# A step up in expressiveness of decidable fixpoint logics

Michael Benedikt[1], Pierre Bourhis[2], and **Michael Vanden Boom**[1]

[1]University of Oxford
[2]CNRS CRIStAL, Université Lille 1, INRIA Lille

LICS 2016
New York, USA

# 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"

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

**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 $\psi$ 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 $\psi$ 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 $[\mathbf{lfp}_{Y,\boldsymbol{y}}.\psi(\boldsymbol{y},Y)]_{\mathfrak{A}} := \bigcup_\alpha \psi^\alpha_{\mathfrak{A}}$

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

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

# Examples

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

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

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

# Examples

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

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

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

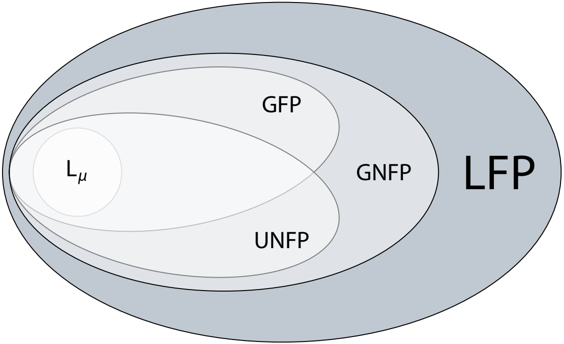
"from $w$, it is possible to $R$-reach $x$", i.e. "$(w, x)$ is in the transitive closure of $R$"

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

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

# Some decidable fragments of LFP (fixpoint extension of FO)

The family of “guarded” fixpoint logics has decidable satisfiability.




Guarded fixpoint logic (GFP): Andréka, van Benthem, Németi ’95-’98; Grädel, Walukiewicz ’99

Unary negation fixpoint logic (UNFP): ten Cate, Segoufin ’11

Guarded negation fixpoint logic (GNFP): Bárány, ten Cate, Segoufin ’11

# Guarded negation fixpoint logic (GNFP)

Let $\sigma$ 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 $\sigma$ or $=$, and $\boldsymbol{t}$ is a tuple over variables and constants.

# Guarded negation fixpoint logic (GNFP)

Let $\sigma$ 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 $\sigma$ 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 $\sigma$ or =)
- cannot use parameters

# Satisfiability

These guarded fixpoint logics all have the tree-like model property (models with tree decompositions of bounded tree-width)

$\Rightarrow$ amenable to tree automata techniques

# Satisfiability

These guarded fixpoint logics all have the tree-like model property
(models with tree decompositions of bounded tree-width)

⇒ amenable to tree automata techniques

**Theorem** (Grädel, Walukiewicz ’99; Bárány, Segoufin, ten Cate ’11; Bárány, Bojańczyk ’12)

Satisfiability and finite satisfiability are decidable for guarded fixpoint logics
(2EXPTIME in general, EXPTIME for fixed-width formulas in GFP).

**Idea:** Reduce to tree automaton emptiness test.

# Examples

**In GNFP:**

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

# Examples

**In GNFP:**

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

# Examples

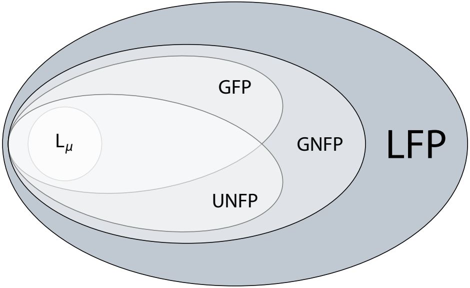
**In GNFP:**

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

**Not in GNFP:**

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

# Can we go further?

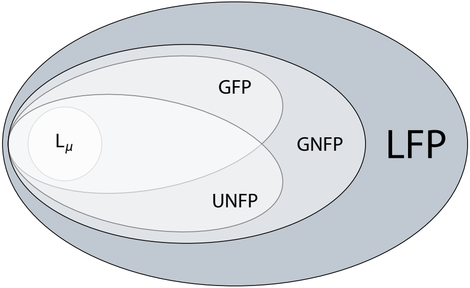



Recall the restrictions on the fixpoint operators in GNFP:

- 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$^{UP}$

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

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

## Syntax of 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 $\sigma$ or $=$, and $\boldsymbol{t}$ is a tuple over variables and constants.

# GNFP$^{\text{UP}}$

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

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

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

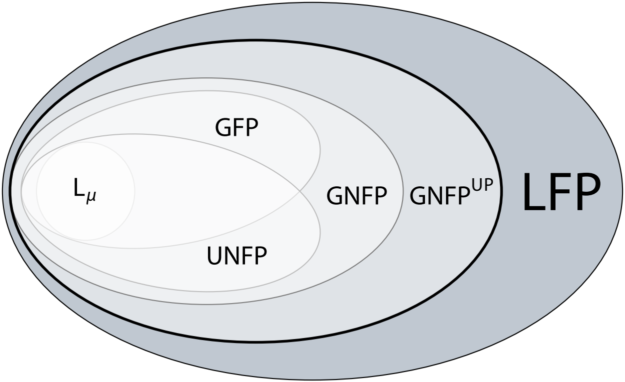
where $R$ and $G$ are relations in $\sigma$ or $=$, and $\boldsymbol{t}$ is a tuple over variables and constants.

**Example**

GNFP$^{\text{UP}}$ 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)
  $\exists xyz \left( [R^* S](x,y) \wedge [S \mid R](y,z) \wedge P(z) \right)$
- MQs and GQs [Rudolph, Krötzsch '13 ; Bourhis, Krötzsch, Rudolph '15]

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

GNFP$^{\mathrm{UP}}$ still has tree-like models
$\Rightarrow$ still 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 height of the tower depends only on the parameter depth: the number of nested parameter changes in the formula.

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

GNFP$^{\mathrm{UP}}$ still has tree-like models
$\Rightarrow$ still 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 height of the tower depends only on the parameter depth: the number of nested parameter changes in the formula.

**Theorem**

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

# Skirting undecidability

It is known that satisfiability is undecidable for GFP (even without fixpoints) when certain relations are required to be transitive.
[Grädel ’99, Ganzinger et al. ’99]

## Skirting undecidability

It is known that satisfiability is undecidable for GFP (even without fixpoints) when certain relations are required to be transitive.
[Grädel '99, Ganzinger et al. '99]

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.

# FO-definability

**Theorem**

It is decidable whether $[\mathbf{lfp}_{Y,\boldsymbol{y}} \,.\, G(\boldsymbol{y}) \wedge \psi(\boldsymbol{x}, \boldsymbol{y}, Y)](\boldsymbol{w}) \in \mathrm{GNFP}^{\mathrm{UP}}$ can be expressed in FO (when $\psi$ does not use any additional fixpoints).

It is decidable whether a C2RPQ can be expressed in FO.

**Idea:** Adapt automata for $\mathrm{GNFP}^{\mathrm{UP}}$, and reduce to a boundedness question for cost automata (automata with counters).

## Conclusion

We can allow unguarded parameters
in guarded fixpoint logics.

### Contributions

We showed that:

- tree automata techniques can be used to analyze $\mathrm{GNFP}^{\mathrm{UP}}$
- satisfiability is decidable for $\mathrm{GNFP}^{\mathrm{UP}}$, and the key factor impacting the complexity is the parameter depth
- some boundedness and FO-definability problems are decidable for $\mathrm{GNFP}^{\mathrm{UP}}$

## Conclusion

We can allow unguarded parameters
in guarded fixpoint logics.

### Contributions

We showed that:

- tree automata techniques can be used to analyze $\mathrm{GNFP}^{\mathrm{UP}}$
- satisfiability is decidable for $\mathrm{GNFP}^{\mathrm{UP}}$, and the key factor impacting the complexity is the parameter depth
- some boundedness and FO-definability problems are decidable for $\mathrm{GNFP}^{\mathrm{UP}}$

### Open question

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