Benchmarks from Reduction Finding

Skip Jordan and Łukasz Kaiser

ERATO Group, JST & Hokkaido University
LIAFA, CNRS & Université Paris Diderot – Paris 7

QBF 2013
Structures and Second-Order Logic

Motivation: Higher Abstraction Level for QBF
Structures and Second-Order Logic

Motivation: Higher Abstraction Level for QBF

Relational Structures $\mathcal{U} = (A, R_1^{\mathfrak{u}}, R_2^{\mathfrak{u}}, \ldots, R_k^{\mathfrak{u}}, c_1^{\mathfrak{u}}, \ldots, c_l^{\mathfrak{u}})$
Structures and Second-Order Logic

Motivation: Higher Abstraction Level for QBF

Relational Structures \( \mathcal{U} = (A, R_1, R_2, \ldots, R_k, c_1, \ldots, c_l) \)

First-Order and Second-Order Logic over \( \sigma = \{E\} \).

The graph is a clique (FO): \( \forall x, y \ (x = y \lor E(x, y)) \)

The graph is 3-colorable (SO):

\[
\exists R, G, B \ (\forall x, y \ (R(x) \lor G(x) \lor B(x)) \land (E(x, y) \rightarrow \\
\neg ((R(x) \land R(y)) \lor (G(x) \land G(y)) \lor (B(x) \land B(y)))))
\]
Model-Checking SO using QBF Solvers

**Transformation:** \( \text{SO} \ni \varphi, \mathcal{A} \leadsto \psi \text{ QBF} \)

\[
\begin{align*}
\text{Rel}(a_1, \ldots, a_k) & \leadsto T / \varphi \perp \\
\text{Var}(a_1, \ldots, a_k) & \leadsto X_{\text{Var},a_1,\ldots,a_k} \\
\varphi_1 \land \varphi_2 & \leadsto \hat{\varphi}_1 \land \hat{\varphi}_2 \\
\exists x \varphi & \leadsto \bigvee_{a \in \mathcal{A}} \hat{\varphi}(a) \\
\forall x \varphi & \leadsto \bigwedge_{a \in \mathcal{A}} \hat{\varphi}(a) \\
\exists V \varphi & \leadsto \exists X_{V,1,1} \ldots X_{V,n,n} \hat{\varphi} \\
\forall V \varphi & \leadsto \forall X_{V,1,1} \ldots X_{V,n,n} \hat{\varphi}
\end{align*}
\]
Model-Checking SO using QBF Solvers

Transformation: \( \text{SO} \ni \varphi, \ A \leadsto \psi \text{ QBF} \)

\[
\begin{align*}
\text{Rel}(a_1, \ldots, a_k) & \leadsto T \setminus \_ \bot \\
\text{Var}(a_1, \ldots, a_k) & \leadsto X_{\text{Var}, a_1, \ldots, a_k} \\
\varphi_1 \land \varphi_2 & \leadsto \hat{\varphi}_1 \land \hat{\varphi}_2 \\
\exists \chi \varphi & \leadsto \bigvee_{a \in A} \hat{\varphi}(a) \\
\forall \chi \varphi & \leadsto \bigwedge_{a \in A} \hat{\varphi}(a) \\
\exists \forall \varphi & \leadsto \exists X_{V, 1, 1} \ldots X_{V, n, n} \hat{\varphi} \\
\forall \forall \varphi & \leadsto \forall X_{V, 1, 1} \ldots X_{V, n, n} \hat{\varphi}
\end{align*}
\]

Steps: (1) construct the QBF (2) use a QBF solver (3) read the answer
Model-Checking SO using QBF Solvers

Transformation: \( \text{SO} \ni \varphi, \mathcal{A} \leadsto \psi \text{ QBF} \)

\[
\begin{align*}
\text{Rel}(a_1, \ldots, a_k) & \leadsto T / \mathcal{A} \perp \\
\text{Var}(a_1, \ldots, a_k) & \leadsto X_{\text{Var}, a_1, \ldots, a_k} \\
\varphi_1 \land \varphi_2 & \leadsto \hat{\varphi}_1 \land \hat{\varphi}_2 \\
\exists x \varphi & \leadsto \bigvee_{a \in \mathcal{A}} \hat{\varphi}(a) \\
\forall x \varphi & \leadsto \bigwedge_{a \in \mathcal{A}} \hat{\varphi}(a) \\
\exists V \varphi & \leadsto \exists X_{V, 1, 1} \ldots X_{V, n, n} \hat{\varphi} \\
\forall V \varphi & \leadsto \forall X_{V, 1, 1} \ldots X_{V, n, n} \hat{\varphi}
\end{align*}
\]

Steps: (1) construct the QBF (2) use a QBF solver (3) read the answer

Example: 3-colouring a graph

http://toss.sf.net/eval.html
Two Basic Applications

Example (and counter-example) finding

\[ \varphi \in \text{FO}, \ n \in \mathbb{N} \quad \rightarrow \quad \mathcal{A} \mid |\mathcal{A}| = n \text{ and } \mathcal{A} \models \varphi \quad (\text{or } \mathcal{A} \models \neg \varphi) \]

**Reduction:** change all relations in \( \varphi \) to SO variables
Two Basic Applications

Example (and counter-example) finding

\[ \varphi \in \text{FO}, \ n \in \mathbb{N} \ \Rightarrow \ \mathcal{A} \ | \ |\mathcal{A}| = n \ \text{and} \ \mathcal{A} \models \varphi \ \text{(or} \ \mathcal{A} \models \neg \varphi) \]

**Reduction:** change all relations in \( \varphi \) to SO variables

Formula finding

**Outline:** formula with Boolean atom guards. Example:

\[
X_1E(x_1, x_1) \ \land \ X_2E(x_1, x_2) \ \land \ X_3E(x_2, x_1) \ \land \ X_4E(x_2, x_2) \ \land \\
X_5\neg E(x_1, x_1) \ \land \ X_6\neg E(x_1, x_2) \ \land \ X_7\neg E(x_2, x_1) \ \land \ X_8\neg E(x_2, x_2)
\]
Automatic Reduction Finding

FO Interpretations (Queries): \( \theta = (k, \varphi_0, \psi_1, \ldots, \psi_r) \)

- \( k \) is the dimension
- \( \varphi_0(x_1, \ldots, x_k) \) defines the new universe
- \( \psi_i(x_1, \ldots, x_{kr_i}) \) define the new relations

Complexity classes under interpretations (Immerman)

- quantifier-free interpretations are weaker than \( \text{PTime} \)
- still \( \text{P} = \text{NP} \) iff \( \text{SAT} \leq \text{qfpCVP} \)
- and \( \text{NL} = \text{NP} \) iff \( \text{SAT} \leq \text{qfpREACH} \)
- and \( \text{coNL} = \text{NL} \) (true) iff \( \neg \text{REACH} \leq \text{qfpREACH} \)

Finding reductions QBF:

- \( \exists \theta \forall A (A/\text{uni}22A7\varphi \Leftrightarrow \theta(A/\text{uni}22A7\varphi)) \)

CEGAR:

- Find a \( k \)-DNF reduction \( \theta \) good on counter-examples \( E/\text{zero.fitted}, \ldots, E_i/\text{one.fitted} \)
- Find a counter-example \( E_i/\text{one.fitted} \) to \( \theta_i \), iterate
Automatic Reduction Finding

FO Interpretations (Queries): \( \theta = (k, \varphi_0, \psi_1, \ldots, \psi_r) \)

- \( k \) is the dimension
- \( \varphi_0(x_1, \ldots, x_k) \) defines the new universe
- \( \psi_i(x_1, \ldots, x_{kr_i}) \) define the new relations

Complexity classes under interpretations (Immerman)

- quantifier-free interpretations are weaker than PTime
- still \( P=NP \) iff \( \text{SAT} \leq_{qfp} \text{CVP} \)
- and \( \text{NL}=NP \) iff \( \text{SAT} \leq_{qfp} \text{REACH} \),
- and \( \text{coNL}=\text{NL} \) (true) iff \( \neg \text{REACH} \leq_{qfp} \text{REACH} \)
Automatic Reduction Finding

FO Interpretations (Queries): $\theta = (k, \varphi_0, \psi_1, \ldots, \psi_r)$
- $k$ is the dimension
- $\varphi_0(x_1, \ldots, x_k)$ defines the new universe
- $\psi_i(x_1, \ldots, x_{kr_i})$ define the new relations

Complexity classes under interpretations (Immerman)
- quantifier-free interpretations are weaker than PTime
- still $P=NP$ iff $\text{SAT} \leq_{qfp} \text{CVP}$
- and $\text{NL}=\text{NP}$ iff $\text{SAT} \leq_{qfp} \text{REACH}$,
- and $\text{coNL}=\text{NL}$ (true) iff $\neg \text{REACH} \leq_{qfp} \text{REACH}$

Finding reductions  QBF: $\exists \theta \forall \mathcal{A} (\mathcal{A} \models \varphi_P \iff \theta(\mathcal{A}) \models \varphi_Q)$ CEGAR:
- Find a /-DNF reduction $\theta_i$ good on counter-examples $\mathcal{E}_0, \ldots, \mathcal{E}_i$
- Find a counter-example $\mathcal{E}_{i+1}$ to $\theta_i$, iterate
Reduction Finding Results

**Easy example:** $s$-$t$ reachability to strongly connected (both NL-complete)

\[
\text{Reach} = \left[ t_{c_{x,y}} E(x, y) \right] (s, t) \quad \text{SC} := \forall x, y (t_{c_{x,y}} E(x, y))
\]
Reduction Finding Results

**Easy example:** s-t reachability to strongly connected (both NL-complete)

\[
\text{Reach} = [tc_{x,y} E(x, y)](s, t) \quad \text{SC} := \forall x, y(tc_{x,y} E(x, y))
\]

http://toss.sf.net/reduct.html
Reduction Finding Results

**Easy example:** s-t reachability to strongly connected (both NL-complete)

\[
\text{Reach} = [t_{c_{x,y}} \ E(x,y)](s, t) \quad \text{SC} := \forall x, y(t_{c_{x,y}} \ E(x,y))
\]

\[
(k = 1, \ \varphi_0 = T, \ \psi_1 = x_1 = s \lor x_2 = t \lor E(x_2, x_1))
\]
Reduction Finding Results

**Easy example:** s-t reachability to strongly connected (both NL-complete)

\[
\text{Reach} = \left[ t_{c_{x,y}} \ E(x, y) \right](s, t) \quad \text{SC} := \forall x, y \left( t_{c_{x,y}} \ E(x, y) \right)
\]

\[
(k = 1, \phi_0 = \top, \psi_1 = x_1 = .s \lor x_2 = .t \lor E(x_2, x_1))
\]

**# Unsolved Cases** (out of 2304)

<table>
<thead>
<tr>
<th>(c, n)</th>
<th>(1, 3)</th>
<th>(2, 3)</th>
<th>(3, 3)</th>
<th>(1, 4)</th>
<th>(2, 4)</th>
<th>(3, 4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>rareqs</td>
<td>0</td>
<td>0</td>
<td>16</td>
<td>19</td>
<td>65</td>
<td>204</td>
</tr>
<tr>
<td>depqbf</td>
<td>0</td>
<td>142</td>
<td>547</td>
<td>16</td>
<td>297</td>
<td>711</td>
</tr>
<tr>
<td>qube</td>
<td>10</td>
<td>536</td>
<td>949</td>
<td>82</td>
<td>760</td>
<td>1082</td>
</tr>
<tr>
<td>cirqit</td>
<td>58</td>
<td>673</td>
<td>1138</td>
<td>511</td>
<td>1092</td>
<td>1357</td>
</tr>
<tr>
<td>skizzo</td>
<td>522</td>
<td>1058</td>
<td>1156</td>
<td>975</td>
<td>1327</td>
<td>1434</td>
</tr>
</tbody>
</table>
Outlook

Experiences with Reduction Finding

- 2QBF in principle, 3QBF after qdimacs conversion
- Better to use 3QBF, qpro helps when using cirqit
- Best performing QBF solver: rareqs
- Missing part against hand-made CEGAR: incremental solving
- Generator at http://toss.sf.net/reductGen.html

Other Applications

- Finding LFP formulas for NP ∩ coNP
- Solving n-step winning in board games
- The same generator plugged into a GGP system
- Easy to generate QBFs for many games and board sizes
- Best performing QBF solver (only a few tests): depqbf
- http://toss.sf.net/gameGen.html

QBF is great for many applications (but needs effort)

Thank You
## Outlook

### Experiences with Reduction Finding
- 2QBF in principle, 3QBF after qdimacs conversion
- Better to use 3QBF, qpro helps when using cirqit
- Best performing QBF solver: rareqs
- Missing part against hand-made CEGAR: incremental solving
- Generator at [http://toss.sf.net/reductGen.html](http://toss.sf.net/reductGen.html)

### Other Applications
- Finding LFP formulas for \( \text{NP} \cap \text{coNP} \) properties
- Solving \( n \)-step winning in board games
  - the same generator plugged into a GGP system
  - easy to generate QBFs for many games and board sizes
  - best performing QBF solver (only a few tests): depqbf
    - [http://toss.sf.net/gameGen.html](http://toss.sf.net/gameGen.html)
- QBF is great for many applications (but needs effort)
Outlook

Experiences with Reduction Finding

• 2QBF in principle, 3QBF after qdimacs conversion
• Better to use 3QBF, qpro helps when using cirqit
• Best performing QBF solver: rareqs
• Missing part against hand-made CEGAR: incremental solving
• Generator at http://toss.sf.net/reductGen.html

Other Applications

• Finding LFP formulas for $NP \cap coNP$ properties
• Solving $n$-step winning in board games
  • the same generator plugged into a GGP system
  • easy to generate QBFs for many games and board sizes
  • best performing QBF solver (only a few tests): depqbf
  • http://toss.sf.net/gameGen.html
• QBF is great for many applications (but needs effort)

Thank You