A step up in expressiveness of decidable fixpoint logics

Michael Vanden Boom

University of Oxford

Oxford OASIS Seminar
April 2016

Based on joint work with
Michael Benedikt and Pierre Bourhis

Fixpoint logics

Fixpoint logics can express dynamic, recursive properties.

Example

binary relation R, unary relation P

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

$$[\texttt{Reach-}P](w)$$

Fixpoint logics

Fixpoint logics can express dynamic, recursive properties.

Example

binary relation R, unary relation P

“from w, it is possible to R-reach some P-element”

$$[\mathbf{lfp}_{Y,y}\,.\,\exists z(Ryz \wedge (Pz \vee Yz))](w)$$

LFP

LFP: extension of first-order logic with fixpoint formulas $[\mathbf{lfp}_{Y,\boldsymbol{y}}.\psi(\boldsymbol{y},Y)](\boldsymbol{w})$ for $\psi(\boldsymbol{y},Y)$ 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 $[\mathbf{lfp}_{Y,\boldsymbol{y}}.\psi(\boldsymbol{y},Y)]_{\mathfrak{A}} := \bigcup_\alpha \psi^\alpha_{\mathfrak{A}}$

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

LFP

LFP: extension of first-order logic with fixpoint formulas $[\mathbf{lfp}_{Y,\boldsymbol{y}}.\psi(\boldsymbol{y}, Y)](\boldsymbol{w})$ for $\psi(\boldsymbol{y}, Y)$ 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: $\mathfrak{A}, \boldsymbol{a} \models [\mathbf{lfp}_{Y,\boldsymbol{y}}.\psi(\boldsymbol{y}, Y)](\boldsymbol{w})$ iff $\boldsymbol{a} \in \bigcup_\alpha \psi_{\mathfrak{A}}^\alpha$

Examples

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

$$[\mathbf{lfp}_{Y,y}\,.\,\exists z(Ryz \wedge (Pz \vee Yz))](w)$$

$$a_1 \longrightarrow a_2 \longrightarrow a_3 \cdots\cdots\to a_k \longrightarrow a_{k+1}$$

Examples

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

$$[\mathbf{lfp}_{Y,y} \, . \, \exists z(Ryz \wedge (Pz \vee Yz))](w)$$

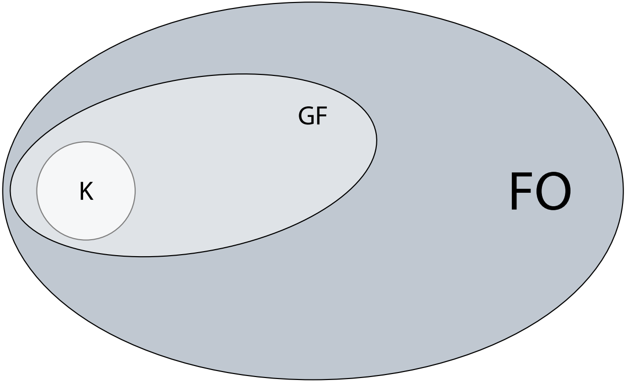
$$a_1 \longrightarrow a_2 \longrightarrow a_3 \cdots\cdots\cdots\cdots\rightarrow a_k \longrightarrow a_{k+1}$$

"from w, it is possible to R-reach x", i.e. "(w, x) is in the transitive closure of R"

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

(Free first-order variable x in the fixpoint predicate is called a parameter.)

Some decidable fragments of first-order logic

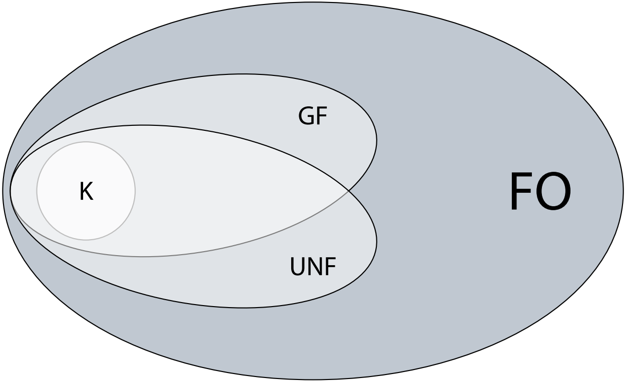


constrain
quantification

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

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

Some decidable fragments of first-order logic



constrain quantification

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

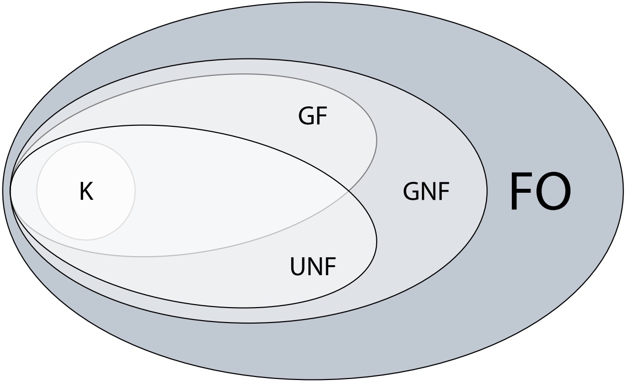
[Andréka, van Benthem, Németi '95-'98]

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

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

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

Guarded negation fragment of first-order logic

Let σ be a signature of relations and constants.

Syntax of GNF$[\sigma]$

$$\varphi \ ::= \ R\,\boldsymbol{t} \ \mid \ Y\,\boldsymbol{t} \ \mid \ \varphi \wedge \varphi \ \mid \ \varphi \vee \varphi \ \mid \ \exists y(\psi(\boldsymbol{x}y)) \ \mid \ G(\boldsymbol{x}) \wedge \neg\psi(\boldsymbol{x})$$

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

"There is an R-cycle of length 3"

$$\exists xyz(Rxy \wedge Ryz \wedge Rzx)$$

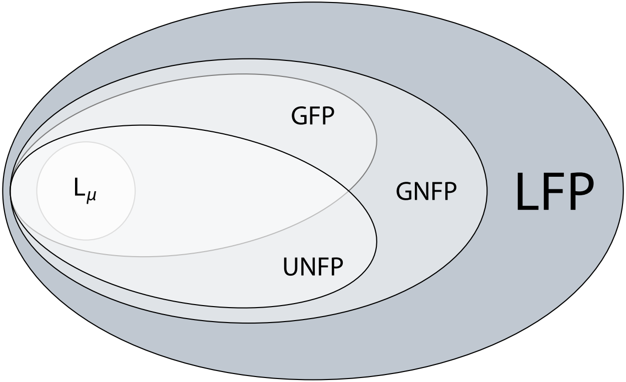
"R is symmetric"

$$\forall xy(Rxy \rightarrow Ryx) \ \equiv \ \neg\exists xy(Rxy \wedge \neg Ryx)$$

"Every element has an R-successor "

$$\forall x(\exists y(Rxy)) \ \equiv \ \neg\exists x(\neg\exists y(Rxy)) \ \equiv \ \neg\exists x(x = x \wedge \neg\exists y(Rxy))$$

Some decidable fragments of LFP (fixpoint extension of FO)



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]
[Grädel, Walukiewicz '99]

constrain negation

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

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

Guarded negation fixpoint logic (GNFP)

Let σ be a signature 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.

Guarded negation fixpoint logic (GNFP)

Let σ be a signature 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.

Restrictions on fixpoint operator:

- must define a guarded relation
  (tuples in the fixpoint must be guarded by an atom from σ)
- cannot use parameters

Examples

In GNFP:

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

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

Examples

In GNFP:

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

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

$$[\mathbf{lfp}_{Y,y}\,.\,\exists z(Ryz \wedge (Pz \vee Yz))](w)$$

Examples

In GNFP:

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

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

$$[\mathbf{lfp}_{Y,y} \,.\, \exists z(Ryz \wedge (Pz \vee Yz))](w)$$

Not in GNFP:

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

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?

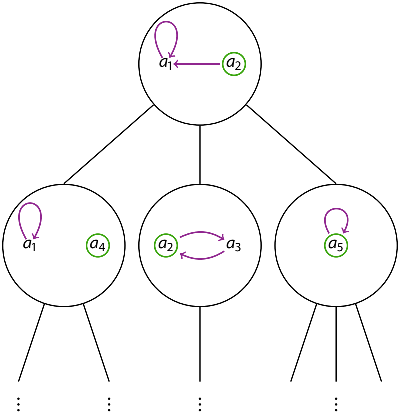
These guarded logics all have tree-like models (bounded tree width).

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.

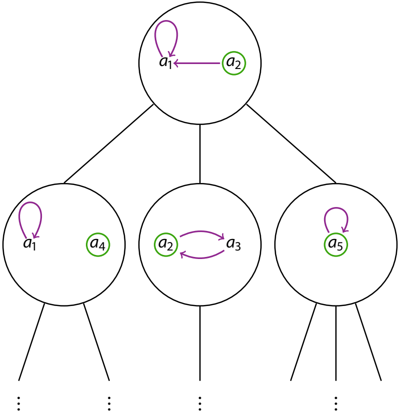


Why so many nice properties?

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.



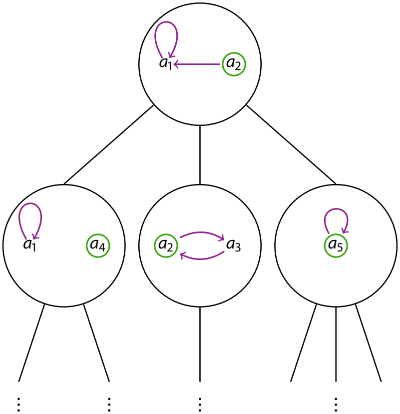
There is a natural encoding of these tree-like models (of some bounded tree width) as trees over a finite alphabet.

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.

$\Rightarrow$ We can reason about tree encodings rather than relational structures.

Automata for satisfiability

We can use tree automata to analyze these tree encodings.

$$\underset{\text{GNFP}}{\varphi} \quad \rightarrow \quad \underset{\substack{\text{2-way alternating} \\ \text{parity tree automaton}}}{\mathcal{A}_{\varphi}}$$

Automata for satisfiability

We can use tree automata to analyze these tree encodings.

$$\varphi \quad \rightarrow \quad \mathcal{A}_\varphi$$

GNFP

2-way alternating parity tree automaton

φ satisfiable? $\leftarrow$ $L(\mathcal{A}_\varphi)$ empty?

[Vardi '98]

Automata for satisfiability

We can use tree automata to analyze these tree encodings.

φ GNFP	$\rightarrow$	$\mathcal{A}_\varphi$ 2-way alternating parity tree automaton
φ satisfiable?	$\leftarrow$	$L(\mathcal{A}_\varphi)$ empty? [Vardi '98]

Theorem (Bárány, ten Cate, Segoufin '11)

Satisfiability is decidable for $\varphi \in$ GNFP in 2EXPTIME.

Game for least fixpoint

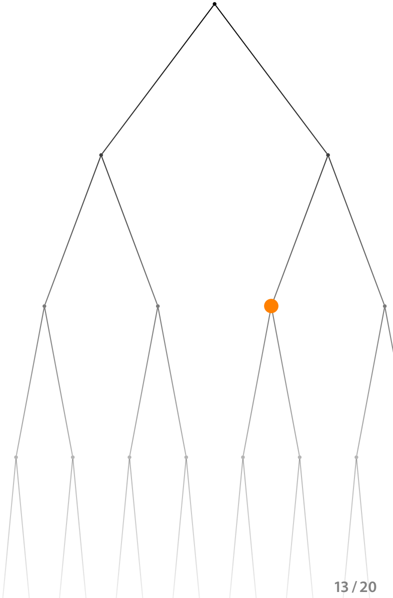
Game for testing if
$\mathfrak{A}, \boldsymbol{a} \models [\mathbf{lfp}_{Y,\boldsymbol{y}}.G(\boldsymbol{y}) \land \psi(\boldsymbol{y}, Y)](\boldsymbol{x})$.

Initial position $\boldsymbol{y} := \boldsymbol{a}$.

Game from position $\boldsymbol{y}$:

- Eve chooses Y such that
  $\mathfrak{A} \models G(\boldsymbol{y}) \land \psi(\boldsymbol{y}, Y)$
  (if it is not possible, she loses).
- Adam chooses some new $\boldsymbol{y}' \in Y$
  (if it is not possible, he loses).
- Game continues in next round from position $\boldsymbol{y} := \boldsymbol{y}'$.

Adam wins if the game continues forever.



Game for least fixpoint

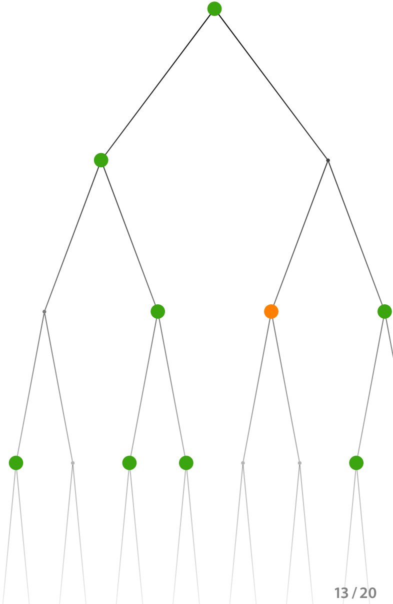
Game for testing if
$\mathfrak{A}, \boldsymbol{a} \models [\mathbf{lfp}_{Y,\boldsymbol{y}}.G(\boldsymbol{y}) \wedge \psi(\boldsymbol{y}, Y)](\boldsymbol{x})$.

Initial position $\boldsymbol{y} := \boldsymbol{a}$.

Game from position $\boldsymbol{y}$:

- Eve chooses Y such that
  $\mathfrak{A} \models G(\boldsymbol{y}) \wedge \psi(\boldsymbol{y}, Y)$
  (if it is not possible, she loses).
- Adam chooses some new $\boldsymbol{y}' \in Y$
  (if it is not possible, he loses).
- Game continues in next round from
  position $\boldsymbol{y} := \boldsymbol{y}'$.

Adam wins if the game continues forever.



Game for least fixpoint

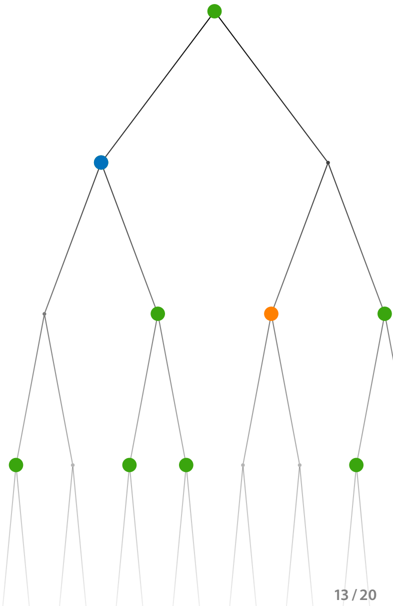
Game for testing if
$\mathfrak{A}, \boldsymbol{a} \models [\mathbf{lfp}_{Y,\boldsymbol{y}}.G(\boldsymbol{y}) \wedge \psi(\boldsymbol{y}, Y)](\boldsymbol{x})$.

Initial position $\boldsymbol{y} := \boldsymbol{a}$.

Game from position $\boldsymbol{y}$:

- Eve chooses Y such that
  $\mathfrak{A} \models G(\boldsymbol{y}) \wedge \psi(\boldsymbol{y}, Y)$
  (if it is not possible, she loses).
- Adam chooses some new $\boldsymbol{y}' \in Y$
  (if it is not possible, he loses).
- Game continues in next round from position $\boldsymbol{y} := \boldsymbol{y}'$.

Adam wins if the game continues forever.



Game for least fixpoint

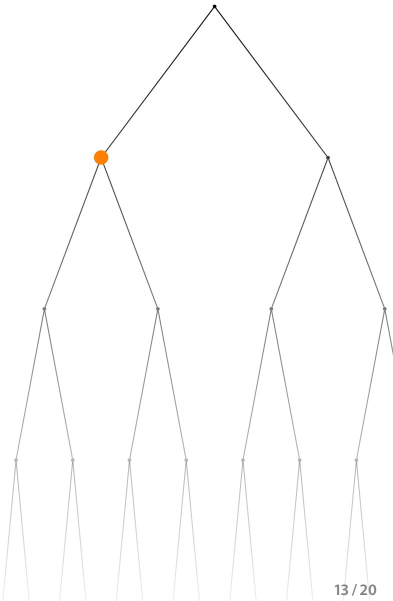
Game for testing if
$\mathfrak{A}, \boldsymbol{a} \models [\mathbf{lfp}_{Y,\boldsymbol{y}}.G(\boldsymbol{y}) \wedge \psi(\boldsymbol{y}, Y)](\boldsymbol{x})$.

Initial position $\boldsymbol{y} := \boldsymbol{a}$.

Game from position $\boldsymbol{y}$:

- Eve chooses Y such that
  $\mathfrak{A} \models G(\boldsymbol{y}) \wedge \psi(\boldsymbol{y}, Y)$
  (if it is not possible, she loses).
- Adam chooses some new $\boldsymbol{y}' \in Y$
  (if it is not possible, he loses).
- Game continues in next round from position $\boldsymbol{y} := \boldsymbol{y}'$.

Adam wins if the game continues forever.



Game for least fixpoint

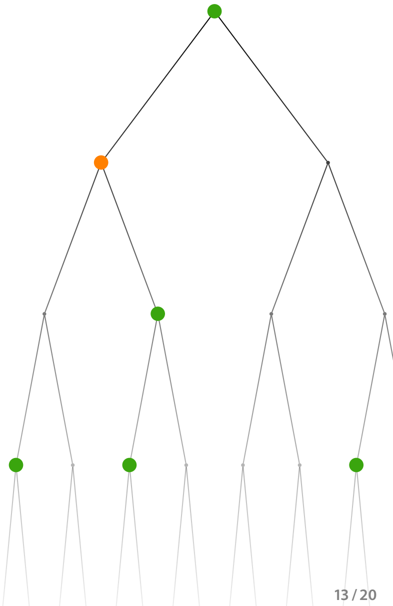
Game for testing if
$\mathfrak{A}, \boldsymbol{a} \models [\mathbf{lfp}_{Y,\boldsymbol{y}}.G(\boldsymbol{y}) \wedge \psi(\boldsymbol{y}, Y)](\boldsymbol{x})$.

Initial position $\boldsymbol{y} := \boldsymbol{a}$.

Game from position $\boldsymbol{y}$:

- Eve chooses Y such that $\mathfrak{A} \models G(\boldsymbol{y}) \wedge \psi(\boldsymbol{y}, Y)$ (if it is not possible, she loses).
- Adam chooses some new $\boldsymbol{y}' \in Y$ (if it is not possible, he loses).
- Game continues in next round from position $\boldsymbol{y} := \boldsymbol{y}'$.

Adam wins if the game continues forever.



Game for least fixpoint

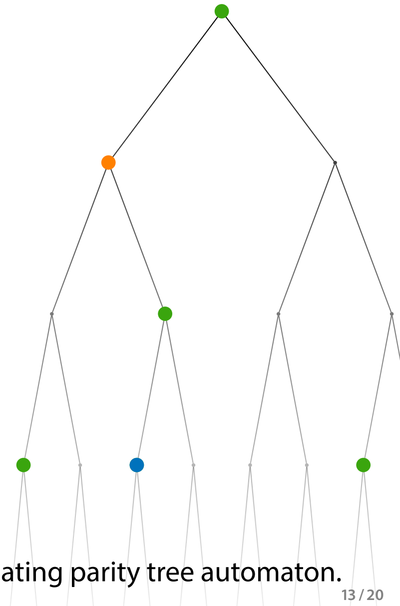
Game for testing if
$\mathfrak{A}, \boldsymbol{a} \models [\mathbf{lfp}_{Y,\boldsymbol{y}}.G(\boldsymbol{y}) \wedge \psi(\boldsymbol{y}, Y)](\boldsymbol{x})$.

Initial position $\boldsymbol{y} := \boldsymbol{a}$.

Game from position $\boldsymbol{y}$:

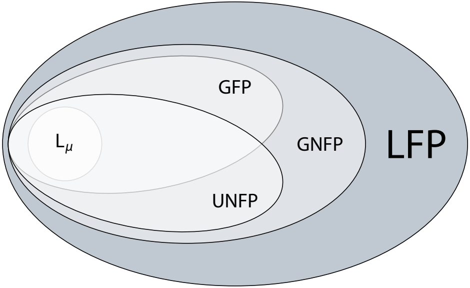
- Eve chooses Y such that $\mathfrak{A} \models G(\boldsymbol{y}) \wedge \psi(\boldsymbol{y}, Y)$ (if it is not possible, she loses).
- Adam chooses some new $\boldsymbol{y}' \in Y$ (if it is not possible, he loses).
- Game continues in next round from position $\boldsymbol{y} := \boldsymbol{y}'$.

Adam wins if the game continues forever.



Game can be implemented by a 2-way alternating parity tree automaton.

Can we go further?

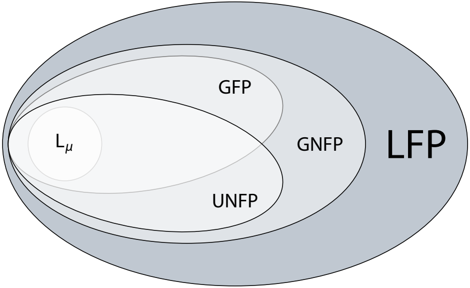


Recall the restrictions on the fixpoint operators in GNFP:

- must define a guarded relation
- cannot use parameters

Which of these restrictions are essential for decidability?

Can we go further?



Recall the restrictions on the fixpoint operators in GNFP:

- must define a guarded relation
- cannot use parameters

Which of these restrictions are essential for decidability?
Answer: only first one!

GNFP$^{\mathrm{UP}}$

GNFP$^{\mathrm{UP}}$: extend GNFP with unguarded parameters in fixpoint

GNFP$^{\mathrm{UP}}$

GNFP$^{\mathrm{UP}}$: extend GNFP with unguarded parameters in fixpoint

Syntax of GNFP$^{\mathrm{UP}}[\sigma]$

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

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

$\mathrm{GNFP}^{\mathrm{UP}}$: extend GNFP with unguarded parameters in fixpoint

Syntax of $\mathrm{GNFP}^{\mathrm{UP}}[\sigma]$

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

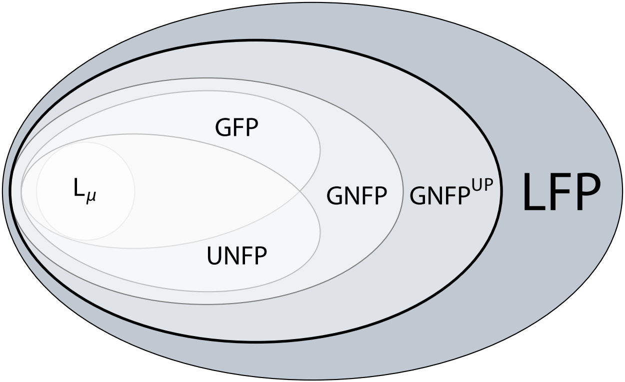
where R and G are relations in σ or $=$, and $\boldsymbol{t}$ is a tuple over variables and constants.

Example

$\mathrm{GNFP}^{\mathrm{UP}}$ can express the transitive closure of a binary relation R using

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

Expressivity of GNFP^{UP}



GNFP^{UP} also subsumes

C2RPQs (conjunctive 2-way regular path queries)

$$\exists xyz\,\big(\,[R^{*}S](x,y) \wedge [S \mid R](y,z) \wedge P(z)\,\big)$$

MQs and GQs [Rudolph, Krötsch '13 ; Bourhis, Krötsch, Rudolph '15]

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

GNFP$^{\mathrm{UP}}$ still has tree-like models
$\Rightarrow$ amenable to tree automata techniques

Unlike other guarded logics, satisfiability testing for $\varphi \in$ GNFP$^{\mathrm{UP}}$ is non-elementary, with running time

$$2^{2^{\cdot^{\cdot^{\cdot^{2^{f(|\varphi|)}}}}}}$$

where the polynomial f and the height of the tower depend only on the parameter depth: the number of nested parameter changes in the formula.

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

$\mathrm{GNFP}^{\mathrm{UP}}$ still has tree-like models
$\Rightarrow$ amenable to tree automata techniques

Unlike other guarded logics, satisfiability testing for $\varphi \in \mathrm{GNFP}^{\mathrm{UP}}$ is non-elementary, with running time

$$2^{2^{\cdot^{\cdot^{\cdot^{2^{f(|\varphi|)}}}}}}$$

where the polynomial f and the height of the tower depend only on the parameter depth: the number of nested parameter changes in the formula.

Theorem (Benedikt, Bourhis, VB. '16)

Satisfiability is decidable for $\varphi \in \mathrm{GNFP}^{\mathrm{UP}}$ in $(n+2)$-EXPTIME, where n is the parameter depth of φ.

Skirting undecidability

It is known that satisfiability is undecidable for GF when certain relations are required to be transitive. [Grädel ’99, Ganzinger et al. ’99]

Skirting undecidability

It is known that satisfiability is undecidable for GF when certain relations are required to be transitive. [Grädel ’99, Ganzinger et al. ’99]

$\mathrm{GNFP}^{\mathrm{UP}}$ can express the transitive closure of a binary relation R using

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

But it cannot enforce that R is transitive.

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

Theorem (Benedikt, Bourhis, VB. '16)

The following boundedness problem is decidable:

Instance: $G(\boldsymbol{y}) \wedge \psi(\boldsymbol{x}, \boldsymbol{y}, Y) \in$ GNF, positive in Y

Question: is there $n \in \mathbb{N}$ s.t. for all structures $\mathfrak{A}$, the least fixpoint $[\mathbf{lfp}_{Y,\boldsymbol{y}}\,.\, G(\boldsymbol{y}) \wedge \psi(\boldsymbol{x}, \boldsymbol{y}, Y)]_{\mathfrak{A}}$ is always reached within n iterations?

Boundedness for GNFP$^{\text{UP}}$

Theorem (Benedikt, Bourhis, VB. '16)

The following boundedness problem is decidable:

Instance: $G(\boldsymbol{y}) \wedge \psi(\boldsymbol{x}, \boldsymbol{y}, Y) \in$ GNF, positive in Y

Question: is there $n \in \mathbb{N}$ s.t. for all structures $\mathfrak{A}$, the least fixpoint $[\mathbf{lfp}_{Y,\boldsymbol{y}} \,.\, G(\boldsymbol{y}) \wedge \psi(\boldsymbol{x}, \boldsymbol{y}, Y)]_{\mathfrak{A}}$ is always reached within n iterations?

$\Rightarrow$ For $\psi(\boldsymbol{x}, \boldsymbol{y}, Y) \in$ GNF positive in Y, it is decidable whether $[\mathbf{lfp}_{Y,\boldsymbol{y}} \,.\, G(\boldsymbol{y}) \wedge \psi(\boldsymbol{x}, \boldsymbol{y}, Y)](\boldsymbol{w})$ can be expressed in FO.

Conclusion

We can allow unguarded parameters
in guarded fixpoint logics while still retaining
nice model theoretic and computational properties.

Theorem (Benedikt, Bourhis, VB. ’16)

Satisfiability is decidable for $\varphi \in \mathrm{GNFP}^{\mathrm{UP}}$ in $(n+2)$-EXPTIME, where n is the parameter depth of φ.

Some boundedness problems are decidable for $\mathrm{GNFP}^{\mathrm{UP}}$.

Conclusion

We can allow unguarded parameters
in guarded fixpoint logics while still retaining
nice model theoretic and computational properties.

Theorem (Benedikt, Bourhis, VB. ’16)

Satisfiability is decidable for $\varphi \in \mathrm{GNFP}^{\mathrm{UP}}$ in $(n+2)$-EXPTIME, where n is the parameter depth of φ.

Some boundedness problems are decidable for $\mathrm{GNFP}^{\mathrm{UP}}$.

Open questions

Is finite satisfiability decidable for $\mathrm{GNFP}^{\mathrm{UP}}$?

Does $\mathrm{GNFP}^{\mathrm{UP}}$ have interpolation?