Satisfiability Checking for the Coalgebraic $\mu$-Calculus

Daniel Hausmann
daniel.hausmann@fau.de
Chair for Theoretical Computer Science, Friedrich-Alexander-Universität Erlangen-Nürnberg

Automata on Infinite Words

Co-Büchi automata (CBA): From some point on, only accepting states are visited
Büchi automata (BA): Some accepting state is visited infinitely often
Parity automata (PA): Highest priority that is visited infinitely often is even

Limit-stationary CBA: Accepting SCCs are single states
Limit-linear CBA: Accepting SCCs are linear
Limit-deterministic BA/PA: Accepting SCCs are deterministic

Example Automata

CBA:

limit-stationary

accepts: $(a + b)^*b^2$

BA:

limit-linear

accepts: $(a((a + b)(a + b))a)^*$

PA:

limit-deterministic

accepts: $((ac^*) + ba)^*(ba)^*$

Determination Methods

Input: Automaton A of size $n$. Output: Equivalent deterministic automaton B.

<table>
<thead>
<tr>
<th>Type of A</th>
<th>Method</th>
<th>Type of B</th>
<th>Size of B</th>
</tr>
</thead>
<tbody>
<tr>
<td>limit-stationary CBA</td>
<td>focusing method</td>
<td>DCBA</td>
<td>$n \cdot 2^m$</td>
</tr>
<tr>
<td>limit-linear CBA</td>
<td>adaptive focusing method</td>
<td>DCBA</td>
<td>$n^2 \cdot 2^m$</td>
</tr>
<tr>
<td>CBA</td>
<td>Miyana/Hayashi</td>
<td>DCBA</td>
<td>$3^n$</td>
</tr>
<tr>
<td>BA or PA</td>
<td>permutation method</td>
<td>DPA</td>
<td>$n!$ or $(n^2)!$</td>
</tr>
<tr>
<td>BA or PA</td>
<td>Safra/Piterman</td>
<td>DPA</td>
<td>$(n!)^2$ or $(n^2)!$</td>
</tr>
</tbody>
</table>

The Coalgebraic $\mu$-Calculus

Syntax: $\phi, \psi ::= T | T \land T | \phi \lor \psi | \phi \land \psi | X | \nu X.\phi | \mu X.\phi | X \in V$ fixpoint variables, $\land \in \Lambda$ modal operators, e.g. $\Lambda = \{\land, \land\}$

Semantics: Use $T$-coalgebras as models, e.g. for $T = \mathcal{P}$ (powerset), models are Kripke frames $(W, R)$; have e.g. $x \models \phi \iff \exists y \in R(x), y \models \phi$. Fixpoint operators iterate the argument formula finitely ($\mu X$) or infinitely ($\nu X$) often, using $X$ to iterate.

Satisfiability Checking for the Coalgebraic $\mu$-Calculus

Input: Fixpoint formula $\phi$. Decide satisfiability of $\phi$ by solving parity game $G$ played over determined tracking automaton $A(\phi)$.

Syntactic shape of $\phi$, Example formula, Intuition for example formula, Type of $A(\phi)$, Size of $G$

<table>
<thead>
<tr>
<th>Syntactic shape of $\phi$</th>
<th>Example formula</th>
<th>Intuition for example formula</th>
<th>Type of $A(\phi)$</th>
<th>Size of $G$</th>
</tr>
</thead>
<tbody>
<tr>
<td>depth-1 linear</td>
<td>$\mu X.\phi \lor \square X$</td>
<td>&quot;a state satisfying $\phi$ is reachable&quot;</td>
<td>limit-stationary CBA</td>
<td>$n \cdot 2^m$</td>
</tr>
<tr>
<td>linear</td>
<td>$\mu X.\phi \lor \square X$</td>
<td>&quot;$\phi$ is reachable by an even number of steps&quot;</td>
<td>limit-linear CBA</td>
<td>$n^2 \cdot 2^m$</td>
</tr>
<tr>
<td>alternation-free</td>
<td>$\nu X.\phi \land \square X \land \Box X$</td>
<td>&quot;all paths are infinite and $\phi$ holds everywhere&quot;</td>
<td>CBA</td>
<td>$3^n$</td>
</tr>
<tr>
<td>aconjunctive</td>
<td>$\nu X, \mu Y. (\psi \land \Box X) \lor \Box Y$</td>
<td>&quot;on all paths, $\psi$ holds infinitely often&quot;</td>
<td>limit-deterministic PA</td>
<td>$(n!)^2$</td>
</tr>
<tr>
<td>no restriction</td>
<td>$\nu X, \mu Y. (\psi \land \Box X \land (\chi \lor \Box Y))$</td>
<td>&quot;$\psi$ holds everywhere and $\chi$ is always reachable&quot;</td>
<td>PA</td>
<td>$(n!)^2$</td>
</tr>
</tbody>
</table>

Implementation as part of COOL (Coalgebraic Ontology Logic Reasoner)

Solves satisfiability games on-the-fly and in coalgebraic generality; https://www8.cs.fau.de/research/software:cool

Theoretical results of this work have been published in the following papers: