Interpolation with decidable fixpoint logics

Michael Benedikt\textsuperscript{1}, Balder ten Cate\textsuperscript{2}, Michael Vanden Boom\textsuperscript{1}

\textsuperscript{1}University of Oxford \textsuperscript{2}LogicBlox and UC Santa Cruz

LICS 2015
Kyoto, Japan
Some decidable fragments of first-order logic

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

[Andréka, van Benthem, Németi ’95-’98]
Some decidable fragments of first-order logic

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

[Andréka, van Benthem, Németi '95-'98]

\[ \exists x (\psi(xy)) \]
\[ \neg \psi(x) \]

[ten Cate, Segoufin '11]
Some decidable fragments of first-order logic

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

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

constrain negation
\[ \exists x (\psi(xy)) \]
\[ G(xy) \land \neg \psi(xy) \]

[ten Cate, Segoufin ‘11]
[Barány, ten Cate, Segoufin ‘11]
Some decidable fragments of FO+LFP

- Constrain quantification
  \[ \exists x(G(xy) \land \psi(xy)) \]
  \[ \forall x(G(xy) \rightarrow \psi(xy)) \]
  [Andréka, van Bentham, Németi ‘95-'98]

- Constrain negation
  \[ \exists x(\psi(xy)) \]
  \[ G(xy) \land \neg \psi(xy) \]
  [ten Cate, Segoufin ‘11]
  [Bárány, ten Cate, Segoufin ‘11]
Why study guarded logics?

Guarded logics are expressive. For instance, UNFP captures:

- mu-calculus, even with backwards modalities;
- positive existential FO (i.e. unions of conjunctive queries);
- description logics including $\mathcal{ALC}, \mathcal{ALCHIO}, \mathcal{ELI}$;
- monadic Datalog.
Why study guarded logics?

Guarded logics are **expressive**. For instance, UNFP captures:

- mu-calculus, even with backwards modalities;
- positive existential FO (i.e. unions of conjunctive queries);
- description logics including $\mathcal{ALC}, \mathcal{ALCHIO}, \mathcal{ELI}$;
- monadic Datalog.

Guarded logics have many nice model theoretic properties.

- GF, UNF, and GNF have **finite models**.
- GFP, UNFP, and GNFP have **tree-like models** (models of bounded tree-width).
Why study guarded logics?

Guarded logics are **expressive**. For instance, UNFP captures:

- mu-calculus, even with backwards modalities;
- positive existential FO (i.e. unions of conjunctive queries);
- description logics including \( \text{ALC}, \text{ALCHIO}, \text{ELI} \);
- monadic Datalog.

Guarded logics have many nice model theoretic properties.

- GF, UNF, and GNF have **finite models**.
- GFP, UNFP, and GNFP have **tree-like models**
  (models of bounded tree-width).

Some guarded logics have **interpolation**...
\[ \phi \models \psi \]
Interpolation

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

\[ \uparrow \text{interpolant} \]

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 \( T \)-guarded
3-cycle using \( R \)”
**Interpolation example**

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

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

\[ \exists xyz (Txyz \land Rxy \land Ryz \land Rzx) \vdash \exists xy (Rxy \land ((Sx \land Sy) \lor (\neg Sx \land \neg Sy))) \]

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

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

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

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

“there is a 3-cycle using \( R \)”
Why study interpolation?

- Interpolation is a **benchmark** property of ML and $L_\mu$. 
Why study interpolation?

- Interpolation is a **benchmark** property of ML and L_μ.

- Interpolation implies several results about going from **semantic properties to syntactic properties** (e.g., Beth definability, preservation theorems, etc.)
Why study interpolation?

- Interpolation is a **benchmark** property of ML and $L_\mu$.

- Interpolation implies several results about going from **semantic properties to syntactic properties** (e.g., Beth definability, preservation theorems, etc.)

- Interpolation is related to **query rewriting** over views.

- Interpolation is related to **modularity** in description logics.
**Interpolation results**

Very little is known about interpolation for fixpoint logics over general relational structures, where relations can have arbitrary arity.

<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>Craig interpolation</td>
<td>✔️</td>
<td>X</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>?</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>


Very little is known about interpolation for fixpoint logics over general relational structures, where relations can have arbitrary arity.

<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>Craig interpolation</td>
<td>✔️</td>
<td>✗</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✗</td>
<td>✔️</td>
<td>✗</td>
</tr>
</tbody>
</table>
Interpolation results

Very little is known about interpolation for fixpoint logics over general relational structures, where relations can have arbitrary arity.

<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>Craig interpolation</td>
<td>✔</td>
<td>✗</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✗</td>
<td>✔</td>
<td>✗</td>
</tr>
</tbody>
</table>

**Contribution:** bootstrapping from ML / $L_{\mu}$ extended to interpolation
Uniform interpolation

**Theorem** (D’Agostino, Hollenberg ‘00)

$L_\mu$ has effective uniform interpolation.
Uniform interpolation

Theorem (D’Agostino, Hollenberg ‘00)

$L_\mu$ has effective uniform interpolation.

A uniform interpolant $\chi$ depends only on the antecedent $\varphi$ and the signature of the consequent (rather than a particular consequent $\psi$).

*Given $\varphi$ and a sub-signature $\sigma$, there is a formula $\chi$ over $\sigma$ such that for all $\psi$ with $\varphi \models \psi$ and common signature $\sigma$, $\varphi \models \chi \models \psi$.***
**Theorem** (D’Agostino, Hollenberg ’00)

$L_\mu$ has effective uniform interpolation.

A **uniform interpolant** $\chi$ depends only on the antecedent $\varphi$ and the signature of the consequent (rather than a particular consequent $\psi$).

*Given $\varphi$ and a sub-signature $\sigma$, there is a formula $\chi$ over $\sigma$ such that for all $\psi$ with $\varphi \models \psi$ and common signature $\sigma$, $\varphi \models \chi \models \psi$.*

Let UNFP$^k$ denote the $k$-variable fragment of UNFP (in normal form...).

**Theorem** (Benedikt, ten Cate, VB. ’15)

UNFP$^k$ has effective uniform interpolation.

UNFP has effective Craig interpolation.
Uniform interpolation example

“$S$ holds at $x$, and from every position $y$ where $S$ holds, there is an $R$-neighbor $z$ where $S$ holds”

$$\varphi(x) := Sx \land \forall y( Sy \rightarrow \exists z( Ryz \land Sz))$$
$$\equiv Sx \land \neg \exists y( Sy \land \neg \exists z( Ryz \land Sz))$$
Uniform interpolation example

“$S$ holds at $x$, and from every position $y$ where $S$ holds, there is an $R$-neighbor $z$ where $S$ holds”

$$\varphi(x) := Sx \land \forall y(Sy \rightarrow \exists z(Ryz \land Sz))$$
$$\equiv Sx \land \lnot \exists y(Sy \land \lnot \exists z(Ryz \land Sz))$$

Uniform interpolant of $\varphi$ over subsignature $\{R\}$

“there is an infinite $R$-path from $x$”

$$[\text{gfp}_{Y,y} . \exists z(Ryz \land Yz)](x)$$
$$\equiv \lnot [\text{lfp}_{Y,y} . \lnot \exists z(Ryz \land \lnot Yz)](x)$$
Uniform interpolation for UNFP\(^k\)

**Theorem** (Benedikt, ten Cate, VB. ’15)

UNFP\(^k\) has effective uniform interpolation.

**Proof strategy:** Exploit tree-like model property and results from modal world.

([Grädel, Walukiewicz ’99], [Grädel, Hirsch, Otto ’00], [D’Agostino, Hollenberg ’00])
Uniform interpolation for UNFP^k

**Theorem** (Benedikt, ten Cate, VB. '15)

UNFP^k has effective uniform interpolation.

**Proof strategy:** Exploit tree-like model property and results from modal world.
Uniform interpolation for UNFP$^k$

**Theorem** (Benedikt, ten Cate, VB. '15)

UNFP$^k$ has effective uniform interpolation.

**Proof strategy:** Exploit tree-like model property and results from modal world.

\[
\text{Relational structures} \quad \xrightarrow{\text{Coded structures}} \quad \text{(tree decompositions of width } k) \\
\text{UNFP}^k \varphi \quad \rightarrow \quad \mathcal{L}_\mu \hat{\varphi}
\]
Uniform interpolation for $\text{UNFP}^k$

**Theorem** (Benedikt, ten Cate, VB. ’15)

$\text{UNFP}^k$ has effective uniform interpolation.

**Proof strategy:** Exploit **tree-like model property** and results from modal world.

- **Relational structures**
- **Coded structures**
  - (tree decompositions of width $k$)

\[
\begin{align*}
\text{UNFP}^k \varphi & \rightarrow L_\mu \hat{\varphi} \\
& \downarrow \text{[D'Agostino, Hollenberg'00]} \\
& L_\mu \hat{\chi} \\
& \text{over subsignature encoding}
\end{align*}
\]
Uniform interpolation for UNFP$^k$

**Theorem** (Benedikt, ten Cate, VB. ’15)

UNFP$^k$ has effective uniform interpolation.

**Proof strategy:** Exploit tree-like model property and results from modal world.

\[
\begin{align*}
\text{UNFP}^k \varphi & \rightarrow L_\mu \hat{\varphi} \\
\text{UNFP}^k \chi & \leftarrow L_\mu \hat{\chi}
\end{align*}
\]

Relational structures

Coded structures (tree decompositions of width $k$)

[D’Agostino, Hollenberg’00]
Conclusion

UNFP is an expressive, decidable fixpoint logic with effective interpolation.

<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>Craig interpolation</td>
<td>✔️</td>
<td>✗</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✗</td>
<td>✔️</td>
<td>✗</td>
</tr>
</tbody>
</table>
Fix a set $K = \{1, \ldots, k\}$ of names for elements. Let $\mathcal{K}_{\sigma,k} := \{\mathcal{C} : \mathcal{C} \text{ is a $\sigma$-structure with universe } C \subseteq K \text{ of size at most } k\}$. 

A $\mathcal{K}_{\sigma,k}$-tree is an unranked infinite tree with

- arbitrary branching (possibly infinite),
- node labels $\mathcal{C} \in \mathcal{K}_{\sigma,k}$,
- edge labels are partial functions $f : K \to K$ describing relationship between names.
Fix a set $K = \{1, \ldots, k\}$ of names for elements.

Let $\mathcal{K}_{\sigma,k} := \{\mathcal{C} : \mathcal{C} \text{ is a } \sigma\text{-structure with universe } C \subseteq K \text{ of size at most } k\}$.

A $\mathcal{K}_{\sigma,k}$-tree is an unranked infinite tree with

- arbitrary branching (possibly infinite),
- node labels $\mathcal{C} \in \mathcal{K}_{\sigma,k}$,
- edge labels are partial functions $f : K \to K$ describing relationship between names.
Fix a set \( K = \{1, \ldots, k\} \) of names for elements.

Let \( K_{\sigma,k} := \{\mathcal{C} : \mathcal{C} \text{ is a } \sigma\text{-structure with universe } C \subseteq K \text{ of size at most } k\} \).

A \( K_{\sigma,k}\)-tree is an unranked infinite tree with

- arbitrary branching (possibly infinite),
- node labels \( \mathcal{C} \in K_{\sigma,k} \),
- edge labels are partial functions \( f : K \to K \) describing relationship between names.

\( K_{\sigma,k}\)-trees are consistent if neighboring nodes agree on any shared names.