Polynomial Guarded Transformation for the Modal Mu-Calculus Is Still Open

Florian Bruse, Oliver Friedmann and Martin Lange

September 19, 2013
Guarded Normal Form

Setting: modal $\mu$-calculus and extensions

$\text{GNF}: \text{Every fixpoint variable under scope of a modal operator}$

Example: $\mu X.\Diamond (P \lor X)$ guarded, $\mu X.\nu Y.X \lor \Diamond Y$ not guarded

Why GNF?
Synchronizes unfolding of fixpoints in tableaux, helps in constructions, translations to automata, etc.

can effectively transform any formula into guarded equivalent (BB89, Wal00, KVW00, Mat02)

Hence GNF commonly assumed when working with $\mu$-calculus
Guarded Normal Form

Setting: modal $\mu$-calculus and extensions

**GNF:** Every fixpoint variable under scope of a modal operator

Example: $\mu X. \Diamond (P \lor X)$ guarded, $\mu X. \nu Y. X \lor \Diamond Y$ not guarded

Why GNF?
Synchronizes unfolding of fixpoints in tableaux, helps in constructions, translations to automata, etc.

can effectively transform any formula into guarded equivalent (BB89, Wal00,KVW00,Mat02)

Hence GNF commonly assumed when working with $\mu$-calculus
Failure of Previous Results

Theorem: guarded transformation possible with no blowup (KVW00)/quadratic blowup (Mat02)
Failure of Previous Results

Theorem: guarded transformation possible with no blowup (KVW00)/quadratic blowup (Mat02)

Counterexample is

\[ \Phi_n = \mu X_n \cdots \mu X_1. (X_n \lor \cdots \lor X_1 \lor \Box (X_n \lor \cdots \lor X_1)) \]

Known GT procedures produce formulae of exponential modal depth

Reason: occurrence of variable at modal depth \( d \) will produce formula at modal depth \( 2d \) after unfolding
Vectorial Form and Hierarchical Equation Systems

Vectorial Form: allow formulae of form

\[
\sigma \left\{ \begin{array}{c}
X_1 \cdot \varphi_1 \\
\vdots \\
X_n \cdot \varphi_n
\end{array} \right\}
\]

where each \( \varphi_i \) may refer to all other \( X_i \).
Vectorial Form and Hierarchical Equation Systems

Vectorial Form: allow formulae of form

$$\sigma \left\{ \begin{array}{c} X_1 \cdot \varphi_1 \\ \vdots \\ X_n \cdot \varphi_n \end{array} \right\}$$

where each $\varphi_i$ may refer to all other $X_i$.

HES: Allow every fixpoint subformula to refer to every variable

Don’t gain expressive power, but succinctness (best known algorithm to unfold is at least exponential)

Notion of guardedness can be generalized
State of the Art

<table>
<thead>
<tr>
<th>$\mu$-calculus</th>
<th>possibly unguarded</th>
<th>guarded</th>
</tr>
</thead>
<tbody>
<tr>
<td>equation system</td>
<td></td>
<td></td>
</tr>
<tr>
<td>with vectorial form</td>
<td></td>
<td></td>
</tr>
<tr>
<td>without vectorial form</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- polynomial (NS99)
- $\leq$ exponential
State of the Art

Our results in blue

<table>
<thead>
<tr>
<th>$\mu$-calculus</th>
<th>possibly unguarded</th>
<th>guarded</th>
</tr>
</thead>
<tbody>
<tr>
<td>equation system</td>
<td>$\geq$ parity</td>
<td></td>
</tr>
<tr>
<td>with vectorial form</td>
<td>polynomial (NS99)</td>
<td>$\geq$ parity</td>
</tr>
<tr>
<td>without vectorial form</td>
<td>open</td>
<td>$\geq$ parity</td>
</tr>
<tr>
<td></td>
<td>$\leq$ exponential</td>
<td></td>
</tr>
</tbody>
</table>
Poly GT for Vect. Formulae → Poly PG Solving

1. Given $\mu$-calc. formula $\varphi$ and TS $T, s$, can obtain vectorial $\varphi'$ and $T'$ s.t. $\varphi'$ $\Diamond, \Box$-free, $T'$ has only one state, both polynomial size, and

$$T, s \models \varphi \iff T' \models \varphi'$$
Poly GT for Vect. Formulae → Poly PG Solving

1. Given $\mu$-calc. formula $\varphi$ and TS $T, s$, can obtain vectorial $\varphi'$ and $T'$ s.t. $\varphi'$ $\Diamond, \square$-free, $T'$ has only one state, both polynomial size, and

$$T, s \models \varphi \iff T' \models \varphi'$$

2. GT for $\Diamond, \square$-free formula must eliminate all fixpoints
Poly GT for Vect. Formulae → Poly PG Solving

1. Given $\mu$-calc. formula $\varphi$ and TS $\mathcal{T}, s$, can obtain vectorial $\varphi'$ and $\mathcal{T}'$ s.t. $\varphi'$ $\diamondsuit,\Box$-free, $\mathcal{T}'$ has only one state, both polynomial size, and

$$\mathcal{T}, s \models \varphi \iff \mathcal{T}' \models \varphi'$$

2. GT for $\diamondsuit,\Box$-free formula must eliminate all fixpoints

3. model-check resulting formula in linear time
Poly GT for Vect. Formulae $\rightarrow$ Poly PG Solving

1. Given $\mu$-calc. formula $\varphi$ and TS $T, s$, can obtain vectorial $\varphi'$ and $T'$ s.t. $\varphi'$ $\Diamond, \Box$-free, $T'$ has only one state, both polynomial size, and

$$T, s \models \varphi \iff T' \models \varphi'$$

2. GT for $\Diamond, \Box$-free formula must eliminate all fixpoints

3. model-check resulting formula in linear time

4. Consider PG, relevant Walukiewicz-formula: apply steps 1-3

Result: Polynomial GT for vectorial $\mu$-calculus gives rise to polynomial solution procedure for parity games

Also holds for HES
Consequences and Outlook

Consequences

- Results inKVW00 still hold
- Some results in other papers only valid for guarded formulae, see our paper (BFL13)
Consequences and Outlook

Consequences

- Results in KVW00 still hold
- Some results in other papers only valid for guarded formulae, see our paper (BFL13)

Open Questions:

- polynomial GT for $\mu$-calculus $\rightarrow$ polynomial parity game solving?
- polynomial parity game solving $\rightarrow$ polynomial GT?
- relation between $\epsilon$-transitions in alternating automata and GNF
The End

Thanks!
The End

Thanks!

Literature:


