Büchi-Automata guided Partial Order Reduction for LTL

Torsten Liebke
Universität Rostock, Germany
Success of Partial Order Reduction (POR)

POR is one of the most powerful reduction techniques
LoLA won the LTL category in the MCC for the last 3 years, 2017 - 2019

LoLA's performance in the MCC

- No POR: 70%
- POR: 90%
Partial order reduction: The stubborn set method

**Given:** Petri net $N = [P,T,F,W,m_0]$ and property $\phi$

**Goal:** produce subgraph of the reachability graph

**Condition:** evaluation of $\phi$ yields same result

In any given marking only a subset of transitions is explored.
Basic approach of LTL model checking

Given: Petri net and Büchi automaton

1. Reduce reachability graph (e.g., with POR)
2. Construction of product automaton with reduced reachability graph and the Büchi automaton
3. Decision whether language is empty
Idea

Conventional:
1. compute stubborn set based on the marking
2. build product automaton

New approach:
1. build product automaton
2. compute stubborn set based on the marking and the Büchi state

Restrict the scope of the property to the current Büchi state $q$. 

Büchi-Automata guided Partial Order Reduction for LTL

Torsten Liebke
Principles

- There exists a list of principles to build the reduced reachability graph, the subgraph.
- Based on the selected principles, properties of a certain class are preserved.

In the following:
\[ \pi' = \text{Path in subgraph} \]
\[ \pi = \text{Path in original graph} \]
COM: The commutativity principle

- Transitions may be executed in another order
- Can shift transitions to the front of the path
KEY: The key transition principle

• Transition $t$ stays enabled
• Can push transition to the right
VIS: The visibility principle

- VIS ensures the order of transitions, visible for $\phi$, does not change.
- Visible transitions in $\pi'$ appear in the same order as in $\pi$, if they appear in $\pi'$. 
IGN: The non-ignoring principle

• All transitions are fired at least once in every circle
• Ensures that all transitions of $\pi$ are eventually occurring in $\pi'$
Preservation of Linear Time Logic

Simplify or drop some of these requirements.

Büchi-Automata guided Partial Order Reduction for LTL

Torsten Liebke
NLG: The non-leaving principle

No transition sequence, from the set of transitions which are not in the stubborn set, can satisfy a progressing formula \((\varphi_1 - \varphi_h)\).
Simplification of VIS: S-INV: The semi-invisibility principle

- All considerations regarding the property are done locally
- Temporal operators expressed within the Büchi automaton
- A satisfied formula has no influence on other state transitions
- Transition can changes $\varphi$ from true to false: not allowed
- Transition can make $\varphi$ only truer: allowed

$\rightarrow$ Simplification of VIS to an implication (semi-invisible)

$\varphi = p_2 \leq 0$
Dropping of IGN

Reduction is only applied within a Büchi state → stuttering becomes irrelevant.

Finite stuttering within a Büchi state does not change the accepting behaviour.

Infinite stuttering can always be avoided, if this was possible in the original product automaton, due to the reduction principle.

Side note:
The X-operator can now also be verified.
Simplification of KEY

In accepting states infinite stuttering is possible and also desired.
→ KEY has to hold only in accepting states.
Main result: automata based POR

1. stubborn\((m, q)\) satisfies COM
2. stubborn\((m, q)\) satisfies S-INV with respect to \(\psi\)
3. stubborn\((m, q)\) = \(T\) or \(\forall t \in \text{stubborn}(m, q): m \rightarrow m' \implies m' \models \psi\)
4. stubborn\((m, q)\) satisfies NLG
5. IF \(q \in Q_F\), then stubborn\((m, q)\) satisfies KEY
Example Petri net

- System of $n$ concurrent processes
- State space consists of $2^n$ markings with $2 \cdot n \cdot 2^n$ edges
Example 1: $\varphi_1 = F(\land_{j=1}^n o_j = 1)$

Conventional:
- Every transition is visible to $\varphi_1$
- Each transition $t_j$ produces a token on $o_j$
- No reduction
- Product automaton has $2^n + 2$ reachable states: initial state, $2^n$ states in $q_0$, one state in $q_1$
Example 1: $\varphi_1 = F(\bigwedge_{j=1}^{n} o_j = 1)$

Conventional: $2^n + 2$

Automaton based:

- Every transition is semi-invisible in $q_0$ and $q_1$
- Retarding formula is always true
- In $q_0$ every singleton set $\{t_j\}$ is a valid stubborn set
- Reduced product automaton has $n + 3$ reachable states: initial state, $n + 1$ chain-shaped states in $q_0$, one state in $q_1$
Example 2: $\varphi_2 = F(i_1 = 2 \land \bigwedge_{j=2}^{n} o_j = 1)$

Conventional:
- $t_1$ is invisible, all $t_2, ..., t_n$ are visible to $\varphi_2$
- In the initial state $t_1$ is a valid stubborn set
- Note that in $q_1$ no state is reachable since $i_1 = 2$ can never be satisfied
- Product automaton has $2^{n-1} + 2$ reachable states: initial state, $2^{n-1}$ states in $q_0$, one state in $q_1$
Example 2: \( \varphi_2 = F(i_1 = 2 \land \land_{j=2}^{n} o_j = 1) \)

Conventional: \(2^{n-1} + 2\)

Automaton based:

- Only in accepting Büchi states an activated transition has to be in the stubborn set (KEY)
- \(q_0\) is not accepting \(\rightarrow\) empty set is a valid stubborn set
- Product automaton has 2 reachable states: initial state, one state in \(q_0\)
## Comparison

<table>
<thead>
<tr>
<th></th>
<th>Conventional</th>
<th>Automata based</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stubborn set</td>
<td>$M \rightarrow 2^T$</td>
<td>$M \times Q \rightarrow 2^T$</td>
</tr>
<tr>
<td>X-operator</td>
<td>Not supported</td>
<td>Improved: supported</td>
</tr>
<tr>
<td>VIS</td>
<td>✔</td>
<td>Improved: weaker S-INV</td>
</tr>
<tr>
<td>IGN</td>
<td>✔</td>
<td>Improved: not required</td>
</tr>
<tr>
<td>KEY</td>
<td>✔</td>
<td>Improved: acc. states only</td>
</tr>
<tr>
<td>NLG</td>
<td>Not required</td>
<td>✔</td>
</tr>
</tbody>
</table>
Summary

- Using current Büchi state for stubborn set computation
- Simplified VIS, KEY. Dropped IGN
- No restriction to LTL-X. Promising reduction efficiency