Fixpoint logics with tree-like models

Michael Vanden Boom

University of Oxford

Oxford Information Systems Seminar
November 2015

Including joint work with
Michael Benedikt, Pierre Bourhis, Balder ten Cate, and Thomas Colcombet

Fixpoint logics give mechanism to express
dynamic, recursive properties.

Example

binary relation R, unary relation P

"from x, it is possible to R-reach some P-element"

$$[\texttt{Reach-}P](x) := [\mathbf{lfp}_{Y,y} \,.\, \exists z(Ryz \wedge (Pz \vee Yz))](x)$$

Least fixpoint

Consider $\psi(\boldsymbol{y}, Y)$ positive in Y (of arity $m = |\boldsymbol{y}|$).

For all structures $\mathfrak{A}$, the formula ψ induces a monotone operation

$$\begin{aligned} \mathcal{P}(A^m) &\longrightarrow \mathcal{P}(A^m) \\ V &\longmapsto \psi_{\mathfrak{A}}(V) := \{\boldsymbol{a} \in A^m : \mathfrak{A}, \boldsymbol{a}, V \vDash \psi\} \end{aligned}$$

$\Rightarrow$ there is a unique least fixpoint $[\mathbf{lfp}_{Y,\boldsymbol{y}}.\psi(\boldsymbol{y}, Y)]_{\mathfrak{A}} := \bigcup_\alpha \psi^\alpha_{\mathfrak{A}}$

$$\begin{aligned} \psi^0_{\mathfrak{A}} &:= \varnothing \\ \psi^{\alpha+1}_{\mathfrak{A}} &:= \psi_{\mathfrak{A}}(\psi^\alpha_{\mathfrak{A}}) \\ \psi^\lambda_{\mathfrak{A}} &:= \bigcup_{\alpha<\lambda} \psi^\alpha_{\mathfrak{A}} \end{aligned}$$

Least fixpoint

Consider $\psi(\boldsymbol{y}, Y)$ positive in Y (of arity $m = |\boldsymbol{y}|$).

For all structures $\mathfrak{A}$, the formula ψ induces a monotone operation

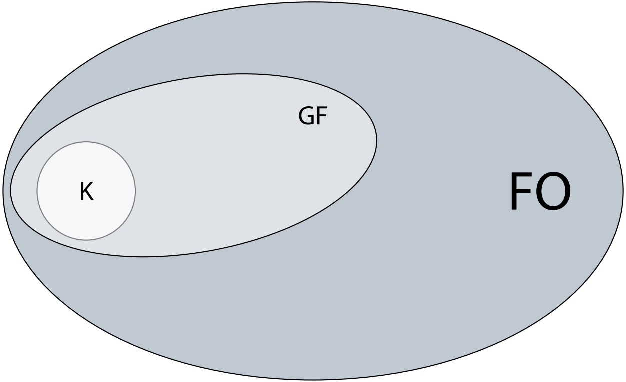
$$\begin{aligned} \mathcal{P}(A^m) &\longrightarrow \mathcal{P}(A^m) \\ V &\longmapsto \psi_{\mathfrak{A}}(V) := \{\boldsymbol{a} \in A^m : \mathfrak{A}, \boldsymbol{a}, V \models \psi\} \end{aligned}$$

$\Rightarrow$ there is a unique least fixpoint $[\mathbf{lfp}_{Y,\boldsymbol{y}}.\psi(\boldsymbol{y}, Y)]_{\mathfrak{A}} := \bigcup_\alpha \psi_{\mathfrak{A}}^\alpha$

$$\begin{aligned} \psi_{\mathfrak{A}}^0 &:= \varnothing \\ \psi_{\mathfrak{A}}^{\alpha+1} &:= \psi_{\mathfrak{A}}(\psi_{\mathfrak{A}}^\alpha) \\ \psi_{\mathfrak{A}}^\lambda &:= \bigcup_{\alpha<\lambda} \psi_{\mathfrak{A}}^\alpha \end{aligned}$$

Semantics of fixpoint operator: $\mathfrak{A}, \boldsymbol{a} \models [\mathbf{lfp}_{Y,\boldsymbol{y}}.\psi(\boldsymbol{y}, Y)](\boldsymbol{x})$ iff $\boldsymbol{a} \in \bigcup_\alpha \psi_{\mathfrak{A}}^\alpha$

Some decidable fragments of first-order logic

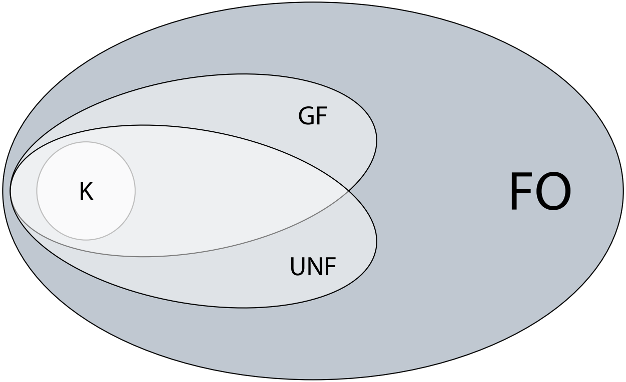


constrain
quantification

$\exists \boldsymbol{x}(G(\boldsymbol{xy}) \wedge \psi(\boldsymbol{xy}))$
$\forall \boldsymbol{x}(G(\boldsymbol{xy}) \rightarrow \psi(\boldsymbol{xy}))$

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

Some decidable fragments of first-order logic



constrain quantification

$\exists \boldsymbol{x}(G(\boldsymbol{xy}) \wedge \psi(\boldsymbol{xy}))$
$\forall \boldsymbol{x}(G(\boldsymbol{xy}) \rightarrow \psi(\boldsymbol{xy}))$

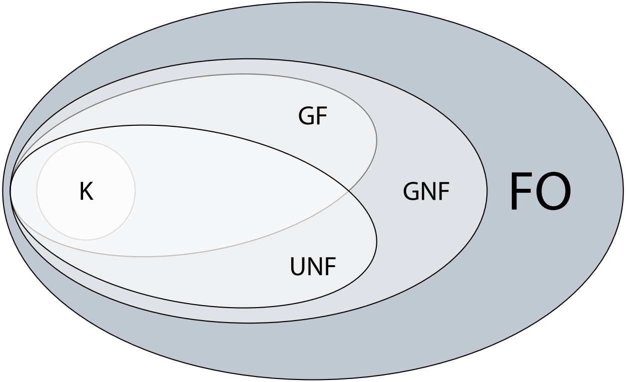
[Andréka, van Benthem, Németi '95-'98]

constrain negation

$\exists \boldsymbol{x}(\psi(\boldsymbol{xy}))$
$\neg\psi(x)$

[ten Cate, Segoufin '11]

Some decidable fragments of first-order logic



constrain quantification

$$\exists \boldsymbol{x}(G(\boldsymbol{xy}) \wedge \psi(\boldsymbol{xy}))$$
$$\forall \boldsymbol{x}(G(\boldsymbol{xy}) \rightarrow \psi(\boldsymbol{xy}))$$

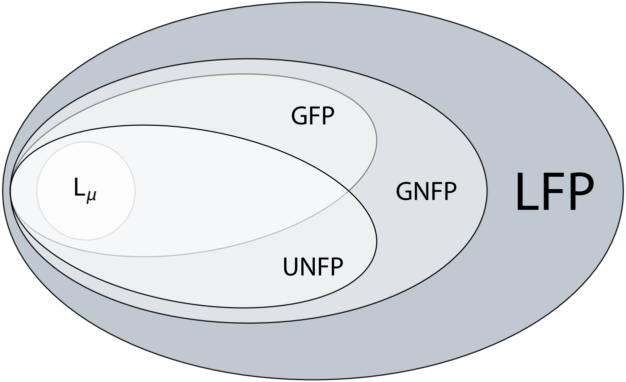
[Andréka, van Benthem, Németi ’95-’98]

constrain negation

$$\exists \boldsymbol{x}(\psi(\boldsymbol{xy}))$$
$$G(\boldsymbol{xy}) \wedge \neg\psi(\boldsymbol{xy})$$

[ten Cate, Segoufin ’11]
[Bárány, ten Cate, Segoufin ’11]

Some decidable fragments of LFP (fixpoint extension of FO)



constrain quantification

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

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

constrain negation

$\exists \boldsymbol{x}(\psi(\boldsymbol{xy}))$
$G(\boldsymbol{xy}) \land \neg\psi(\boldsymbol{xy})$

[ten Cate, Segoufin '11]
[Bárány, ten Cate, Segoufin '11]

Guarded fixpoints: tuples in fixpoint are guarded by atom in original signature.
(UNFP has only monadic fixpoints, which are trivially guarded.)

Guarded negation fixpoint logic (GNFP)

Fix some signature σ of relations and constants.

Syntax of GNFP$[\sigma]$

$$\varphi \ ::= \ R\boldsymbol{t} \ \mid \ Y\boldsymbol{t} \ \mid \ \varphi \land \varphi \ \mid \ \varphi \lor \varphi \ \mid \ \exists y(\psi(\boldsymbol{x}y)) \ \mid \ G(\boldsymbol{x}) \land \neg\psi(\boldsymbol{x}) \ \mid$$
$$[\mathbf{lfp}_{Y,\boldsymbol{y}} \,.\, G(\boldsymbol{y}) \land \varphi(\boldsymbol{y}, Y, \boldsymbol{Z})](\boldsymbol{t}) \quad \text{where } Y \text{ only occurs positively in } \varphi$$

where R and G are relations in σ or $=$, and $\boldsymbol{t}$ is a tuple over variables and constants.

Guarded negation fixpoint logic (GNFP)

Fix some signature σ of relations and constants.

Syntax of GNFP$[\sigma]$

$$\varphi \ ::= \ R\boldsymbol{t} \ | \ Y\boldsymbol{t} \ | \ \varphi \wedge \varphi \ | \ \varphi \vee \varphi \ | \ \exists y(\psi(\boldsymbol{x}y)) \ | \ G(\boldsymbol{x}) \wedge \neg\psi(\boldsymbol{x}) \ |$$
$$[\mathbf{lfp}_{Y,\boldsymbol{y}} \,.\, G(\boldsymbol{y}) \wedge \varphi(\boldsymbol{y}, Y, \boldsymbol{Z})](\boldsymbol{t}) \quad \text{where } Y \text{ only occurs positively in } \varphi$$

where R and G are relations in σ or =, and $\boldsymbol{t}$ is a tuple over variables and constants.

Examples

- unions of conjunctive queries (in GNF)
- frontier-guarded tgds (in GNF):
  $\forall xyz\,((Rxy \wedge Ryz) \rightarrow \exists w(Twyz)) \equiv \neg\exists xyz\,(Rxy \wedge Ryz \wedge \neg\exists w(Twyz))$

Guarded negation fixpoint logic (GNFP)

Fix some signature σ of relations and constants.

Syntax of GNFP$[\sigma]$

$$\varphi ::= R\boldsymbol{t} \mid Y\boldsymbol{t} \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \exists y(\psi(\boldsymbol{x}y)) \mid G(\boldsymbol{x}) \wedge \neg\psi(\boldsymbol{x}) \mid$$
$$[\mathbf{lfp}_{Y,\boldsymbol{y}} \,.\, G(\boldsymbol{y}) \wedge \varphi(\boldsymbol{y}, Y, \boldsymbol{Z})](\boldsymbol{t}) \quad \text{where } Y \text{ only occurs positively in } \varphi$$

where R and G are relations in σ or =, and $\boldsymbol{t}$ is a tuple over variables and constants.

Examples

- unions of conjunctive queries (in GNF)
- frontier-guarded tgds (in GNF):
  $\forall xyz\,((Rxy \wedge Ryz) \rightarrow \exists w(Twyz)) \equiv \neg\exists xyz\,(Rxy \wedge Ryz \wedge \neg\exists w(Twyz))$
- description logics including $\mathcal{ALC}$, $\mathcal{ALCHIO}$, $\mathcal{ELI}$ (in GNF)

Guarded negation fixpoint logic (GNFP)

Fix some signature σ of relations and constants.

Syntax of GNFP$[\sigma]$

$$\varphi \ ::= \ R\boldsymbol{t} \mid Y\boldsymbol{t} \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \exists y(\psi(\boldsymbol{xy})) \mid G(\boldsymbol{x}) \wedge \neg\psi(\boldsymbol{x}) \mid$$
$$[\mathbf{lfp}_{Y,\boldsymbol{y}} \,.\, G(\boldsymbol{y}) \wedge \varphi(\boldsymbol{y}, Y, \boldsymbol{Z})](\boldsymbol{t}) \quad \text{where } Y \text{ only occurs positively in } \varphi$$

where R and G are relations in σ or $=$, and $\boldsymbol{t}$ is a tuple over variables and constants.

Examples

- unions of conjunctive queries (in GNF)
- frontier-guarded tgds (in GNF):
  $\forall xyz\,((Rxy \wedge Ryz) \rightarrow \exists w(Twyz)) \equiv \neg\exists xyz\,(Rxy \wedge Ryz \wedge \neg\exists w(Twyz))$
- description logics including $\mathcal{ALC}$, $\mathcal{ALCHIO}$, $\mathcal{ELI}$ (in GNF)
- mu-calculus, even with backwards modalities

Guarded negation fixpoint logic (GNFP)

Fix some signature σ of relations and constants.

Syntax of GNFP$[\sigma]$

$$\varphi \ ::= \ R\boldsymbol{t} \ | \ Y\boldsymbol{t} \ | \ \varphi \wedge \varphi \ | \ \varphi \vee \varphi \ | \ \exists y(\psi(\boldsymbol{xy})) \ | \ G(\boldsymbol{x}) \wedge \neg\psi(\boldsymbol{x}) \ |$$
$$[\mathbf{lfp}_{Y,\boldsymbol{y}} \,.\, G(\boldsymbol{y}) \wedge \varphi(\boldsymbol{y}, Y, \boldsymbol{Z})](\boldsymbol{t}) \quad \text{where } Y \text{ only occurs positively in } \varphi$$

where R and G are relations in σ or =, and $\boldsymbol{t}$ is a tuple over variables and constants.

Examples

- unions of conjunctive queries (in GNF)
- frontier-guarded tgds (in GNF):
  $\forall xyz\,((Rxy \wedge Ryz) \rightarrow \exists w(Twyz)) \equiv \neg\exists xyz\,(Rxy \wedge Ryz \wedge \neg\exists w(Twyz))$
- description logics including $\mathcal{ALC}$, $\mathcal{ALCHIO}$, $\mathcal{ELI}$ (in GNF)
- mu-calculus, even with backwards modalities
- monadic Datalog

GNFP example

$$[\mathbf{lfp}_{Z,xy}\,.\,Sxy \wedge \exists uv(Rxu \wedge Ryv \wedge (Zuv \vee (Pu \wedge Pv)))](xy)$$

$$\begin{tikzcd}
a_1 \arrow[r] \arrow[d] & a_2 \arrow[r] \arrow[d] & a_3 \arrow[r, dotted] \arrow[d] & a_k \arrow[r] \arrow[d] & a_{k+1} \\
b_1 \arrow[r] & b_2 \arrow[r] & b_3 \arrow[r, dotted] & b_k \arrow[r] & b_{k+1}
\end{tikzcd}$$

Some nice computational properties for guarded fixpoint logics

Decidable satisfiability and finite satisfiability
(2EXPTIME in general, EXPTIME for fixed-width formulas in GFP)

[Grädel, Walukiewicz '99 ; Bárány, Segoufin, ten Cate '11; Bárány, Bojańczyk '12]

Some nice computational properties for guarded fixpoint logics

Decidable satisfiability and finite satisfiability
(2EXPTIME in general, EXPTIME for fixed-width formulas in GFP)

[Grädel, Walukiewicz '99 ; Bárány, Segoufin, ten Cate '11; Bárány, Bojańczyk '12]

Decidable boundedness
(given $\psi(\boldsymbol{y}, Y)$ positive in Y, is there $n \in \mathbb{N}$ such that for all $\mathfrak{A}$, $\psi^n_{\mathfrak{A}} = \psi^{n+1}_{\mathfrak{A}}$?)

[Blumensath, Otto, Weyer '14 ; Bárány, ten Cate, Otto '12 ; Benedikt, ten Cate, Colcombet, VB. '15]

Some nice computational properties for guarded fixpoint logics

Decidable satisfiability and finite satisfiability
(2EXPTIME in general, EXPTIME for fixed-width formulas in GFP)

[Grädel, Walukiewicz '99 ; Bárány, Segoufin, ten Cate '11; Bárány, Bojańczyk '12]

Decidable boundedness
(given $\psi(\boldsymbol{y}, Y)$ positive in Y, is there $n \in \mathbb{N}$ such that for all $\mathfrak{A}$, $\psi^n_{\mathfrak{A}} = \psi^{n+1}_{\mathfrak{A}}$?)

[Blumensath, Otto, Weyer '14 ; Bárány, ten Cate, Otto '12 ; Benedikt, ten Cate, Colcombet, VB. '15]

Constructive interpolation for UNFP

[Benedikt, ten Cate, VB. '15]

Why so many nice properties?

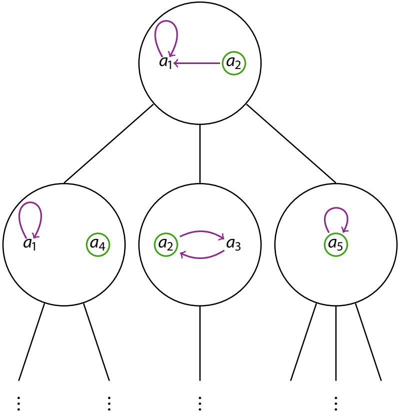
These guarded logics all have tree-like models (bounded tree width).

Why so many nice properties?

These guarded logics all have tree-like models (bounded tree width).

A structure $\mathfrak{A}$ has tree width $k - 1$ if it can be covered by (overlapping) bags of size at most k, arranged in a tree t s.t.

- every fact appears in some bag in t;
- for each element, the set of bags with this element is connected in t.

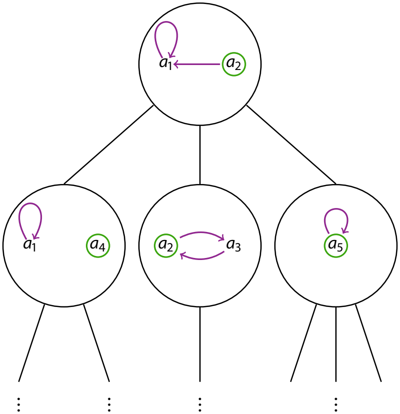


Why so many nice properties?

These guarded logics all have tree-like models (bounded tree width).

A structure $\mathfrak{A}$ has tree width $k - 1$ if it can be covered by (overlapping) bags of size at most k, arranged in a tree t s.t.

- every fact appears in some bag in t;
- for each element, the set of bags with this element is connected in t.



There is a natural encoding of these tree-like models (of some bounded tree width) as trees over a finite alphabet.

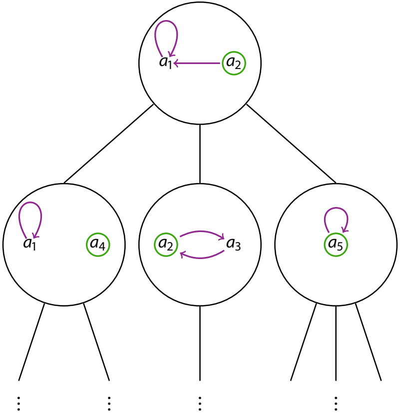
Why so many nice properties?

These guarded logics all have tree-like models (bounded tree width).

A structure $\mathfrak{A}$ has tree width $k - 1$ if it can be covered by (overlapping) bags of size at most k, arranged in a tree t s.t.

- every fact appears in some bag in t;
- for each element, the set of bags with this element is connected in t.

There is a natural encoding of these tree-like models (of some bounded tree width) as trees over a finite alphabet.



⇒ We can reason about tree encodings rather than relational structures.

Interpolation

$$\varphi \quad \models \quad \psi$$

Interpolation

$$\varphi \quad \models \quad \overset{\text{interpolant}}{\theta} \quad \models \quad \psi$$

only uses
relations
common to
φ and ψ

Interpolation

interpolant

$$\varphi \quad \models \quad \theta \quad \models \quad \psi$$

$\downarrow$

only uses
relations
common to
φ and ψ

Craig interpolation: θ depends on φ and ψ

Interpolation

interpolant

$$\varphi \quad \models \quad \theta \quad \models \quad \psi$$

$\downarrow$

only uses
relations
common to
φ and ψ

Craig interpolation: θ depends on φ and ψ

Uniform interpolation: θ depends only on φ and common signature (not on a particular ψ)

Uniform interpolation example

"P holds at x, and from every position y where P holds,
there is an R-neighbor z where P holds"

$$
\begin{aligned}
\varphi(x) &:= Px \wedge \forall y\big(Py \to \exists z(Ryz \wedge Pz)\big) \\
&\equiv Px \wedge \neg\exists y\big(Py \wedge \neg\exists z(Ryz \wedge Pz)\big)
\end{aligned}
$$

Uniform interpolation example

"P holds at x, and from every position y where P holds,
there is an R-neighbor z where P holds"

$$\begin{aligned} \varphi(x) \; &:= Px \wedge \forall y\big(Py \to \exists z(Ryz \wedge Pz)\big) \\ &\equiv Px \wedge \neg\exists y\big(Py \wedge \neg\exists z(Ryz \wedge Pz)\big) \end{aligned}$$

Uniform interpolant of φ over subsignature $\{R\}$
"there is an infinite R-path from x"

$$\begin{aligned} & \neg[\mathbf{lfp}_{Y,y} \,.\, \forall z(Ryz \to Yz)](x) \\ \equiv \;\; & \neg[\mathbf{lfp}_{Y,y} \,.\, \neg\exists z(Ryz \wedge \neg Yz)](x) \end{aligned}$$

Why study interpolation?

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

Why study interpolation?

- 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.

Why study interpolation?

- 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.

Why study interpolation?

- 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.

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

Interpolation for L_μ and UNFP

Theorem (D'Agostino, Hollenberg '00)

L_μ has effective uniform interpolation.

Interpolation for L_μ and UNFP

Theorem (D'Agostino, Hollenberg '00)

L_μ has effective uniform interpolation.

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.

Interpolation for L_μ and UNFP

Theorem (D'Agostino, Hollenberg '00)

L_μ has effective uniform interpolation.

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.

Proof strategy: Exploit tree-like models and ideas / results from
[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.

Uniform interpolation for UNFP^k

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

UNFP^k has effective uniform interpolation.

Proof structure:

Relational structures

Encodings of tree-like models of width k

$$\text{UNFP}^k\ \varphi \longrightarrow \text{L}_\mu\ \widehat{\varphi}$$

Uniform interpolation for UNFP^k

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

UNFP^k has effective uniform interpolation.

Proof structure:

Relational structures

Encodings of tree-like models of width k

$$
\begin{array}{ccc}
\text{UNFP}^k\ \varphi & \longrightarrow & \mathsf{L}_\mu\ \widehat{\varphi} \\
 & & \big\downarrow \ \text{[D'Agostino, Hollenberg'00]} \\
 & & \mathsf{L}_\mu\ \widehat{\theta}
\end{array}
$$

over subsignature encoding

Uniform interpolation for UNFP^k

Theorem (Benedikt, ten Cate, VB. ’15)

UNFP^k has effective uniform interpolation.

Proof structure:

$$
\begin{tikzcd}
\text{Relational structures} & \text{Encodings of tree-like models of width } k \\
\mathrm{UNFP}^k\ \varphi \arrow[r] & \mathrm{L}_\mu\ \widehat{\varphi} \arrow[d, "\text{[D'Agostino, Hollenberg'00]}"] \\
\underset{\text{over subsignature}}{\mathrm{UNFP}^k\ \theta} & \underset{\text{over subsignature encoding}}{\mathrm{L}_\mu\ \widehat{\theta}} \arrow[l]
\end{tikzcd}
$$

Summary of interpolation results

UNFP has effective interpolation,
and the construction takes advantage
of its tree-like models.

	K	GF	UNF	GNF	L_μ	GFP	UNFP	GNFP
Craig interpolation	✓	✗	✓	✓	✓	✗	✓	✗

Can we go further?

Can we go further?

GNFP$^{\mathbf{UP}}$: extend GNFP with parameters in fixpoint
(while retaining restrictions on negation).

Can we go further?

$\mathbf{GNFP}^{\mathbf{UP}}$: extend GNFP with parameters in fixpoint
(while retaining restrictions on negation).

"from y, it is possible to R-reach some P-element"

$$[\texttt{Reach-}P](y) := [\mathbf{lfp}_{Y,y} \,.\, \exists z(Ryz \wedge (Pz \vee Yz))](y)$$

"from y, it is possible to R-reach x"

$$[\texttt{Reach-}x](y) := [\mathbf{lfp}_{Y,y} \,.\, \exists z(Ryz \wedge (z = x \vee Yz))](y)$$

Can we go further?

GNFP$^{\text{UP}}$: extend GNFP with parameters in fixpoint
(while retaining restrictions on negation).

"from y, it is possible to R-reach some P-element"

$$[\texttt{Reach-}P](y) := [\mathbf{lfp}_{Y,y} \,.\, \exists z(Ryz \wedge (Pz \vee Yz))](y)$$

"from y, it is possible to R-reach x"

$$[\texttt{Reach-}x](y) := [\mathbf{lfp}_{Y,y} \,.\, \exists z(Ryz \wedge (z = x \vee Yz))](y)$$

Can we go further?

GNFP$^{\text{UP}}$: extend GNFP with parameters in fixpoint
(while retaining restrictions on negation).

"from y, it is possible to R-reach some P-element"

$$[\texttt{Reach-}P](y) := [\mathbf{lfp}_{Y,y} \,.\, \exists z(Ryz \wedge (Pz \vee Yz))](y)$$

"from y, it is possible to R-reach x"

$$[\texttt{Reach-}x](y) := [\mathbf{lfp}_{Y,y} \,.\, \exists z(Ryz \wedge (z = x \vee Yz))](y)$$

Subsumes

C2RPQs (conjunctive 2-way regular path queries) and
MQs and GQs [Rudolph, Krötsch '13 ; Bourhis, Krötsch, Rudolph '15]

Can we go further?

GNFP$^{\mathbf{UP}}$: extend GNFP with parameters in fixpoint
(while retaining restrictions on negation).

"from y, it is possible to R-reach some P-element"

$$[\texttt{Reach-}P](y) := [\mathbf{lfp}_{Y,y}\,.\,\exists z(Ryz \wedge (Pz \vee Yz))](y)$$

"from y, it is possible to R-reach x"

$$[\texttt{Reach-}x](y) := [\mathbf{lfp}_{Y,y}\,.\,\exists z(Ryz \wedge (z = x \vee Yz))](y)$$

Subsumes

C2RPQs (conjunctive 2-way regular path queries) and
MQs and GQs [Rudolph, Krötsch '13 ; Bourhis, Krötsch, Rudolph '15]

But still has tree-like models!

GNFP^{UP}: extend GNFP with parameters in fixpoint
(while retaining restrictions on negation).

$\mathrm{GNFP}^{\mathrm{UP}}$

$\mathrm{GNFP}^{\mathrm{UP}}$: extend GNFP with parameters in fixpoint
(while retaining restrictions on negation).

Theorem (Benedikt, Bourhis, VB. unpublished)

Satisfiability is decidable for $\varphi \in \mathrm{GNFP}^{\mathrm{UP}}$ in $(n+2)$-EXPTIME,
where n is "nesting depth of UCQ-shaped formulas with parameters" in φ.

Boundedness is decidable for $\varphi \in \mathrm{GNFP}^{\mathrm{UP}}$.

GNFP^{UP}

GNFP^{UP}: extend GNFP with parameters in fixpoint
(while retaining restrictions on negation).

Theorem (Benedikt, Bourhis, VB. unpublished)

Satisfiability is decidable for $\varphi \in \text{GNFP}^{\text{UP}}$ in $(n+2)$-EXPTIME,
where n is "nesting depth of UCQ-shaped formulas with parameters" in φ.

Boundedness is decidable for $\varphi \in \text{GNFP}^{\text{UP}}$.

Open questions

Does GNFP^{UP} have interpolation?

Is finite satisfiability decidable for GNFP^{UP}?

Conclusion

Guarded fixpoint logics are expressive logics
with nice computational properties
coming from their tree-like models.

Examples

Expressible in GNFP

R is symmetric

$$\begin{aligned} &\forall xy(Rxy \to Ryx) \\ \equiv\ &\neg\exists xy(Rxy \land \neg Ryx) \end{aligned}$$

Every element has an R-successor

$$\begin{aligned} &\forall x(\exists y(Rxy)) \\ \equiv\ &\neg\exists x(\neg\exists y(Rxy)) \end{aligned}$$

Every element is on R-cycle of length 3

$$\begin{aligned} &\forall x\exists yz(Rxy \land Ryz \land Rzx) \\ \equiv\ &\neg\exists x(\neg\exists yz(Rxy \land Ryz \land Rzx)) \end{aligned}$$

Not expressible in GNFP

R is total

$$\begin{aligned} &\forall xy(Rxy \lor Ryx) \\ \equiv\ &\neg\exists xy(\neg Rxy \land \neg Ryx) \end{aligned}$$

Every element has a unique R-successor

$$\begin{aligned} &\forall x\exists y(Rxy \land \forall z(Rxz \to y = z)) \\ \equiv\ &\neg\exists x(\neg\exists y(Rxy \land \neg\exists z(Rxz \land y \neq z))) \end{aligned}$$

Every element is on R-cycle

$$\begin{aligned} &\forall x\,[\mathbf{lfp}^{x}_{Y,y}.\exists z(Ryz \land (z = x \lor Yz))](x) \\ \equiv\ &\neg\exists x\,[\mathbf{lfp}^{x}_{Y,y}.\exists z(Ryz \land (z = x \lor Yz))](x) \end{aligned}$$

Examples, continued

Expressible in GNFP

R is well-founded:

$$\forall yz \Big(Ryz \to [\mathbf{lfp}_{Y,y}.\forall x(Rxy \to Yx)](y) \Big)$$

$$\equiv \neg\exists yz \Big(Ryz \wedge \neg[\mathbf{lfp}_{Y,y}.\neg\exists x(Rxy \wedge \neg Yx)](y) \Big)$$