REDUCED STATE SPACE CONSTRUCTION
- STUBBORN SET REDUCED REACHABILITY GRAPH

QUALITATIVE ANALYSIS METHODS, OVERVIEW

- NET REDUCTION
- STRUCTURAL PROPERTIES
- LINEAR PROGRAMMING
  - place / transition invariants
  - state equation
  - trap equation
- REACHABILITY ANALYSIS
  - (complete) reachability graph
  - compressed state spaces
    - BDDs, NDDs, ..., XDDs
    - Kronecker products
  - reduced state spaces
    - coverability graph
    - symmetry
    - stubborn sets
  - branching process
### STUBBORN SETS & REDUCED RG

- **basic principle** - *lazy state space construction*
  - only a subset of the complete rg is constructed
  - this subset still allows the decision of certain properties
  - \( \text{RG}_{\text{red}} \equiv \text{RG} \) equivalent with respect to some properties
  - suitable equivalence relation?

- **basic idea** - *partial order reduction techniques*
  - not all interleaving sequences of concurrent behavior (= partially ordered behavior) are considered

- **preserved properties**
  - all dead states
  - cyclic behavior

---

**EXAMPLE SYSTEM DEADLOCK, PETRI NET**

![Petri Net Diagram]

- **IN**
  - ORD, HOM, NBM, PUR, CSV, SCF, CON, SC, Ft, F0, Fp0, MG, SM, FC, RFC, ES
  - Y Y Y Y N Y N N N N N N N N
  - DTP, SMC, SMD, SMA, CPI, CTI, B, SB, REV, BSt, DSt, DCF, L, LV, L4S
  - N Y Y Y Y Y Y Y Y Y Y N N N N

---

monika.heiner@informatik.tu-cottbus.de
EXAMPLE
SYSTEM DEADLOCK, (COMPLETE) RG

CONCURRENCY BRANCHING

CONFLICT BRANCHING

DEAD STATE

19 nodes, 32 arcs

EXAMPLE, SYSTEM DEADLOCK, REDUCED RG

10 nodes, 12 arcs
STUBBORN SET, CHARACTERISTICS

- a marking-dependent selection of a set of independent transitions
  - a set of independent transitions
    - their behavior cannot be influenced by the excluded transitions
    - "they are stubborn"
    - any sequence of excluded transitions cannot enable or disable an included transition
    - their firing can be postponed
    - contains at least one enabled transition

- stubborn set reduced rg
  - slight variation of the standard algorithm
  - at each marking (node):
    - instead of firing all enabled transitions, only transitions of a stubborn set are fired

REACHABILITY GRAPH, CONSTRUCTION ALGORITHM

PROCEDURE rg (IN Net pn, IN Marking m0, OUT MSet nodes, OUT ArcSet arcs);

MSet $U = \{ m0 \}$; // unprocessed markings
$N = \emptyset$; // rg nodes
ArcSet $E = \emptyset$; // rg arcs (pre, post, t)
Marking $m'$; // successor marking
Transition $t$;

WHILE $U \neq \emptyset$ DO
  choose one $m \in U$;
  $U = U - \{ m \}$; $N = N \cup \{ m \}$;

  FOR ALL $t$ enabled at $m$ DO
    $m' = m + \Delta t$;
    IF $m' \notin N \cup U$ // new marking
      THEN $U = U \cup \{ m' \}$
    ENDIF;
    $E = E \cup \{ (m, m', t) \}$
  ENDFOR

ENDWHILE;

nodes = $N$; arcs = $E$;

ENDPROC rg.
STUBBORN REDUCED RG, CONSTRUCTION ALGORITHM

PROCEDURE rg (IN Net pn, IN Marking m0, 
OUT MSet nodes, OUT ArcSet arcs);

MSet \( U = \{m0\} \), // unprocessed markings 
N = \( \emptyset \); // rg nodes
ArcSet \( E = \emptyset \); // rg arcs (pre, post, t)
Marking \( m' \); // successor marking
Transition \( t \);

WHILE \( U \neq \emptyset \) DO
    choose one \( m \in U \);
    \( U = U – \{m\} \); \( N = N \cup \{m\} \);

    FOR ALL enabled \( t \) of a stubborn set at \( m \) DO
        \( m' = m + \Delta t \);
        \( \text{IF } m' \in N \cup U \) // new marking
        THEN \( U = U \cup \{m'\} \)
        ENDIF;
        \( E = E \cup \{(m, m', t)\} \)
    ENDFOR

ENDWHILE;

nodes = N; arcs = E;

ENDPROC rg.

HOW TO CONSTRUCT STUBBORN SETS

- three basic steps

  1. choose an enabled transition \( t \) and put it into \( U \)

  2. FOR ALL enabled transition \( t \) in \( U \):
     - put into \( U \) also all transitions in conflict with \( t \)
     - conflict transitions: \((F t)F\)
     - any sequence of excluded transitions cannot disable an included transition

  3. FOR ALL disabled transition \( t \) in \( U \):
     - choose a scapegoat
     - (a place \( p \) which prevents \( t \)
          from being enabled),
     - and put all pre-transitions of \( p \) \((F p)\) into \( U \)
     - any sequence of excluded transitions cannot enable an included transition

- repeat (2) and (3) as long as necessary
STUBBORN SETS, EXAMPLES (1)

- any conflict-free enabled transition -> is a stubborn set for itself
- for any dead state -> there is no stubborn set
- for non-dead states -> set of all transitions is a stubborn set
STUBBORN SETS, OBSERVATIONS

- Each set $U$ constructed by this way is a stubborn set at $m$
- Result $U$ depends on the current marking $m$
- Non-deterministic stubborn set construction
  - Result depends on non-deterministic choices
  - Choose an enabled transition $t$
  - Choose a scapegoat $p$
- Smaller stubborn sets result generally into smaller reduced $rg$
  - BUT, there are counter examples
- There are various heuristics to determine smaller stubborn sets, -> basic step (3)
- At best: minimal stubborn sets (contain no stubborn subset)
- BUT, increasing computational effort
  - May exceed benefit gained
  - What is more worth: space or run time?
EXAMPLE
TRAVEL PLANING

- begin_preparations
- camping, hotel
- ask_for_rooms
- train, car, feet
- ask_for_schedule
- check
- pack
- repeat
- end_preparations

-> only one interleaving sequence is represented

DINING PHILOSOPHERS,
RG AND REDUCED RG SIZES

<table>
<thead>
<tr>
<th># phils</th>
<th>P / T</th>
<th>Rstub</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6 / 4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>10 / 8</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>15 / 12</td>
<td>20</td>
<td>35</td>
</tr>
<tr>
<td>4</td>
<td>20 / 16</td>
<td>38</td>
<td>118</td>
</tr>
<tr>
<td>5</td>
<td>25 / 20</td>
<td>62</td>
<td>392</td>
</tr>
<tr>
<td>6</td>
<td>30 / 24</td>
<td>92</td>
<td>1.297</td>
</tr>
<tr>
<td>7</td>
<td>35 / 28</td>
<td>128</td>
<td>4.286</td>
</tr>
<tr>
<td>8</td>
<td>40 / 32</td>
<td>170</td>
<td>14.158</td>
</tr>
<tr>
<td>9</td>
<td>45 / 36</td>
<td>218</td>
<td>46.763</td>
</tr>
<tr>
<td>10</td>
<td>50 / 40</td>
<td>272</td>
<td>154.450</td>
</tr>
<tr>
<td>11</td>
<td>55 / 44</td>
<td>332</td>
<td>510.116</td>
</tr>
<tr>
<td>12</td>
<td>60 / 48</td>
<td>398</td>
<td>1.56 e+6</td>
</tr>
<tr>
<td>13</td>
<td>65 / 52</td>
<td>470</td>
<td>(5.56 e+6)</td>
</tr>
<tr>
<td>14</td>
<td>70 / 56</td>
<td>548</td>
<td>632</td>
</tr>
</tbody>
</table>
### PRODUCTION CELL, COOPERATION MODEL

<table>
<thead>
<tr>
<th>Places/Transitions</th>
<th>DTP</th>
<th>$R_{stub}$</th>
<th>$R$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Table/Press</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>with init part</td>
<td>13 / 9</td>
<td>(N)</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>12 / 8</td>
<td>28</td>
<td>8</td>
</tr>
<tr>
<td>without init part</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Crane</strong></td>
<td>12 / 8</td>
<td>31</td>
<td>11</td>
</tr>
<tr>
<td><strong>Arms</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>version 1</td>
<td>13 / 8</td>
<td>38</td>
<td>11</td>
</tr>
<tr>
<td>version 2</td>
<td>17 / 12</td>
<td>109</td>
<td>15</td>
</tr>
<tr>
<td>version 3</td>
<td>17 / 12</td>
<td>88</td>
<td>15</td>
</tr>
<tr>
<td><strong>Belts</strong></td>
<td>12 / 8</td>
<td>26</td>
<td>8</td>
</tr>
<tr>
<td><strong>Subsystem with Arm Version 1</strong></td>
<td>25 / 16</td>
<td>175</td>
<td>47</td>
</tr>
<tr>
<td></td>
<td>33 / 24</td>
<td>3.851 (N)</td>
<td>75</td>
</tr>
<tr>
<td><strong>Open System</strong></td>
<td>51 / 36</td>
<td>1.145</td>
<td>299</td>
</tr>
</tbody>
</table>

### PRODUCTION CELL, CONTROL MODEL

<table>
<thead>
<tr>
<th>System Part</th>
<th>P / T</th>
<th>PROD</th>
<th>R</th>
<th>$R_{stub}$ a)</th>
<th>time</th>
<th>$R_{stub}$ b)</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Controllers</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Crane</td>
<td>45/34</td>
<td>256</td>
<td>0.78&quot;</td>
<td>51</td>
<td>0.16&quot;</td>
<td>38</td>
<td>0.08&quot;</td>
</tr>
<tr>
<td>Feed Belt</td>
<td>22/16</td>
<td>69</td>
<td>0.20&quot;</td>
<td>31</td>
<td>0.10&quot;</td>
<td>16</td>
<td>0.07&quot;</td>
</tr>
<tr>
<td>Table</td>
<td>32/24</td>
<td>88</td>
<td>0.38&quot;</td>
<td>36</td>
<td>0.15&quot;</td>
<td>24</td>
<td>0.09&quot;</td>
</tr>
<tr>
<td>Arm, V3</td>
<td>66/60</td>
<td>365&quot;</td>
<td>1.19&quot;</td>
<td>62</td>
<td>0.23&quot;</td>
<td>51</td>
<td>0.09&quot;</td>
</tr>
<tr>
<td>Press</td>
<td>28/20</td>
<td>140</td>
<td>0.42&quot;</td>
<td>48</td>
<td>0.10&quot;</td>
<td>20</td>
<td>0.09&quot;</td>
</tr>
<tr>
<td>Deposit Belt</td>
<td>22/16</td>
<td>69</td>
<td>0.20&quot;</td>
<td>31</td>
<td>0.11&quot;</td>
<td>16</td>
<td>0.07&quot;</td>
</tr>
</tbody>
</table>

**Composed Systems**

<table>
<thead>
<tr>
<th>System Part</th>
<th>P / T</th>
<th>PROD</th>
<th>R</th>
<th>$R_{stub}$ a)</th>
<th>time</th>
<th>$R_{stub}$ b)</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Robot</td>
<td>124/120</td>
<td>63,232</td>
<td>11.26</td>
<td>992</td>
<td>5.99&quot;</td>
<td>205</td>
<td>0.21&quot;</td>
</tr>
<tr>
<td>Robot/Press</td>
<td>140/132</td>
<td>18,344</td>
<td>3.10&quot;</td>
<td>557</td>
<td>3.46&quot;</td>
<td>305</td>
<td>0.35&quot;</td>
</tr>
<tr>
<td>Open System</td>
<td>198/176</td>
<td>2,776,936</td>
<td>?</td>
<td>798</td>
<td>5.90&quot;</td>
<td>507</td>
<td>0.62&quot;</td>
</tr>
</tbody>
</table>

**Closed Systems**

<table>
<thead>
<tr>
<th>System Part</th>
<th>P / T</th>
<th>PROD</th>
<th>R</th>
<th>$R_{stub}$ a)</th>
<th>time</th>
<th>$R_{stub}$ b)</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 plate</td>
<td>231/202</td>
<td>30,952</td>
<td>7.54&quot;</td>
<td>162</td>
<td>0.68&quot;</td>
<td>163</td>
<td>0.32&quot;</td>
</tr>
<tr>
<td>2 plates</td>
<td></td>
<td>543,480</td>
<td>3.3 h</td>
<td>406</td>
<td>2.53&quot;</td>
<td>456</td>
<td>0.72&quot;</td>
</tr>
<tr>
<td>3 plates</td>
<td></td>
<td>&gt; 1.7 Mio</td>
<td>&gt;20 h</td>
<td>523</td>
<td>4.51&quot;</td>
<td>635</td>
<td>0.95&quot;</td>
</tr>
<tr>
<td>4 plates</td>
<td></td>
<td>&gt; 3.1 Mio</td>
<td>&gt;42 h</td>
<td>471</td>
<td>4.02&quot;</td>
<td>678</td>
<td>1.06&quot;</td>
</tr>
<tr>
<td>5 plates</td>
<td></td>
<td>1,687,242</td>
<td>14 h</td>
<td>585</td>
<td>5.05&quot;</td>
<td>608</td>
<td>0.98&quot;</td>
</tr>
</tbody>
</table>

**a)** deletion algorithm  
**b)** incremental algorithm
### Example: Pushers

<table>
<thead>
<tr>
<th># pushers</th>
<th>R</th>
<th>version 1</th>
<th>version 1</th>
<th>version 2</th>
<th>version 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>P / T</td>
<td>R_{stub}</td>
<td>P / T</td>
<td>R_{stub}</td>
</tr>
<tr>
<td>1</td>
<td>88</td>
<td>24 / 25</td>
<td>22</td>
<td>24 / 21</td>
<td>22</td>
</tr>
<tr>
<td>2</td>
<td>464</td>
<td>42 / 46</td>
<td>42</td>
<td>42 / 38</td>
<td>42</td>
</tr>
<tr>
<td>3</td>
<td>3.088</td>
<td>60 / 67</td>
<td>79</td>
<td>60 / 55</td>
<td>79</td>
</tr>
<tr>
<td>4</td>
<td>18.848</td>
<td>78 / 88</td>
<td>133</td>
<td>78 / 72</td>
<td>133</td>
</tr>
<tr>
<td>5</td>
<td>118.624</td>
<td>96 / 109</td>
<td>204</td>
<td>96 / 89</td>
<td>204</td>
</tr>
<tr>
<td>6</td>
<td>0.7 e+6</td>
<td>114 / 130</td>
<td>292</td>
<td>114 / 106</td>
<td>292</td>
</tr>
<tr>
<td>7</td>
<td>4.6 e+6</td>
<td>132 / 151</td>
<td>397</td>
<td>132 / 123</td>
<td>397</td>
</tr>
<tr>
<td>8</td>
<td>28.9 e+6</td>
<td>150 / 172</td>
<td>519</td>
<td>150 / 140</td>
<td>519</td>
</tr>
<tr>
<td>9</td>
<td>179.8 e+6</td>
<td>168 / 193</td>
<td>658</td>
<td>168 / 157</td>
<td>658</td>
</tr>
<tr>
<td>10</td>
<td>1.1 e+9</td>
<td>186 / 214</td>
<td>814</td>
<td>186 / 174</td>
<td>814</td>
</tr>
</tbody>
</table>

- version 1 - many dynamic conflicts
- version 2 - persistent

### Summary, Outlook

- **reduction effect needs**
  - concurrently enabled transitions
  - more than one
  - no conflict in between
- for system without concurrency
  - RG = RG_{red}
- on-the-fly model checking of LTL\(\Box\)
REFERENCES

[Gerth 95]

[Godefroid 96]

[Pogrell 95]

[Starke 92]

[Valmari 92]

[Valmari 92]

[Varpaaniemi 95]