6. Exploit (multiple) multi-core processor(s)

7. Resort to discrete event simulation

\(^1\)In modern terminology: statistical model checking.
Probabilistic Bisimulation

[Larsen & Skou, 1989]
Consider a DTMC with state space $S$ and equivalence $R \subseteq S \times S$. Then: $R$ is a probabilistic bisimulation on $S$ if for any $(s, t) \in R$:

1. $L(s) = L(t)$, and
2. $P(s, C) = P(t, C)$ for all equivalence classes $C \in S/R$

where $P(s, C) = \sum_{s' \in C} P(s, s')$. 
Consider a DTMC with state space $S$ and equivalence $R \subseteq S \times S$. Then: $R$ is a probabilistic bisimulation on $S$ if for any $(s, t) \in R$: 

1. $L(s) = L(t)$, and
2. $P(s, C) = P(t, C)$ for all equivalence classes $C \in S/R$

where $P(s, C) = \sum_{s' \in C} P(s, s')$.

Let $\sim$ denote the largest possible probabilistic bisimulation.
Properties

Quotienting: using partition-refinement in $O(|P| \cdot \log |S|)$

Preservation: all probabilistic CTL*-formulas

Congruence: with respect to parallel composition

Continuous: can all readily be adapted to CTMCs

Stuttering: weak variants are around and preserve PCTL* without next

Savings: potentially exponentially in time and space
Reducing Crowds Protocol

[Reiter & Rubin, 1998]

The diagram shows the state space size in terms of the number of protocol runs for different values of N. The graph illustrates how the state space size grows exponentially as the number of runs increases.

- For N = 10, the state space size grows from $10^1$ to $10^6$ as the number of runs increases from 2.8 to 6.4.
- For N = 15, the state space size grows from $10^2$ to $10^7$ as the number of runs increases from 2.8 to 6.4.

The diagram helps to visualize the impact of increasing N on the state space size.
### Reducing IEEE 802.11 Group Communication Protocol

<table>
<thead>
<tr>
<th>OD</th>
<th>states</th>
<th>transitions</th>
<th>ver. time</th>
<th>blocks</th>
<th>total time</th>
<th>red. factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>1125</td>
<td>5369</td>
<td>122</td>
<td>71</td>
<td>13</td>
<td>15.9</td>
</tr>
<tr>
<td>12</td>
<td>37349</td>
<td>236313</td>
<td>7180</td>
<td>1821</td>
<td>642</td>
<td>20.5</td>
</tr>
<tr>
<td>20</td>
<td>231525</td>
<td>1590329</td>
<td>50133</td>
<td>10627</td>
<td>5431</td>
<td>21.8</td>
</tr>
<tr>
<td>28</td>
<td>804837</td>
<td>5750873</td>
<td>195086</td>
<td>35961</td>
<td>24716</td>
<td>22.4</td>
</tr>
<tr>
<td>36</td>
<td>2076773</td>
<td>15187833</td>
<td>5103900</td>
<td>91391</td>
<td>77694</td>
<td>22.7</td>
</tr>
<tr>
<td>40</td>
<td>3101445</td>
<td>22871849</td>
<td>7725041</td>
<td>135752</td>
<td>127489</td>
<td>22.9</td>
</tr>
</tbody>
</table>

all times in milliseconds
## Reducing BitTorrent-like P2P protocol

<table>
<thead>
<tr>
<th>$N$</th>
<th>states</th>
<th>ver. time</th>
<th>$N$ states</th>
<th>blocks</th>
<th>lump time</th>
<th>ver. time</th>
<th>states</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1024</td>
<td>5.6</td>
<td>528</td>
<td>56</td>
<td>1.4</td>
<td>0.3</td>
<td>18.3</td>
<td>3.3</td>
</tr>
<tr>
<td>3</td>
<td>32768</td>
<td>410</td>
<td>5984</td>
<td>252</td>
<td>170</td>
<td>1.3</td>
<td>130</td>
<td>2.4</td>
</tr>
<tr>
<td>4</td>
<td>1048576</td>
<td>22000</td>
<td>52360</td>
<td>792</td>
<td>10200</td>
<td>4.8</td>
<td>1324</td>
<td>2.2</td>
</tr>
</tbody>
</table>

Bisimulation may reduce a factor 66 after (manual) symmetry reduction.
Principle of Compositional Minimisation

- Interactive Markov chains
  - mix of labeled transition systems and CTMCs
  - allow for compositional modeling
  - and non-determinism (aka: CTMDPs)
Treating Gigantic Markov Models

Principle of Compositional Minimisation

- Interactive Markov chains
  - mix of labeled transition systems and CTMCs
  - allow for compositional modeling
  - and non-determinism (aka: CTMDPs)

- Congruence property

\[(\mathcal{M}_1 \sim \mathcal{N}_1 \text{ and } \mathcal{M}_2 \sim \mathcal{N}_2) \implies \mathcal{M}_1 \parallel_A \mathcal{M}_2 \sim \mathcal{N}_1 \parallel_A \mathcal{N}_2\]
Principle of Compositional Minimisation

- Interactive Markov chains
  - mix of labeled transition systems and CTMCs
  - allow for compositional modeling
  - and non-determinism (aka: CTMDPs)

- Congruence property

  \((M_1 \sim N_1 \text{ and } M_2 \sim N_2)\) implies \(M_1 \parallel_A M_2 \sim N_1 \parallel_A N_2\)

- Component-wise minimisation\(^a\)
  1. Consider \(M_1 \parallel_A \ldots \parallel_A M_i \parallel_A \ldots \parallel_A M_k\)
  2. Pick process \(M_i\) and consider its quotient under \(\sim\)
  3. Yielding \(M_1 \parallel_A \ldots \parallel_A M_i/\sim \parallel_A \ldots \parallel_A M_k\)
  4. This can also be applied to groups of processes

\(^a\)This paradigm is well-supported by the CADP tool.
Compositional Minimisation of DFTs

(a) DFT  (b) Transformation

(c) Composition  (d) Minimisation  (e) IMC
### Compositional Minimisation of DFTs

<table>
<thead>
<tr>
<th>case study</th>
<th>peak # states</th>
<th># transitions</th>
<th>unreliability</th>
<th>time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPS</td>
<td>4113</td>
<td>24608</td>
<td>.00135</td>
<td>490</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPS</td>
<td>133</td>
<td>465</td>
<td>.00135</td>
<td>67</td>
</tr>
</tbody>
</table>

Comparing Galileo DIFTree (top) to new approach (bottom)
### Compositional Minimisation of DFTs

<table>
<thead>
<tr>
<th>case study</th>
<th>peak # states</th>
<th># transitions</th>
<th>unreliability</th>
<th>time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPS</td>
<td>4113</td>
<td>24608</td>
<td>.00135</td>
<td>490</td>
</tr>
<tr>
<td>CAS</td>
<td>8</td>
<td>10</td>
<td>.65790</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>case study</th>
<th>peak # states</th>
<th># transitions</th>
<th>unreliability</th>
<th>time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPS</td>
<td>133</td>
<td>465</td>
<td>.00135</td>
<td>67</td>
</tr>
<tr>
<td>CAS</td>
<td>36</td>
<td>119</td>
<td>.65790</td>
<td>94</td>
</tr>
</tbody>
</table>

Comparing Galileo DIFTree (top) to new approach (bottom)
Compositional Minimisation of DFTs

<table>
<thead>
<tr>
<th>case study</th>
<th>peak # states</th>
<th># transitions</th>
<th>unreliability</th>
<th>time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPS</td>
<td>4113</td>
<td>24608</td>
<td>.00135</td>
<td>490</td>
</tr>
<tr>
<td>CAS</td>
<td>8</td>
<td>10</td>
<td>.65790</td>
<td>1</td>
</tr>
<tr>
<td>CAS-PH</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>NDPS</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>

Comparing Galileo DIFTree (top) to new approach (bottom)

<table>
<thead>
<tr>
<th>case study</th>
<th>peak # states</th>
<th># transitions</th>
<th>unreliability</th>
<th>time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPS</td>
<td>133</td>
<td>465</td>
<td>.00135</td>
<td>67</td>
</tr>
<tr>
<td>CAS</td>
<td>36</td>
<td>119</td>
<td>.65790</td>
<td>94</td>
</tr>
<tr>
<td>CAS-PH</td>
<td>40052</td>
<td>265442</td>
<td>.112</td>
<td>231</td>
</tr>
<tr>
<td>NDPS</td>
<td>61</td>
<td>169</td>
<td>[.00586, .00598]</td>
<td>266</td>
</tr>
</tbody>
</table>
## Compositional Minimisation of DFTs

<table>
<thead>
<tr>
<th>Case Study</th>
<th>Peak # States</th>
<th># Transitions</th>
<th>Unreliability</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPS</td>
<td>4113</td>
<td>24608</td>
<td>0.00135</td>
<td>490</td>
</tr>
<tr>
<td>CAS</td>
<td>8</td>
<td>10</td>
<td>0.65790</td>
<td>1</td>
</tr>
<tr>
<td>CAS-PH</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>NDPS</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>FTTP-4</td>
<td>32757</td>
<td>426826</td>
<td>0.01922</td>
<td>13111</td>
</tr>
<tr>
<td>FTTP-5</td>
<td>MO</td>
<td>MO</td>
<td>MO</td>
<td>MO</td>
</tr>
<tr>
<td>CPS</td>
<td>133</td>
<td>465</td>
<td>0.00135</td>
<td>67</td>
</tr>
<tr>
<td>CAS</td>
<td>36</td>
<td>119</td>
<td>0.65790</td>
<td>94</td>
</tr>
<tr>
<td>CAS-PH</td>
<td>40052</td>
<td>265442</td>
<td>0.112</td>
<td>231</td>
</tr>
<tr>
<td>NDPS</td>
<td>61</td>
<td>169</td>
<td>[0.00586, 0.00598]</td>
<td>266</td>
</tr>
<tr>
<td>FTTP-4</td>
<td>1325</td>
<td>13642</td>
<td>0.01922</td>
<td>65</td>
</tr>
<tr>
<td>FTTP-6</td>
<td>11806565</td>
<td>22147378</td>
<td>0.00045</td>
<td>1989</td>
</tr>
</tbody>
</table>

Comparing Galileo DIFTree (top) to new approach (bottom)
<table>
<thead>
<tr>
<th>case study</th>
<th>peak # states</th>
<th># transitions</th>
<th>unreliability</th>
<th>time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPS</td>
<td>4113</td>
<td>24608</td>
<td>.00135</td>
<td>490</td>
</tr>
<tr>
<td>CAS</td>
<td>8</td>
<td>10</td>
<td>.65790</td>
<td>1</td>
</tr>
<tr>
<td>CAS-PH</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>NDPS</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>FTTP-4</td>
<td>32757</td>
<td>426826</td>
<td>.01922</td>
<td>13111</td>
</tr>
<tr>
<td>FTTP-5</td>
<td>MO</td>
<td>MO</td>
<td>MO</td>
<td>MO</td>
</tr>
</tbody>
</table>

Comparing Galileo DIFTree (top) to new approach (bottom)

In practice, DFTs of >50 nodes are not an exception.
**Key idea**

Simplify DFTs by graph rewriting prior to (compositional) state space generation.
Treating Gigantic Markov Models

Tailored DFT Abstraction

- Total verification and minimisation time
- State space size of resulting CTMDP

49 out of 179 case studies could be treated now that could not be treated before
More Aggressive Abstraction

- Partition the state space into groups of concrete states
  - allow any partitioning, not just grouping of bisimilar states
More Aggressive Abstraction

- **Partition the state space** into groups of concrete states
  - allow any partitioning, not just grouping of bisimilar states

- **Use three-valued** semantics
  - abstraction is conservative for both negative and positive results
  - if verification yields *don’t know*, validity in concrete model is unknown
More Aggressive Abstraction

- **Partition the state space** into groups of concrete states
  - allow any partitioning, not just grouping of bisimilar states

- **Use three-valued semantics**
  - abstraction is conservative for both negative and positive results
  - if verification yields *don’t know*, validity in concrete model is unknown

- **Important aspects:**
  - ingredients of abstract probabilistic models
  - how to verify abstracts models?
  - how accurate are abstractions in practice?
Intuition of Abstraction

Interval abstraction

CTMDP abstraction
Theoretical Results on Abstraction

1. For a given state-space partitioning: abstract probabilistic model “simulates” concrete model (but not the converse)

2. If \( s \sqsubseteq s' \) and \( \text{[}\Phi\text{](s')} \neq ? \) then: \( \text{[}\Phi\text{](s')} = \text{[}\Phi\text{](s)} \) for any formula \( \Phi \) in continuous stochastic logic (without next)

3. Extreme policies suffice for verifying interval-probabilistic models

4. Step-bounded and time-bounded reachability can be checked in polynomial time

5. Interval Markov chains + modal transition systems yields a useful and elegant framework for compositional abstraction

6. “Simulation” is a pre-congruence with respect to parallel composition, so:

\[
M_1 \sqsubseteq N_1 \text{ and } M_2 \sqsubseteq N_2 \implies M_1 \parallel_A M_2 \sqsubseteq N_1 \parallel_A N_2
\]
Substrate Conversion

- Verification takes \textit{days}
- \(\approx 6 \cdot 10^7\) iterations needed
- Mainly due to stiffness
- No bisimilar states
- Solution: \textcolor{blue}{abstraction}
Example: Substrate Conversion

rule of thumb: group sets of “fast” connected states
Improving Lower Bounds

\[
\begin{align*}
[1 - p_1, 1] & \quad [1 - p_2, 1] & \quad [1 - p_3, 1] & \quad [1 - p_4, 1] & \quad 1 \\
[0, p_1] & \quad [0, p_2] & \quad [0, p_3] & \quad [0, p_4] & \\
[1 - p_1, 1 - p_1 - q_1, 1 - p_4'] & \quad [1 - p_2, 1 - p_2 - q_2, 1 - p_3'] & \quad [1 - p_3, 1 - p_3 - q_3, 1 - p_4'] & \quad [1 - p_4, 1 - p_4'] & \quad 1 \\
[0, p_1'] & \quad [0, p_2'] & \quad [0, p_3'] & \quad [0, p_4'] & \\
[0, q_1] & \quad [0, q_2] & \quad [0, q_3] & \\
\end{align*}
\]
probability of only having products in deadline $t$ (200 substrates, 20 enzymes)

results using Markov Chain Model Checker www.mrmc-tool.org
Example: Abstracting Queueing Networks

- Application: a $M/PH_n/1$ queueing station with preemptive scheduling
- Model: tree-based quasi-birth death (QBD) process
- Alternatively: a probabilistic push-down automaton
- Chance from a given configuration to serve up to $k$ jobs within a deadline?
Experimental Results

Comparing different partitioning schemes and influence of cut level:

- qbd
- grid
- bd
- tree

Prob vs. time (t)

- 16
- 24
- 32
- 40
- 48+
### Experimental Results

Grid abstraction versus tree analysis techniques (error bound is $10^{-6}$):

<table>
<thead>
<tr>
<th>diff</th>
<th>grid 12</th>
<th>grid 16</th>
<th>grid 20</th>
<th>grid 24</th>
<th>grid 28</th>
<th>grid 32</th>
<th>grid 36</th>
<th>grid 40</th>
<th>trunc</th>
<th>≈ states</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.5</td>
<td>0.0224</td>
<td>0.001</td>
<td>$10^{-6}$</td>
<td>$10^{-6}$</td>
<td>$10^{-6}$</td>
<td>$10^{-6}$</td>
<td>$10^{-6}$</td>
<td>$10^{-6}$</td>
<td>185</td>
<td>$10^{129}$</td>
</tr>
<tr>
<td>t</td>
<td>10</td>
<td>0.3117</td>
<td>0.0580</td>
<td>0.0062</td>
<td>0.0004</td>
<td>$10^{-5}$</td>
<td>$10^{-6}$</td>
<td>$10^{-6}$</td>
<td>270</td>
<td>$10^{188}$</td>
</tr>
<tr>
<td>15</td>
<td>0.4054</td>
<td>0.1345</td>
<td>0.0376</td>
<td>0.0086</td>
<td>0.0015</td>
<td>0.0002</td>
<td>$2 \cdot 10^{-5}$</td>
<td>$3 \cdot 10^{-6}$</td>
<td>398</td>
<td>$10^{278}$</td>
</tr>
<tr>
<td>states</td>
<td>6188</td>
<td>20349</td>
<td>53130</td>
<td>118755</td>
<td>237336</td>
<td>435894</td>
<td>749398</td>
<td>1221759</td>
<td></td>
<td></td>
</tr>
<tr>
<td>distributions</td>
<td>28666</td>
<td>96901</td>
<td>256796</td>
<td>579151</td>
<td>1164206</td>
<td>2146761</td>
<td>3701296</td>
<td>6047091</td>
<td></td>
<td></td>
</tr>
<tr>
<td>time (h:m:s)</td>
<td>0:00:26</td>
<td>0:01:33</td>
<td>0:04:15</td>
<td>0:09:50</td>
<td>0:20:14</td>
<td>0:38:13</td>
<td>1:07:57</td>
<td>2:06:04</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Grid abstraction versus tree analysis techniques (error bound is $10^{-6}$):

<table>
<thead>
<tr>
<th>diff</th>
<th>grid 12</th>
<th>grid 16</th>
<th>grid 20</th>
<th>grid 24</th>
<th>grid 28</th>
<th>grid 32</th>
<th>grid 36</th>
<th>grid 40</th>
<th>trunc</th>
<th>$\approx$ states</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.5</td>
<td>0.0224</td>
<td>0.001</td>
<td>$10^{-6}$</td>
<td>$10^{-6}$</td>
<td>$10^{-6}$</td>
<td>$10^{-6}$</td>
<td>$10^{-6}$</td>
<td>$10^{-6}$</td>
<td>185</td>
<td>$10^{129}$</td>
</tr>
<tr>
<td>10</td>
<td>0.3117</td>
<td>0.0580</td>
<td>0.0062</td>
<td>0.0004</td>
<td>$10^{-5}$</td>
<td>$10^{-6}$</td>
<td>$10^{-6}$</td>
<td>$10^{-6}$</td>
<td>270</td>
<td>$10^{188}$</td>
</tr>
<tr>
<td>15</td>
<td>0.4054</td>
<td>0.1345</td>
<td>0.0376</td>
<td>0.0086</td>
<td>0.0015</td>
<td>0.0002</td>
<td>$2 \cdot 10^{-5}$</td>
<td>$3 \cdot 10^{-5}$</td>
<td>398</td>
<td>$10^{278}$</td>
</tr>
<tr>
<td>states</td>
<td>6188</td>
<td>20349</td>
<td>53130</td>
<td>118755</td>
<td>237336</td>
<td>435894</td>
<td>749398</td>
<td>1221759</td>
<td></td>
<td></td>
</tr>
<tr>
<td>distributions</td>
<td>28666</td>
<td>96901</td>
<td>256796</td>
<td>579151</td>
<td>1164206</td>
<td>2146761</td>
<td>3701296</td>
<td>6047091</td>
<td></td>
<td></td>
</tr>
<tr>
<td>time (h:m:s)</td>
<td>0:00:26</td>
<td>0:01:33</td>
<td>0:04:15</td>
<td>0:09:50</td>
<td>0:20:14</td>
<td>0:38:13</td>
<td>1:07:57</td>
<td>2:06:04</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$\Rightarrow$ Abstraction yields same accuracy by 1.2 million state as $10^{278}$ concrete ones

$\Rightarrow$ First time that tree-based QBDs of this size have been successfully analysed
Compositional Abstraction

- **Interactive Markov chains (IMCs)**
  - mix of transition systems and CTMCs
  - allow for compositional modeling
  - and compositional minimisation
Compositional Abstraction

- **Interactive Markov chains (IMCs)**
  - mix of transition systems and CTMCs
  - allow for compositional modeling
  - and compositional minimisation

- **Abstract IMCs**
  - use interval abstraction
  - and modal transition systems (MTS)
Compositional Abstraction

- **Interactive Markov chains (IMCs)**
  - mix of transition systems and CTMCs
  - allow for compositional modeling
  - and compositional minimisation

- **Abstract IMCs**
  - use interval abstraction
  - and modal transition systems (MTS)

- **Aim: abstract component-wise**
  - replace $\mathcal{M}_i$ by $\alpha(\mathcal{M}_i)$
  - then $\mathcal{M}_1 \parallel \ldots \parallel \mathcal{M}_n$ by $\alpha(\mathcal{M}_1) \parallel \ldots \parallel \alpha(\mathcal{M}_n)$
Compositional Abstraction
Compositional Abstraction

Diagram:

- States: \( s_0, s_1, u_0, u_1 \)
- Edges:
  - \( s_0 \rightarrow s_1 \) with label \( a \)
  - \( s_1 \rightarrow s_0 \) with label \( a \)
  - \( s_0 \rightarrow u_0 \) with label \( b \)
  - \( s_1 \rightarrow u_0 \) with label \( \frac{1}{2} \)
  - \( u_0 \rightarrow s_0 \) with label \( \frac{1}{2} \)
  - \( u_0 \rightarrow u_1 \) with label \( \frac{2}{3} \)
  - \( u_1 \rightarrow u_0 \) with label \( \frac{1}{3} \)

- Labels:
  - \( s_0 \) and \( s_1 \) have an initial probability of \( 1 \)
  - \( u_0 \) and \( u_1 \) have a probability distribution of \( \frac{1}{2} \) and \( \frac{2}{3} \) respectively.
Compositional Abstraction
Compositional Abstraction

Diagram:

- States: $s_0$, $s_1$, $u_0$, $u_1$, $s$, $u$
- Edges:
  - $s_0$ to $u_0$ with probability $\frac{1}{2}$
  - $s_1$ to $u_1$ with probability $\frac{1}{3}$
  - $u_0$ to $u_1$ with probability $\frac{2}{3}$
  - $s$ to $u$ with probability $\frac{1}{3}$ and $\frac{2}{3}$
- Labels:
  - $a$ and $\frac{1}{3}$ for $s_0$ to $s_1$
  - $b$ for $u_0$ to $u_1$
  - May $b$ for $s$ to $u$

Joost-Pieter Katoen
Compositional Abstraction

![Diagram of compositional abstraction](image-url)
Compositional Abstraction

\[ \text{Diagram with states and transitions:} \]

- States: \( s_0, s_1, u_0, u_1, s, u \)
- Transitions: \( a, b \)
- Probabilities: \( \frac{1}{2}, \frac{2}{3}, \frac{1}{3}, [\frac{1}{3}, \frac{1}{2}], [\frac{1}{2}, \frac{2}{3}] \)

- \( s_0 \) to \( u_0 \): \( b \) with probability \( \frac{1}{2} \)
- \( s_0 \) to \( s_1 \): \( 1 \)
- \( s_1 \) to \( u_1 \): \( a \) with probability \( \frac{1}{3} \)
- \( u_0 \) to \( u_1 \): \( a \) with probability \( \frac{2}{3} \)
- \( u_1 \) to \( s_0 \): \( b \) with probability \( \frac{1}{2} \)
- \( s \) to \( u \): \( a \) with probability \( [\frac{1}{3}, \frac{1}{2}] \)
- \( u \) to \( s \): \( [\frac{1}{2}, \frac{2}{3}] \)

- "may b" label on the transition from \( s \) to \( u \).
Parallel Composition

Treating Gigantic Markov Models

Joost-Pieter Katoen
Parallel Composition

\[
\begin{align*}
\text{may } b & \quad \text{(s, u)} \\
\text{may } b & \quad \text{(u, u)} \\
\text{may } b & \quad \text{(s, s)} \\
\end{align*}
\]

\[
\begin{align*}
\frac{1}{3}, \frac{1}{2} & \quad \frac{1}{2}, \frac{2}{3} \\
\frac{3}{4}, \frac{5}{6} & \quad \frac{1}{2}, \frac{2}{3} \\
\frac{1}{6}, \frac{1}{4} & \quad \frac{1}{6}, \frac{1}{4} \\
\frac{3}{4}, \frac{5}{6} & \quad \frac{1}{6}, \frac{1}{4} \\
\frac{1}{2}, \frac{2}{3} & \quad \frac{1}{6}, \frac{1}{4} \\
\end{align*}
\]
Parallel Composition

Treating Gigantic Markov Models

Joost-Pieter Katoen

What, Why, and How of Probabilistic Verification

38/44
Symmetric Composition

Multisets representing tuples: $\{s, u\} \approx \{(s, u), (u, s)\}$
Theoretical Results

- Symmetric composition and parallel composition are bisimilar

\[ \mathcal{M}^n \sim \mathcal{M} \mathcal{M} \ldots \mathcal{M} \]

\( n \) times
Theoretical Results

- Symmetric composition and parallel composition are bisimilar

\[ \mathcal{M} \sim A^\times A^\times \cdots A^\times A^\times \mathcal{M} \]

- Simulation is a pre-congruence wrt. \( \parallel \) and symmetric composition

\[ \mathcal{M}_1 \subseteq \mathcal{N}_1 \text{ and } \mathcal{M}_2 \subseteq \mathcal{N}_2 \text{ implies } \mathcal{M}_1 \parallel A \mathcal{M}_2 \subseteq \mathcal{N}_1 \parallel A \mathcal{N}_2 \]
Theoretical Results

- Symmetric composition and parallel composition are bisimilar

\[ \|\|_A^n \mathcal{M} \sim _A \mathcal{M} \|_A \ldots \|_A \mathcal{M} \]

- Simulation is a pre-congruence wrt. \( \| \) and symmetric composition

\[ \mathcal{M}_1 \subseteq \mathcal{N}_1 \text{ and } \mathcal{M}_2 \subseteq \mathcal{N}_2 \text{ implies } \mathcal{M}_1 \|_A \mathcal{M}_2 \subseteq \mathcal{N}_1 \|_A \mathcal{N}_2 \]

- Bisimulation is a congruence wrt. \( \| \) and symmetric composition
Theoretical Results

- Symmetric composition and parallel composition are bisimilar

\[ \mathcal{M} \leadsto \mathcal{M} \]

- Simulation is a pre-congruence wrt. \( \| \) and symmetric composition

\[ \mathcal{M}_1 \subseteq \mathcal{N}_1 \text{ and } \mathcal{M}_2 \subseteq \mathcal{N}_2 \quad \text{implies} \quad \mathcal{M}_1 \|_A \mathcal{M}_2 \subseteq \mathcal{N}_1 \|_A \mathcal{N}_2 \]

- Bisimulation is a congruence wrt. \( \| \) and symmetric composition

- Abstracting many parallel “similar” components:

\[ (\text{for all } i. \mathcal{M}_i \subseteq \mathcal{N}) \quad \text{implies} \quad \mathcal{M}_1 \|_A \ldots \|_A \mathcal{M}_n \subseteq \|_A^n \mathcal{N} \]
A Production Example

- **Workers** $\mathcal{M}_i$ (8 states)
- **Counting process** $Q$ (44 states)

\[(\mathcal{M}_1 \parallel_0 \mathcal{M}_2 \parallel_0 \mathcal{M}_3) \parallel_A Q\]  
22528 states

- **Replace** $\mathcal{M}_i$ by abstract worker $\mathcal{N}$ (6 states)

\[(\mathcal{N} \parallel_0 \mathcal{N} \parallel_0 \mathcal{N}) \parallel_A Q\]  
9504 states

- **Exploit symmetry by using multisets:**
  \[\{s, s, u\}\] instead of $(s, s, u), (s, u, s), (u, s, s)$

\[(\|-^3_0 \mathcal{N}) \parallel_A Q\]  
2464 states
Confluence (aka: Partial-Order) Reduction

[Timmer et al., 2015]

- Confluence reduction in a nutshell
  - State space reduction technique based on commutativity of transitions
  - Remove spurious non-determinism resulting from independent $||$
  - Construct a subset of the invisible transitions satisfying the confluence restrictions.
  - Choose a representative state for each state in the original system.
  - Skip confluent transitions until reaching a representative state
Confluence (aka: Partial-Order) Reduction [Timmer et al., 2015]

- Confluence reduction in a nutshell
  - State space reduction technique based on commutativity of transitions
  - Remove spurious non-determinism resulting from independent \( || \)
  - Construct a subset of the invisible transitions satisfying the confluence restrictions.
  - Choose a representative state for each state in the original system.
  - Skip confluent transitions until reaching a representative state

- Confluence links divergence-sensitive branching bisimilar states
Confluence (aka: Partial-Order) Reduction

[Timmer et al., 2015]

- **Confluence reduction in a nutshell**
  - State space reduction technique based on *commutativity* of transitions
  - Remove *spurious non-determinism* resulting from independent ∥
  - Construct a subset of the invisible transitions satisfying the confluence restrictions.
  - Choose a representative state for each state in the original system.
  - Skip confluent transitions until reaching a representative state

- Confluence links *divergence-sensitive branching bisimilar* states

- Confluence is detected *symbolically* on model descriptions
  - based on conservative modelling-language-specific heuristics
Confluence (aka: Partial-Order) Reduction [Timmer et al., 2015]

- Confluence reduction in a nutshell
  - State space reduction technique based on commutativity of transitions
  - Remove spurious non-determinism resulting from independent \( \parallel \)
  - Construct a subset of the invisible transitions satisfying the confluence restrictions.
  - Choose a representative state for each state in the original system.
  - Skip confluent transitions until reaching a representative state

- Confluence links divergence-sensitive branching bisimilar states

- Confluence is detected symbolically on model descriptions
  - based on conservative modelling-language-specific heuristics

- On-the-fly reduction while generating the state space
Main Principle of Confluence Reduction
## Experimental Results

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Original state space</th>
<th>Reduction</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Gen. Analysis</td>
<td>Gen. Analysis</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1e-3-7</td>
<td>S: 25,505, P: 34,257</td>
<td>4.7, 103</td>
<td>5.1, 9</td>
</tr>
<tr>
<td>1e-3-9</td>
<td>S: 52,465, P: 71,034</td>
<td>9.7, 212</td>
<td>10.4, 18</td>
</tr>
<tr>
<td>1e-3-11</td>
<td>S: 93,801, P: 127,683</td>
<td>18.0, 429</td>
<td>19.2, 32</td>
</tr>
<tr>
<td>1e-4-3</td>
<td>S: 35,468, P: 50,612</td>
<td>9.0, 364</td>
<td>8.7, 33</td>
</tr>
<tr>
<td>1e-4-4</td>
<td>S: 101,261, P: 148,024</td>
<td>25.8, 1,310</td>
<td>24, 94.4</td>
</tr>
<tr>
<td>poll-2-2-6</td>
<td>S: 27,651, P: 51,098</td>
<td>12.7, 91</td>
<td>5.4, 49</td>
</tr>
<tr>
<td>poll-2-5-2</td>
<td>S: 27,659, P: 47,130</td>
<td>4.0, 1,572</td>
<td>4.0, 1,054</td>
</tr>
<tr>
<td>poll-4-6-1</td>
<td>S: 15,439, P: 29,506</td>
<td>3.1, 331</td>
<td>3.0, 109</td>
</tr>
<tr>
<td>poll-5-4-1</td>
<td>S: 21,880, P: 43,760</td>
<td>5.1, 816</td>
<td>5.1, 318</td>
</tr>
<tr>
<td>proc-3</td>
<td>S: 10,852, P: 20,872</td>
<td>3.1, 66</td>
<td>3.3, 23</td>
</tr>
<tr>
<td>proc-4</td>
<td>S: 31,832, P: 62,356</td>
<td>10.8, 925</td>
<td>10.3, 366</td>
</tr>
</tbody>
</table>

2.4 GHz 4 GB Intel Core 2 Duo MacBook

CR removes 90% of states that are probabilistically branching bisimilar\(^2\)

\(^2\)Checked using the tool CADP.