Interpolation for guarded logics

Michael Vanden Boom

University of Oxford

Highlights 2014
Paris, France

Joint work with
Michael Benedikt and Balder ten Cate
Some guarded logics

constrain quantification

\[ \exists x (G(xy) \land \psi(xy)) \]
\[ \forall x (G(xy) \rightarrow \psi(xy)) \]

[Andréka, van Benthem, Németi '95-'98]
Some guarded logics

- Constrain quantification:
  \[ \exists x (G(xy) \land \psi(xy)) \]
  \[ \forall x (G(xy) \rightarrow \psi(xy)) \]

  [Andréka, van Benthem, Németi ’95–’98]

- Constrain negation:
  \[ \exists x (\psi(xy)) \]
  \[ \neg \psi(x) \]

  [ten Cate, Segoufin ’11]
Some guarded logics

constrain quantification

$$\exists x(G(xy) \land \psi(xy))$$
$$\forall x(G(xy) \rightarrow \psi(xy))$$

[Andréka, van Benthem, Németi ’95–’98]

constrain negation

$$\exists x(\neg \psi(xy))$$
$$G(xy) \land \neg \psi(xy)$$

[ten Cate, Segoufin ’11]
[Barány, ten Cate, Segoufin ’11]
Some guarded logics

These guarded logics are **decidable**, and **expressive** enough to capture many query languages and integrity constraints of interest in databases and knowledge representation.
\( \varphi \models \psi \)
Interpolation

\[ \varphi \models \chi \models \psi \]

only uses relations in both \( \varphi \) and \( \psi \)
Interpolation example

\[ \exists xyz(T_{xyz} \land R_{xy} \land R_{yz} \land R_{zx}) \models \exists xy(R_{xy} \land ((S_x \land S_y) \lor (\neg S_x \land \neg S_y))) \]

“there is a \textit{T}-guarded 3-cycle using \textit{R}”
Interpolation example

\[ \exists xyz(T_{xyz} \land R_{xy} \land R_{yz} \land R_{zx}) \models \exists xy(R_{xy} \land ((S_x \land S_y) \lor (\neg S_x \land \neg S_y))) \]

"there is a $T$-guarded 3-cycle using $R$"
Interpolation example

$$\exists xyz (T_{xyz} \land R_{xy} \land R_{yz} \land R_{zx}) \models \exists xy (R_{xy} \land ((S_x \land S_y) \lor (S_x \land \neg S_y)))$$

“there is a $T$-guarded 3-cycle using $R$”
Interpolation example

\[ \exists xyz (T_{xyz} \land R_{xy} \land R_{yz} \land R_{zx}) \models \exists y (R_{xy} \land ((S_x \land S_y) \lor (\neg S_x \land \neg S_y))) \]

“there is a \( T \)-guarded 3-cycle using \( R \)”

\[ \text{interpolant } \chi := \exists xyz (R_{xy} \land R_{yz} \land R_{zx}) \]

“there is a 3-cycle using \( R \)”
Why do we care?
Why do we care?

Why might someone care?
Why do we care?

Why might someone care?

- interpolation is a benchmark property of modal logic
Why do we care?

Why might someone care?

- Interpolation is a benchmark property of modal logic.
- Interpolation implies the Beth definability property (implicit definability = explicit definability) which indicates a good balance between syntax and semantics.
Why do we care?

Why might someone care?

- Interpolation is a **benchmark** property of modal logic

- Interpolation implies the **Beth definability** property (implicit definability = explicit definability) which indicates a good balance between syntax and semantics

- For these guarded logics with connections to databases, interpolation is related to **query rewriting** over views
**Theorem** (ten Cate, Segoufin ’11; Barany, Benedikt, ten Cate ’13)

Given GNF (respectively, UNF) formulas $\varphi$ and $\psi$ such that $\varphi \models \psi$, there is a GNF (respectively, UNF) interpolant $\chi$. 

\[ \varphi \models \chi \models \psi \]

only uses relations in both $\varphi$ and $\psi$
Interpolation

\[ \varphi \models \chi \models \psi \]

only uses relations in both \( \varphi \) and \( \psi \)

**Theorem** (ten Cate, Segoufin ’11; Barany, Benedikt, ten Cate ’13)
Given GNF (respectively, UNF) formulas \( \varphi \) and \( \psi \) such that \( \varphi \models \psi \), there is a GNF (respectively, UNF) interpolant \( \chi \).

No idea how to **compute** interpolants (or other rewrites related to interpolation).
Interpolation

\[
\varphi \vdash \chi \vdash \psi
\]

only uses relations in both \(\varphi\) and \(\psi\)

**Theorem** (ten Cate, Segoufin ’11; Barany, Benedikt, ten Cate ’13)

Given GNF (respectively, UNF) formulas \(\varphi\) and \(\psi\) such that \(\varphi \vdash \psi\), there is a GNF (respectively, UNF) interpolant \(\chi\).

**Theorem** (Benedikt, ten Cate, VB. ’14)

Given GNF (respectively, UNF) formulas \(\varphi\) and \(\psi\) s.t. \(\varphi \vdash \psi\), we can construct a GNF (respectively, UNF) interpolant \(\chi\) of doubly exponential DAG-size.
Conclusion

<table>
<thead>
<tr>
<th>Interpolation?</th>
<th>ML</th>
<th>GF</th>
<th>UNF</th>
<th>GNF</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

adapted mosaic method from ML

[Benedikt, ten Cate, VB.'14]
## Conclusion

<table>
<thead>
<tr>
<th>Interpolation?</th>
<th>ML</th>
<th>GF</th>
<th>UNF</th>
<th>GNF</th>
<th>$L_\mu$</th>
<th>GFP</th>
<th>UNFP</th>
<th>GNFP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
</tr>
</tbody>
</table>

adapted
mosaic method
from ML

[Benedikt, ten Cate, VB.'14]
## Conclusion

<table>
<thead>
<tr>
<th>Interpolation?</th>
<th>ML</th>
<th>GF</th>
<th>UNF</th>
<th>GNF</th>
<th>$L_\mu$</th>
<th>GFP</th>
<th>UNFP</th>
<th>GNFP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>✓</td>
<td>X</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>X</td>
<td>✓</td>
</tr>
</tbody>
</table>

adapted mosaic method from ML

[Benedikt, ten Cate, VB. ’14]

used automata for $L_\mu$

[Benedikt, ten Cate, VB. unpublished]
## Conclusion

<table>
<thead>
<tr>
<th></th>
<th>ML</th>
<th>GF</th>
<th>UNF</th>
<th>GNF</th>
<th>$L_\mu$</th>
<th>GFP</th>
<th>UNFP</th>
<th>GNFP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interpolation?</td>
<td>✔️</td>
<td>✖️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✖️</td>
<td>✔️</td>
<td>✖️</td>
</tr>
</tbody>
</table>

- Adapted mosaic method from ML
  [Benedikt, ten Cate, VB. ’14]

- Used automata for $L_\mu$
  [Benedikt, ten Cate, VB. unpublished]

## Open question

Is there a decidable extension of GNFP that has interpolation?