Petri Net (versus) State Spaces

Karsten Wolf
Why

State Space:
- Asynchronous communication!

- Global changes $\rightarrow$ Local changes!

- Sequential order $\rightarrow$ Causality!
Petri net principles

**Monotonicity** of firing

- Asynchronous communication!

**Linearity** of firing rule

- Global changes → Local changes!

**Locality**

- Sequential order → Causality!

**Partially ordered event structures**

Variables → Ressources
Petri net specific verification

Monotonicity of firing

Linearity of firing rule

Locality

Partially ordered event structures

Coverability graphs ... available in LoLA

Invariants ... used in LoLA

Net reduction

Branching prefixes
Explicit State Reduction Techniques in the LoLA tool

Symmetry          Sweep-Line          Coverability          PN Structure Theory

Partial Order
1. PN Structure Theory

- The Petri net state equation:

\[
\text{If } m \rightarrow^* m' \text{ then } \quad N x = (m - m') \quad \text{has a solution}
\]

Tool **Sara:**

- Search *state* space

→

- Search *solution* space
2. The sweep-line method

- Relies on progress measure

LoLA computes measure automatically:
3. The symmetry method

LoLA: Symmetry = graph automorphism of the PT-Net
Example

Karsten-Wolfs-MacBook-Pro:demo_munich karsten$ ./lola --check=full --symmetry data17.llnet
lola: reading net from data17.llnet
lola: finished parsing
lola: closed net file data17.llnet
lola: 162/65536 symbol table entries, 0 collisions
lola: preprocessing net
lola: computing forward-conflicting sets
lola: computing back-conflicting sets
lola: 55 transition conflict sets
lola: finding significant places
lola: 90 places, 72 transitions, 36 significant places
lola: computing symmetries (--symmetry)
lola: computed 306 generators (35 in search tree, 271 by composition)
lola: representing 4.09904E+31 symmetries
lola: 0 dead branches visited in search tree
lola: using a bit-perfect encoder (--encoder)
lola: using 144 bytes per marking, with 0 unused bits
lola: using a prefix store (--store)
lola: killed reporter thread
lola: building the complete state space (--check=full)
lola: finished preprocessing
lola: result: no
lola: 20 markings, 361 edges
Karsten-Wolfs-MacBook-Pro:demo_munich karsten$
Example

Karsten-Wolfs-MacBook-Pro:demo_munich karsten$ ./lola --check=full --symmetry data17.llnet
lola: reading net from data17.llnet
lola: finished parsing
lola: closed net file data17.llnet
lola: 162/65536 symbol table entries, 0 collisions
lola: preprocessing net
lola: computing forward-conflicting sets
lola: computing back-conflicting sets
lola: 55 transition conflict sets
lola: finding significant places
lola: 90 places, 72 transitions, 36 significant places
lola: computing symmetries (--symmetry)
lola: computed 306 generators (35 in search tree, 271 by composition)
lola: representing 4.09904E+31 symmetries
lola: 0 dead branches visited in search tree
lola: using a bit-perfect encoder (--encoder)
lola: using 144 bytes per marking, with 0 unused bits
lola: using a prefix store (--store)
lola: killed reporter thread
lola: building the complete state space (--check=full)
lola: finished preprocessing
lola: result: no
lola: 20 markings, 361 edges
Karsten-Wolfs-MacBook-Pro:demo_munich karsten$
Example

Karsten-Wolfs-MacBook-Pro:demo_munich karsten$ ./lola --check=full --symmetry data17.llnet
lola: reading net from data17.llnet
lola: finished parsing
lola: closed net file data17.llnet
lola: 162/65536 symbol table entries, 0 collisions
lola: preprocessing net
lola: computing forward-conflicting sets
lola: computing back-conflicting sets
lola: 55 transition conflict sets
lola: finding significant places
lola: 90 places, 72 transitions, 36 significant places
lola: computing symmetries (--symmetry)
lola: computed 306 generators (35 in search tree, 271 by composition)
lola: representing 4.09904E+31 symmetries
lola: 0 dead branches visited in search tree
lola: using a bit-perfect encoder (--encoder)
lola: using 144 bytes per marking, with 0 unused bits
lola: using a prefix store (--store)
lola: killed reporter thread
lola: building the complete state space (--check=full)
lola: finished preprocessing
lola: result: no
lola: 20 markings, 361 edges
Karsten-Wolfs-MacBook-Pro:demo_munich karsten$
Example

Karsten-Wolfs-MacBook-Pro:demo_munich karsten$ ./lola --check=full --symmetry data17.llnet
lola: reading net from data17.llnet
lola: finished parsing
lola: closed net file data17.llnet
lola: 162/65536 symbol table entries, 0 collisions
lola: preprocessing net
lola: computing forward-conflicting sets
lola: computing back-conflicting sets
lola: 55 transition conflict sets
lola: finding significant places
lola: 90 places, 72 transitions, 36 significant places
lola: computing symmetries (--symmetry)

lola: computed 306 generators (35 in search tree, 271 by composition)

lola: representing 4.09904E+31 symmetries
lola: 0 dead branches visited in search tree
lola: using a bit-perfect encoder (--encoder)

lola: using 144 bytes per marking, with 0 unused bits
lola: using a prefix store (--store)

lola: killed reporter thread
lola: building the complete state space (--check=full)

lola: finished preprocessing

lola: result: no
lola: 20 markings, 361 edges
Karsten-Wolfs-MacBook-Pro:demo_munich karsten$
# Functional Verification of Task Partitioning for Multiprocessor Embedded Systems

DIPANKAR DAS, P. P. CHAKRABARTI, and RAJEEV KUMAR  
Indian Institute of Technology Kharagpur

<table>
<thead>
<tr>
<th></th>
<th>nodes UML</th>
<th>places Petri Net</th>
<th>transitions Petri Net</th>
<th>Spin</th>
<th>LoLA</th>
<th>PROD</th>
<th>PEP</th>
<th>TraceMatch</th>
</tr>
</thead>
<tbody>
<tr>
<td>crc32_T</td>
<td>70</td>
<td>90</td>
<td>30</td>
<td>580 ms</td>
<td>10 ms</td>
<td>2170 ms</td>
<td>30 ms</td>
<td>1 ms</td>
</tr>
<tr>
<td>crc32_F</td>
<td>70</td>
<td>90</td>
<td>30</td>
<td>600 ms</td>
<td>10 ms</td>
<td>2200 ms</td>
<td>30 ms</td>
<td>2 ms</td>
</tr>
<tr>
<td>adpcm_T</td>
<td>85</td>
<td>102</td>
<td>66</td>
<td>630 ms</td>
<td>10 ms</td>
<td>2340 ms</td>
<td>20 ms</td>
<td>2 ms</td>
</tr>
<tr>
<td>adpcm_F</td>
<td>85</td>
<td>102</td>
<td>66</td>
<td>590 ms</td>
<td>10 ms</td>
<td>2420 ms</td>
<td>20 ms</td>
<td>2 ms</td>
</tr>
<tr>
<td>fft_T</td>
<td>98</td>
<td>76</td>
<td>106</td>
<td>600 ms</td>
<td>&lt;10 ms</td>
<td>2520 ms</td>
<td>20 ms</td>
<td>1 ms</td>
</tr>
<tr>
<td>fft_F</td>
<td>98</td>
<td>76</td>
<td>106</td>
<td>640 ms</td>
<td>20 ms</td>
<td>2660 ms</td>
<td>20 ms</td>
<td>3 ms</td>
</tr>
<tr>
<td>basicmath_T</td>
<td>265</td>
<td>320</td>
<td>157</td>
<td>1700 ms</td>
<td>20 ms</td>
<td>11300 ms</td>
<td>170 ms</td>
<td>9 ms</td>
</tr>
<tr>
<td>basicmath_F</td>
<td>265</td>
<td>320</td>
<td>157</td>
<td>1880 ms</td>
<td>40 ms</td>
<td>12420 ms</td>
<td>180 ms</td>
<td>27 ms</td>
</tr>
<tr>
<td>dijkstra_T</td>
<td>285</td>
<td>302</td>
<td>229</td>
<td>1340 ms</td>
<td>20 ms</td>
<td>8570 ms</td>
<td>140 ms</td>
<td>23 ms</td>
</tr>
<tr>
<td>dijkstra_F</td>
<td>285</td>
<td>302</td>
<td>229</td>
<td>1330 ms</td>
<td>80 ms</td>
<td>8750 ms</td>
<td>140 ms</td>
<td>27 ms</td>
</tr>
<tr>
<td>sha_T</td>
<td>309</td>
<td>314</td>
<td>239</td>
<td>2050 ms</td>
<td>20 ms</td>
<td>13640 ms</td>
<td>200 ms</td>
<td>8 ms</td>
</tr>
<tr>
<td>sha_F</td>
<td>309</td>
<td>314</td>
<td>239</td>
<td>2020 ms</td>
<td>50 ms</td>
<td>13550 ms</td>
<td>210 ms</td>
<td>15 ms</td>
</tr>
</tbody>
</table>
Figure 1 shows the result of the Petri net query corresponding to prop3 found by the LoLA [13] Petri net analysis tool, as displayed by PLA using the Pathalyzer tool. Lola uses “stubborn set reduction”, which is a technique that exploits the ease of determining the independence of certain transitions in the Petri nets. For reachability queries on our nets, answering a reachability query that would have taken hours using a general purpose model-checking tool takes on the order of a second in LoLA—fast enough to permit interactive use.
Commercial III

Global trophies for ReachabilityMix

For this trophy, we use the following formula: results on “Known” models + 2 x results on “Surprise” models.

Trophies for All Models

<table>
<thead>
<tr>
<th>Trophy</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>LoLA optimistic</td>
<td>313 (points)</td>
</tr>
<tr>
<td>LoLA</td>
<td>209 (points)</td>
</tr>
<tr>
<td>Merci</td>
<td>166 (points)</td>
</tr>
<tr>
<td>LoLA optimistic</td>
<td>incomplete 166 (points)</td>
</tr>
</tbody>
</table>

Tool classification for Reachability examinations

<table>
<thead>
<tr>
<th>Tool</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>curf</td>
<td>14391</td>
</tr>
<tr>
<td>greatapn</td>
<td>1318</td>
</tr>
<tr>
<td>helena</td>
<td>12802</td>
</tr>
<tr>
<td>lola</td>
<td>27200</td>
</tr>
<tr>
<td>praxdd</td>
<td>112</td>
</tr>
<tr>
<td>tapaai</td>
<td>24445</td>
</tr>
</tbody>
</table>
Applications

- Hazards in asynchronous circuits
- Information flow security in web services
- AI planning for web service composition
- Soundness in business processes
- Biochemical reaction chains
- Multiprocessor embedded systems
- Parameterized problems
Conclusion

Further reading:
• Tools:  www.service-technology.org
• Group / Papers:  www.informatik.uni-rostock.de/tpp/