Unary negation

Balder ten Cate, Luc Segoufin

Unary negation*

Balder ten Cate[1] and Luc Segoufin[2]

1 **University of California, Santa Cruz**
`http://users.soe.ucsc.edu/~btencate`
2 **INRIA and ENS Cachan, LSV**
`http://www-rocq.inria.fr/~segoufin`

Abstract

We study fragments of first-order logic and of least fixed point logic that allow only unary negation: negation of formulas with at most one free variable. These logics generalize many interesting known formalisms, including modal logic and the μ-calculus, as well as conjunctive queries and monadic Datalog. We show that satisfiability and finite satisfiability are decidable for both fragments, and we pinpoint the complexity of satisfiability, finite satisfiability, and model checking. We also show that the unary negation fragment of first-order logic is model-theoretically very well behaved. In particular, it enjoys Craig interpolation and the Beth property.

1998 ACM Subject Classification F.4.1 Mathematical Logic

Keywords and phrases Decidability, Logic, Unary negation

Digital Object Identifier 10.4230/LIPIcs.STACS.2011.344

1 Introduction

Vardi [24] raised the question "why is modal logic so robustly decidable?". His explanation was that modal logic enjoys a combination of three properties, namely (i) the *tree model property* (if a sentence has a model, it has a model which is a tree), (ii) *translatability into monadic second-order logic* (MSO), and thereby into tree automata, and (iii) the *finite model property* (if there is a model, there is also a finite one). The decidability of (finite) satisfiability follows immediately from these three properties. The guarded fragment of first-order logic (GFO) [1] was subsequently proposed as a large fragment of first-order logic that generalizes modal logic while retaining these properties. It consists of FO formulas in which all quantifiers are "guarded". GFO has the tree-like model property (if a sentence has a model, it has a model of bounded tree width), it can be interpreted into MSO (each formula can be transformed into a tree automata recognizing a tree decomposition of its models of bounded tree width) and it has the finite model property [1, 15].

In this paper we provide another, orthogonal generalization of modal logic that enjoys the same nice properties. We introduce UNFO, a fragment of FO in which *negation* is restricted to formulas having only one free variable. UNFO is incomparable in term of expressive power to GFO but it generalizes modal logic, as well as other formalisms, such as conjunctive queries, that are not contained in GFO. We show that UNFO has the tree-like model property, interpretation into MSO and the finite model property. Hence UNFO is robustly decidable.

We also introduce UNFP, which extends UNFO with least and greatest monadic fixpoints, in the same way that the μ-calculus extends modal logic, and guarded fixpoint logic GFP

* Balder ten Cate has been funded partially by the ERC grant Webdam, agreement 226513, and partially by the NSF grant IIS-0905276.

28th Symposium on Theoretical Aspects of Computer Science (STACS'11).
Editors: Thomas Schwentick, Christoph Dürr; pp. 344–355

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

extends GFO [16]. UNFP generalizes the μ-calculus but also monadic Datalog and remains incomparable with GFP. It still has the tree-like model property and can be interpreted into MSO, but it no longer has the finite model property. Nevertheless, we show that finite satisfiability for UNFP is decidable (recall that the decidability of finite satisfiability for GFP is still open at the time of submission[1]). The satisfiability problem is 2ExpTime-complete, both for UNFO and for UNFP, both on arbitrary and finite structures.

We also study the model checking problem. In contrast with GFO, whose model checking problem is PTime-complete [7], we show that for UNFO it is complete for $\mathrm{P}^{\mathrm{NP}[O(\log^2 n)]}$, providing one of the few natural complete problems for that complexity class. For UNFP, model checking is hard for P^{NP} and contained in $\mathrm{NP}^{\mathrm{NP}} \cap \mathrm{coNP}^{\mathrm{NP}}$. The gap between the upper-bound and the lower-bound reflects a similar open problem for GFP and the μ-calculus where the model checking problems lies between PTime and $\mathrm{NP} \cap \mathrm{coNP}$ [7].

UNFO is not only computationally but also model-theoretically very well behaved. We characterize the expressive power of UNFO in terms of an appropriate notion of invariance, and we show that UNFO has the Craig Interpolation Theorem as well as the Projective Beth Property. Note that Craig Interpolation fails for GFO [18]. On trees, UNFO and UNFP correspond to well known formalisms.

Unary negation vs. guarded quantification As mentioned, UNFO and GFO are incomparable in term of expressive power. For instance the properties "some node lies on a cycle of length 4" and "every node lies on a cycle of length 4" (and their negations) are definable in UNFO but not in GFO (and not even in GFP). Conversely, the property "the binary relation R is contained in binary relation S" (and its negation) is definable in GFO but not in UNFO (and not even in UNFP). See Section 6 for more discussion. In spite of these differences, our proofs often follow similar strategies as proofs for GFO and GFP.

Due to space limitations many proofs are omitted or only sketched. They will appear in the journal version of this paper.

2 Preliminaries

We deal with relational structures. We assume given a relational schema providing a finite set of relation symbols and fixing an arity to each relation. A *model*, or *structure*, over a relational schema is a set, the *domain*, together with an interpretation of each relation symbol of the schema as a relation over the domain of the arity given by the schema. A model is said to be *finite* if its domain is finite. We assume familiarity with first-order logic, FO, and least fixpoint logic, FP, over relational structures. We use classical syntax and semantics for FO and FP. In particular we write $M \models \phi(\overline{u})$ for the fact that the tuple $\overline{u}$ of elements of the model M makes the FO-formula ϕ true on M.

2.1 UNFO and UNFP

We define *unary negation FO*, UNFO, as the fragment of FO given by the following grammar (where R is an arbitrary relation name from the underlying schema):

$$\phi :: \quad R(\overline{x}) \mid x = y \mid \phi \wedge \phi \mid \phi \vee \phi \mid \exists x \phi \mid \neg\varphi(x)$$

[1] Possibly recently solved [4].

where φ has no free variables besides (possibly) x. Throughout this paper, we will keep using the notation $\varphi(x)$ to indicate that a formula has at most one free variable. In other words, UNFO is the restriction of FO where negation is only allowed if the subformula has at most one free variable. In particular $x \neq y$ is not expressible in UNFO.

We say that a formula of UNFO is in UN-normal form if, in the syntax tree of the formula, every existential quantifier (except for the root) is either directly below another existential quantification, or the subformula starting with that quantifier has at most one free variable. It is immediate to see that each formula of UNFO can be turned into an equivalent one in UN-normal form in linear time by "pulling out existential quantifiers" as much as possible. For instance the formula $\exists x R(x) \wedge \exists x \neg(\exists y S(x,y))$ is not in UN-normal form but it is equivalent to $\exists x \exists x' \;\; R(x) \wedge \neg(\exists y S(x',y))$ which is in UN-normal form.

A formula of UNFO is said to be *of width* k if, when put in UN-normal form, it uses at most k variables. We denote by UNFO^k all UNFO formulas of width k.

In order to define UNFP we introduce extra unary predicates that will serve for computing unary fixpoints. We denote the unary predicates given by the relational schema using the letters $P, Q \ldots$ and the unary predicates serving for computing the fixpoints by $X, Y \ldots$. By $\mathrm{UNFO}(\overline{X})$ we mean UNFO defined over the schema extended with the unary predicates $\overline{X}$. In particular it allows formulas of the form $\neg\phi(x, \overline{X})$. UNFP is the extension of $\mathrm{UNFO}(\overline{X})$ by means of the following least fixpoint construction:

$$[\mathrm{LFP}_{X,x}\ \phi(X, \overline{X}, x)](y)$$

where X occurs only positively in ϕ. An analogous greatest fixed point operator is definable by dualization. Note that no first-order parameters (i.e., free variables in the body of ϕ other than the bound variable x) are permitted.

Since UNFP is a syntactic fragment of least fixed point logic LFP, we omit the definition of the semantics, cf. [19]. The definition of the normal form naturally extend to UNFP. As in the case of UNFO, a UNFP formula *has width* k if, when put in UN-normal form, it uses at most k variables. In particular, a formula of UNFP has width k if all the $\mathrm{UNFO}(\overline{X})$-parts of its subformulas have width k. We denote by UNFP^k all UNFP formulas of width k.

The *negation depth* of a UNFO or UNFP formula will be an important parameter. It is the maximal nesting depth of negations in its syntax tree.

2.2 Logics that are contained in UNFO and UNFP

UNFO and UNFP generalize many known formalisms. We list here some well known logics that can be embedded into UNFO and UNFP.

Conjunctive queries and monadic datalog. A conjunctive query (CQ) is a query of the form $\exists x_1 \cdots x_n \; \tau_1 \wedge \cdots \wedge \tau_l$ where each τ_i is a positive atomic formula. Unions of conjunctive queries (UCQs) are contained in UNFO. Actually, UNFO can naturally be viewed as an extension of the UCQ language with unary negation. Similarly, monadic datalog (i.e., datalog queries in which all IDB relations are unary [12]) is contained in UNFP. As a matter of fact, if one allows the answer predicate of monadic datalog programs to have any arity, then monadic datalog corresponds precisely to the positive fragment of UNFP.

Modal logic and the μ-calculus. Modal logic is contained in UNFO. Indeed, the standard translation from modal logic to first-order logic produces first-order formulas that belong to UNFO. Similarly, the μ-calculus (and in fact the two-way μ-calculus) is contained in UNFP.

Unary conjunctive view logic. First-order unary-conjunctive-view logic (UCV) was introduced in [3] as a fragment of FO. A UCV query is an equality-free first-order formula

over a signature consisting of unary predicates only, but where each of these unary predicates is in fact a view defined by a unary conjunctive query. UCV queries can be translated into UNFO, more precisely into the fragment of UNFO that has negation depth 1 (as follows from the fact that equality-free monadic FO admits quantifier elimination, cf. [17]).

The temporal logic CTL*(X). CTL*(X) is the fragment of CTL* in which only the modality X ("next") is allowed (For the definition of the semantics, see [11]). CTL*(X) is a fragment of UNFO. The model checking problem for CTL*(X) is known to be complete for the complexity class $\mathrm{P}^{\mathrm{NP}[O(\log^2 n)]}$ [23]. We will show that, in fact, the model checking problem for full UNFO is $\mathrm{P}^{\mathrm{NP}[O(\log^2 n)]}$-complete.

Core XPath. On XML trees with all axes, it is known that *Core XPath* = FO^2 for unary queries while *Core XPath* = *UCQ-over*-FO^2-*unary-predicates* for binary queries [20]. It turns out that UNFO has the same expressive power as Core XPath, both for unary and for binary queries [10] and therefore UNFO characterizes Core XPath in a more uniform way. Similarly, Regular XPath embeds into UNFP.

Data tree patterns. In [13], *Boolean combinations of tree patterns* are studied as a query language for data trees (or, XML documents containing data). If we represent data trees by relational structures, then tree patterns can be seen as conjunctive queries. Therefore the Boolean combinations of tree patterns studied in [13] can be seen as a fragment of UNFO.

Description logics. The *conjunctive query answering problem for description logics* can be reduced to the UNFO entailment problem. Take for example the basic description logic $\mathcal{ALC}$. The conjunctive query answering problem for $\mathcal{ALC}$ is the following problem (cf. [2] for basic terminology): *Given a TBox T, an ABox A consisting of atomic formulas speaking about individuals (constant symbols) $c_1 \dots c_n$, a conjunctive query $q(x_1, \dots, x_k)$, and a tuple of constants $(c_{i_1}, \dots, c_{i_k})$, is $(c_{i_1}, \dots, c_{i_k})$ an answer to q in every model of $T \cup A$?*

It is easy to see that this is equivalent to the validity of the UNFO-entailment

$$\phi_T \wedge \bigwedge A[c_1/x_1, \dots, c_n/x_n] \quad \models \quad q(x_{i_1}, \dots, x_{i_k})$$

where ϕ_T is the UNFO-translation of T. It follows that conjunctive query answering for $\mathcal{ALC}$ is decidable and has the finite model property (cf. Remark 3). The same argument works not only for $\mathcal{ALC}$ but for any description logic whose TBoxes can be expressed in UNFP. Moreover, the argument works not only for conjunctive queries, but to any class of queries expressible in UNFP.

3 Model theory

In this section we give results about the expressive power of UNFO and UNFP, and we show that UNFO has Craig Interpolation Theorem and the Projective Beth Property. We only state the results here, the proofs will appear in the journal version of this paper.

Invariance for UN-bisimulations We define a game that captures model indistinguishability, and we use it to characterize the expressive power of UNFO and UNFP. The game is as follows: the two players maintain a single pair (a, b) of objects from the two structures. A move of Abelard consists of choosing a set X of points in one of the two structures. Then Eloise responds with a homomorphism from that set into a set of points in the other structure, where the homomorphism maps a to b (respectively b to a) if a (respectively b) belongs to the set X. Finally, Abelard picks a pair from the homomorphism and the players continue with that pair. The game is parametrized by the size of the sets chosen by Abelard at each round.

Equivalently, we can present the game in terms of a back-and-forth system:

▶ **Definition 1.** Let M, N be two structures. A UN-bisimulation (of width $k \geq 1$) is a binary relation $Z \subseteq M \times N$ such that the following hold for every pair $(a, b) \in Z$:

- For every finite set $X \subseteq \mathrm{dom}(M)$ (with $|X| \leq k$) there is a partial homomorphism $h : M \to N$ whose domain is X, such that $h(a) = b$ if $a \in X$, and such that every pair $(a', b') \in h$ belongs to Z.
- Likewise in the other direction, where $X \subseteq \mathrm{dom}(N)$.

We write $M \approx_{\mathrm{UN}} N$ if there is a non-empty UN-bisimulation between M and N, and we write $M \approx^k_{\mathrm{UN}} N$ if there is a non-empty UN-bisimulation of width k between M and N.

It is not difficult to see that UN-bisimulation implies UNFP-indistinguishability.

▶ **Proposition 1.** *For any $k \geq 1$, if $M \approx^k_{UN} N$ then M and N satisfy the same sentences of* UNFP^k*. In particular, if $M \approx_{UN} N$ then M and N satisfy the same sentences of* UNFP*.*

A similar invariance property holds for formulas with free variables. For simplicity, we only state a version of the result without reference to the width of formulas. Let a *UN-homomorphism* $h : M \to N$ be a homomorphism with the property that $M, a \approx_{\mathrm{UN}} N, h(a)$ for all $a \in \mathrm{dom}(M)$. We write $M, \overline{a} \to_{\mathrm{UN}} N, \overline{b}$ if there is a UN-homomorphism $h : M \to N$ such that $h(\overline{a}) = \overline{b}$. Then we have the following:

▶ **Proposition 2.** *If $M, \overline{a} \to_{\mathrm{UN}} N, \overline{b}$ and $M \models \phi(\overline{a})$ then $N \models \phi(\overline{b})$, for all* UNFP*-formulas $\phi(\overline{x})$.*

Tree-like model property and finite model property From the invariance for UN-bisimulation it follows by a standard infinite unraveling argument that UNFP has the tree-like model property. A more involved partial unraveling, using back-edges in order to keep the structure finite, can also be used to show that UNFO has the finite model property. We will not give the details of these constructions, as it turns out that both results will follow from the material presented in Section 4.

▶ **Theorem 2.** *Every satisfiable* UNFO *formula has a finite model.*

▶ **Theorem 3.** *Every satisfiable* UNFP *formula of width k has a model of tree-width $k - 1$.*

Note, that UNFP does *not* have the finite model property. This follows from the fact that UNFP contains the two-way μ-calculus which is known to lack the finite model property [8]. Indeed, if $\max(x)$ is shorthand for $\neg\exists y(E(x, y))$, then the formula

$$\exists x \max(x) \vee \exists x[\mathrm{GFP}_{X,y}\ \exists z(Xz \wedge E(z, y))](x)$$

is valid on finite structures (if a finite structure has no maximal elements, it must contain a cycle, and hence an infinite backward path) but it is false in the infinite structure $(\mathbb{N}, suc)$.

Characterization We have seen in Proposition 1 that UNFO sentences are first-order formulas that are closed under $\approx_{\mathrm{UN}}$-equivalence. It turns out that the converse is also true. Indeed, in the same way that bisimulation-invariance characterizes modal logic [6, 22] and guarded bisimulation-invariance characterizes the guarded fragment of FO [1], we will see that $\approx_{\mathrm{UN}}$-invariance characterizes UNFO. We state this result over arbitrary models. We believe the result can also be proved over finite structures. We postpone this issue to the journal version of this paper.

Call a FO sentence $\approx_{UN}$*-invariant* if for all structures $M \approx_{\mathrm{UN}} N$, we have $M \models \phi$ iff $N \models \phi$, and define $\approx^k_{\mathrm{UN}}$-invariance similarly. Then

▶ **Theorem 4.** UNFO *is the* $\approx_{UN}$*-invariant fragment of* FO*, and for all* $k \geq 1$*,* UNFO^k *is the* $\approx^k_{UN}$*-invariant fragment of* FO *(on arbitrary structures)*

The result above applies to sentences. A similar characterization can be obtained for formulas with free variables, using UN-homomorphisms instead of UN-bisimulations.

▶ **Theorem 5.** UNFO *formulas (with free variables) form the fragment of* FO *that is preserved under UN-homomorphisms.*

Craig interpolation and Beth definability We conclude the list of nice model-theoretic properties of UNFO by showing that it has Craig Interpolation Theorem and the Projective Beth Property. In fact, we can show strong versions of these results, which take into account also the width of formulas. This is remarkable, given that both Craig Interpolation and the Beth Property fail for the k-variable fragment of first-order logic, for all $k > 1$. Moreover, the results presented in this section hold both on arbitrary structures and on finite structures.

For all UNFO-formulas $\phi(\overline{x}), \psi(\overline{x})$, we write $\phi \models \psi$ to express that the first-order formula $\forall\overline{x}(\phi \to \psi)$ (which is not necessarily a UNFO-formula) is valid.

▶ Remark. For all UNFO-formulas $\phi(\overline{x}), \psi(\overline{x})$, $\phi \models \psi$ holds (on finite structures) if and only if the formula

$$\exists\overline{x}(\phi \wedge \bigwedge_i P_i(x_i)) \wedge \neg\exists\overline{x}(\psi \wedge \bigwedge_i P_i(x_i))$$

is not satisfiable (on finite structures). Hence, all results we prove for sentences (e.g., the complexity of satisfiability, the finite model property, etc.) all apply to entailment as well.

Below, we state and prove our results for arbitrary structures, but the analogous results for finite structures follow by Theorem 2 and Remark 3.

▶ **Theorem 6.** UNFO^k *has Craig interpolation: for all* $k \geq 1$ *and for every pair of* UNFO^k*-formulas* $\phi(\overline{x}), \psi(\overline{x})$ *in the same free variables such that* $\phi \models \psi$*, there is a* UNFO^k*-formula* $\chi(\overline{x})$ *over the common vocabulary of* ϕ *and* ψ *such that* $\phi \models \chi$ *and* $\chi \models \psi$*.*

As usual, this Craig interpolation theorem implies a Beth definability theorem. Let Σ be a UNFO-theory in a signature σ and let $R \in \sigma$ and $\tau \subseteq \sigma$. We say that Σ *implicitly defines* R in terms of τ if for all τ-structures M and for all σ-expansions M_1, M_2 of M satisfying Σ, we have that $R^{M_1} = R^{M_2}$. We say that a formula $\phi(\overline{x})$ in signature τ is an *explicit definition* of R relative to Σ if $\Sigma \models \forall\overline{x}\ \ (R\overline{x} \leftrightarrow \phi(\overline{x}))$. Note that the formula $\forall\overline{x}\ \ (R\overline{x} \leftrightarrow \phi(\overline{x}))$ is itself not necessarily a UNFO-formula, but this is irrelevant.

▶ **Theorem 7.** UNFO *has the Projective Beth property: whenever a* UNFO*-theory* Σ *in a signature* σ *implicitly defines a* k*-relation* R *in terms of a signature* $\tau \subseteq \sigma$*, then there is a* UNFO*-formula in signature* τ *that is an explicit definition of* R *relative to* Σ*. Moreover, if* Σ *belongs to* UNFO^k *(*$k \geq 1$*), then the explicit definition can be found in* UNFO^k *as well.*

4 Satisfiability

In this section, we show that the satisfiability problem for UNFP and for UNFO is 2ExpTime-complete, both on arbitrary structures and on finite structures. The lower-bound holds already for UNFO^3 over finite trees. Note that this is in contrast with GFO whose complexity drops from 2ExpTime-complete to ExpTime-complete when the arity of relations is bounded [15]. The upper-bound is obtained by a reduction to the two-way modal μ-calculus, whose

satisfiability and finite satisfiability problems are known to be ExpTime-complete [8]. Given a formula φ of UNFP we construct in exponential time a formula φ^* in the μ-calculus such that φ has a (finite) model iff φ^* has a (finite) model. The construction of a finite model of φ from a finite model of φ^* uses a result from [21], which implies that we can restrict attention to locally acyclic structures (i.e., structures that do not contain short cycles). The same reduction to the two-way modal μ-calculus allows us to prove the finite model property of UNFO and the tree-like model property of UNFP.

We describe the reduction from φ to φ^* in two parts. In the first one we consider only a special case of UNFP formulas that we call *simple* where, intuitively, each conjunctive query inside φ is a single atomic formula. The construction of φ^* is then polynomial. In a second part we show how the general case reduces to this one (with an exponential blow-up).

4.1 Simple UNFP formulas

We first consider a fragment of UNFP, which we call *simple* UNFP (sUNFP). It is a common fragment of UNFP and GFP, which embeds the two-way μ-calculus. The syntax of sUNFP is given by the following grammar (recall that we use the notation $\phi(x)$ to indicate that a formula has no first-order free variables besides possibly x, but may contain some monadic second order free variables):

$$\begin{aligned}\phi(x) ::= \quad & P(x) \mid X(x) \mid \phi(x) \wedge \phi(x) \mid \phi(x) \vee \phi(x) \mid \neg\phi(x) \mid [\mathrm{LFP}_{X,y}\phi(y)](x) \mid \\ & \exists y_1 \dots y_n (R(y_1 \dots y_n) \wedge y_i = x \wedge \bigwedge_{j \in \{1\dots n\}\setminus\{i\}} \phi_j(y_j)) \mid \exists x \phi(x)\end{aligned}$$

Note that all formulas generated by this inductive definition have at most one free variable. We denote by sUNFO the first-order (fixed point free) fragment of sUNFP.

We need the following notions. The *incidence graph* $inc(M)$ of a structure M is the bi-partite graph containing facts of M and elements of M, and with an edge between a fact and an element if the element occurs in the fact. We say that a structure M is l-acyclic, for $l \geq 1$, if $inc(M)$ has no cycle of length less than $2l$, and no element of M occurs twice in the same fact. We call a structure acyclic, if it is l-acyclic for all l (i.e., the incidence graph is acyclic and no element occurs in the same fact twice).

A simple formula is essentially a formula of the two-way μ-calculus with navigation through arbitrary relations instead of just binary relations. Based on a simple coding of relations of arbitrary arity using binary relations we can transform a simple formula into a formula of the μ-calculus and obtain:

▶ **Proposition 3.**

1. *The satisfiability problem for* sUNFP *is* ExpTime*-complete, both on arbitrary structures and on finite structures.*
2. *If a* sUNFP *formula has a model, it has an acyclic model*
3. *If a* sUNFP *formula has a finite model, then it has a l-acyclic finite model, $\forall l \geq 1$.*
4. sUNFO *has the finite model property.*

4.2 Arbitrary UNFP-formulas

A formula of UNFO is said to be in *disjunctive* UN-normal form if it is a disjunction of formulas that are in UN-normal form and in which only unary disjunction, of the form $\phi(x) \vee \psi(x)$, is used. It is immediate to see that every formula of UNFO can be turned into

an equivalent (but possibly exponentially longer) one in disjunctive UN-normal form. It turns out that the parameters that will occur in the exponent of the reduction described below (for instance the width) are not affected when going from an arbitrary formula to one in disjunctive normal form. Hence we can now fix an arbitrary UNFP-formula ϕ in disjunctive UN-normal form without loss of generality.

Step 1: Simplifying assumptions (without loss of generality)

1. ϕ is a sentence (all free variables can be existentially quantified, cf. also Remark 3).
2. Every subformula of the form $\exists \overline{z} \psi(y, \overline{z})$ with $\overline{z} = z_1 \dots z_n$ is more precisely of the form

$$\exists \overline{z}(\tau(\overline{z}) \wedge z_i = y \wedge \bigwedge_{j \in \{1 \dots n\} \setminus \{i\}} \phi_j(z_j)) \qquad \text{or} \qquad \exists \overline{z}(\tau(\overline{z}) \wedge \bigwedge_{j \in \{1 \dots n\}} \phi_j(z_j))$$

for some $i \leq n$, where $\tau(\overline{z})$ is a conjunction of relational atomic formulas (no equalities, those can be eliminated by identifying the respective quantified variables).

Step 2: Collecting subformulas and neighborhood types We denote by SUBF_ϕ the set of all subformulas $\psi(y)$ of ϕ that have one free first-order variable. Next, we collect all conjunctive queries occurring in ϕ, viewing each as describing a neighborhood type. For any subformula of ϕ of the form

$$\exists \overline{z}(\tau(\overline{z}) \wedge z_i = y \wedge \bigwedge_{j \in \{1 \dots n\} \setminus \{i\}} \phi_j(z_j)) \qquad \text{or} \qquad \exists \overline{z}(\tau(\overline{z}) \wedge \bigwedge_{j \in \{1 \dots n\}} \phi_j(z_j)) \ ,$$

we call $\tau(\overline{z})$ a *neighborhood type*. We denote the set of neighborhood types in ϕ by NTYPES_ϕ.

Step 3: Stitching neighborhood types together. We now consider structures that are "stitched together" from copies of the neighborhood types in NTYPES_ϕ. To make this precise, we introduce for each neighborhood type $\tau(z_1, \dots, z_n) \in \text{NTYPES}_\phi$ an n-ary relation symbol R_τ. A structure in the signature consisting of these new relations will be called a *stitch diagram*. Each stitch diagram M gives rise to a *stitching* $M^\times$, which is the structure (over the same domain of M) obtained by replacing each R_τ-fact with a copy of the neighborhood type τ, for all $\tau \in \text{NTYPES}_\phi$. Notice that, when going from M to $M^\times$, the distance between nodes can only increase (by second simplifying assumption, τ does not contain equalities hence no nodes are merged during the process).

At this point, our basic strategy for reducing UNFP to sUNFP should be clear: we will produce an sUNFP-formula to describe stitch diagrams whose stitchings satisfy a certain desired UNFP formula. In the rest of this section, we work out the details of this strategy.

It is important to realize that, even if a stitch diagram M does not contain an atomic fact $R_\tau(\overline{a})$, it may still be the case that $M^\times \models \tau(\overline{a})$. In this case we say that the fact $R_\tau(\overline{a})$ is *implicit* in M. For example, this could happen if $M \models R_{\tau'}(\overline{a})$ and τ is homomorphically contained in τ'. The following claim gives us a handle on when this phenomenon may occur. For any $\tau \in \text{NTYPES}_\phi$, we denote by $|\tau|$ the number of atomic formulas in τ. We write $N \subseteq M$ if N is a not-necessarily-induced substructure of M.

▶ **Claim 1.** *If $R_\tau(\overline{a})$ is implicit in a stitch diagram M then there is an $N \subseteq M$ containing at most $|\tau|$ many facts, such that $R_\tau(\overline{a})$ is already implicit in N. Moreover N is connected whenever τ is.*

Proof. We need at most one fact of M to account for each atom in $\tau(\overline{a})$. ◀

Let $l = \max_{\tau \in \text{NTYPES}_\phi} |\tau|$. We will restrict attention to stitch diagrams M that are l-acyclic. By Item (3) of Proposition 3 this is without loss of generality. This implies that every $N \subseteq M$ containing at most l facts is fully acyclic. The importance of the above claim, then, shows in two facts: (i) intuitively, there are finitely many reasons why a fact may be implicit in M, and (ii) each of these reasons is acyclic, and hence can be described in sUNFP as we will see.

Step 4: The translation from UNFP to sUNFP Let $\psi(y)$ be any subformula of ϕ with at most one free variable. By induction on the structure of $\psi(y)$ we construct a sUNFP formula $\psi'(y)$ such that, assuming $\overline{X}$ are the free monadic variables of ψ, for all l-acyclic M, all $a \in M$, and all sets $\overline{S}$ of elements of M, $M, \overline{S} \models \psi'(a)$ iff $M^{\mathsf{x}}, \overline{S} \models \psi(a)$.

The inductive definition commutes with all Boolean operator and with the LFP operator. Fix now any $\tau(\overline{z}) \in \text{NTYPES}_\phi$ with $\overline{z} = z_1, \ldots, z_n$, fix an $i \leq n$, and fix a sequence of formulas $\psi_1, \ldots, \psi_{i-1}, \psi_{i+1}, \ldots, \psi_n \in \text{SUBF}_\phi$ and assume ϕ is of the form:

$$\psi(y) \;:=\; \exists \overline{z}(\tau \wedge z_i = y \wedge \bigwedge_{j \in \{1 \ldots n\} \setminus \{i\}} \psi_j(z_j)) \; .$$

(the argument if ψ is of the form $\exists \overline{z}(\tau \wedge \bigwedge_{j \in \{i \ldots n} \psi_j(z_j))$ is similar. Note that these two cases also account for the base of the induction, if we let $|\overline{z}| = 1$).

By induction we already have constructed sUNFP formulas $\psi'_1, \ldots, \psi'_{i-1}, \psi'_{i+1}, \ldots, \psi'_n$ corresponding to $\psi_1, \ldots, \psi_{i-1}, \psi_{i+1}, \ldots, \psi_n$.

We are interested in detecting in M how a node in M^{x} may come to satisfy ψ. We will construct a sUNFP formula that lists all the cases in M that make this happen. It clearly suffices to consider each connected component of τ at a time. Hence by Claim 1 it only depends on a small neighborhood of x in M. The formula will then be essentially a long disjunction, where each disjunct corresponds to the description of a small neighborhood of M in which τ is implicitly satisfied by a tuple of nodes satisfying in addition the formulas ψ_j. Note that since we assume M to be l-acyclic, these small substructures are all acyclic, which will make it possible to describe them by an (existential) formula of sUNFP.

More precisely, consider any connected acyclic stitch diagram N containing at most l facts, and any homomorphism $h : \tau \to N^{\mathsf{x}}$. We now construct an sUNFP formula $\chi_{\psi,N,h}(y)$ that describes N (existentially positively) from the point of view of $h(z_i)$, and expressing also that each $h(z_j)$ satisfies ψ_j. In other words, we view N as a tree rooted in $h(z_i)$ and the formula describes that tree from top to bottom. We construct the desired sUNFP formula by induction on $|N|$.

Assume N is a tree whose root element is $N_0 = R_\tau(a_1, \cdots, a_n)$ and with several subtrees $N_1, N_2 \cdots$ (the base case where N is a single node is treated similarly). Notice that by l-acyclicity it follows that for all $m > 0$, N_0 and N_m share at most one element, say a_{β_m}. For $m \geq 0$, let h_m be the restriction of h to N_m. Finally assume that h_0 maps z_i to a_{α_i}. The desired formula $\chi_{\phi,N,h}(y)$ is then:

$$\exists z_1 \cdots z_{\alpha_i - 1} z_{\alpha_i + 1} \cdots z_n \Big(R_\tau(z_1 \cdots z_{\alpha_i - 1} y z_{\alpha_i + 1} \cdots z_n) \wedge \bigwedge_{j \neq i, h_0(z_j) = a_{\alpha_j}} \psi'_j(z_{\alpha_j}) \wedge \bigwedge_m \chi_{\psi,N_m,h_m}(z_{\beta_m}) \Big)$$

Finally $\psi'(y)$ is the disjunction, for each N and h as above, of the formulas $\chi_{\psi,N,h}(y)$.

It follows from the construction that, for all l-acyclic stitch diagrams M, $M \models \psi'$ if and only if $M^{\mathsf{x}} \models \psi$. This shows that, if ψ' is satisfiable, then so is ψ. Conversely, it is easy to construct, from a model M of ψ, a model M' of ψ' (indeed, it suffices to take the domain M and to populate the relations R_τ with all tuples satisfying τ in M. Therefore, ψ is satisfiability (on finite structures) if and only if ψ' is.

A careful analysis of the complexity of the above translation yields:

▶ **Theorem 8.** *The satisfiability problem for* UNFP *is in* 2ExpTime*, both on arbitrary structures and on finite structures.*

Notice now that when starting with a formula of UNFO we obtain a formula of sUNFO. In view of Item (4) of Proposition 3, this immediately implies the finite model property of UNFO and proves Theorem 2. Similarly, it follows from Item (2) of Proposition 3 that UNFP has the tree-like model property. Indeed if a stitch diagram M is acyclic, then $M^\times$ has tree-width at most $k-1$, where k is the width of the formula. This proves Theorem 3.

The complexity result of Theorem 8 is tight:

▶ **Proposition 4.** *There is a fixed finite signature such that the satisfiability problem for* UNFO *is* 2ExpTime*-hard, both on arbitrary structures and on finite structures.*

The lower bound can be shown on arbitrary structures, on finite trees, and on any class in-between. The proof uses formulas that have width 3 and negation depth 2. For width 2 we can actually show that satisfiability of UNFO^2 is NExpTime-complete. For negation depth 1 it turns out to be NP^{NP}-complete.

5 Model Checking

In this section we study the model checking complexity of UNFO and UNFP. We are concerned here with the *combined complexity* of the model checking problem, where the input consists of a formula and a structure. It was already observed in [10] that the model checking problem for UNFO is in P^{NP}. Here, we show that the problem is in fact $\text{P}^{\text{NP}[O(\log^2 n)]}$-complete, and that the model checking problem for UNFP is in $\text{NP}^{\text{NP}} \cap \text{coNP}^{\text{NP}}$. The complexity class P^{NP} consists of all problems that are computable by a Turing machine running in time polynomial in the size of its input, where the Turing machine, at any point during its computation, can ask yes/no queries to an NP oracle, and take the answers of the oracle into account in subsequent steps of the computation (including subsequent queries to the NP oracle). Analogously we can define the complexity classes NP^{NP} and coNP^{NP}. The complexity class $\text{P}^{\text{NP}[O(\log^2 n)]}$ is defined in the same way as P^{NP}, except that the number of yes/no queries that can be asked to the NP oracle is bounded by $O(\log^2(n))$, where n is the size of the input. Similarly $\text{P}^{\text{NP}[O(\log n)]}$ restricts the number of yes/no queries to $O(\log(n))$. We refer to [9, 14, 23, 25] for the precise definitions and proprieties of these oracle complexity classes.

▶ **Theorem 9.** *The model checking problem for* UNFO *is* $\text{P}^{\text{NP}[O(\log^2 n)]}$*-complete.*

The lower bound follows from the fact that UNFO embeds $\text{CTL}^*(\mathsf{X})$, since the model checking problem for $\text{CTL}^*(\mathsf{X})$ is already known to be $\text{P}^{\text{NP}[O(\log^2 n)]}$-complete [23]. The upper-bound argument is more involved. It uses $1 \times M$ *TB-trees* [23], a model for $\text{P}^{\text{NP}[O(\log^2 n)]}$ computations based on tree-shaped circuits containing SAT tests. The argument relies on the fact that UNFO formulas can be viewed as such tree-shaped circuits, and it breaks down if subformula sharing is allowed. Indeed for a simple extension of UNFO with a let construct of the form $\mathsf{let}\ b = \phi\ \mathsf{in}\ \psi$ the model checking problem can be shown to be complete for P^{NP}.[2]

Recall that the *negation depth* of a UNFO formula is the maximal nesting depth of negations in its syntactic tree. We can show the following result:

[2] Here, b is a Boolean variable and ϕ a sentence; $\mathsf{let}\ b = \phi\ \mathsf{in}\ \psi$ can be viewed as a succinct notation for the formula obtained by replacing all free occurrences of b in ψ by ϕ.

▶ **Theorem 10.** *For any $l > 0$, the complexity of the model checking problem for* UNFO *formula of negation depth $\leq l$ is $\mathrm{P}^{\mathrm{NP}[O(\log n)]}$-complete. The lower bound holds even when the structure is fixed.*

The upper bound is obtained by induction on l using a naive bottom-up evaluation algorithm. Each level requires one series of parallel calls to the NP-oracle, hence the result when l is a constant. The lower-bound proof uses the fact that every problem in $\mathrm{P}^{\mathrm{NP}[O(\log n)]}$ admits a PTime truth-table reduction to a problem in NP [9]. The proof will be detailed in the journal version of this paper.

Finally, we turn to the complexity of the model checking of UNFP.

▶ **Theorem 11.** *The* UNFP *model checking problem is in $\mathrm{NP}^{\mathrm{NP}} \cap \mathrm{coNP}^{\mathrm{NP}}$ and P^{NP}-hard.*

The upper bound is proved using the obvious extension of the known NP∩coNP algorithm for the model checking of the μ-calculus, using the NP-oracle for solving the unary conjunctive queries. The lower bound will be detailed in the journal version of this paper.

6 Discussion

Trees Over trees, UNFO and UNFP correspond to well known formalisms. We have already mentioned that on XML trees UNFO captures Core XPath and it is not hard to see that UNFP captures the regular languages. When only the child relation of the tree is present, definability in UNFO correspond to "Locally Testability" while UNFP defines the bisimulation invariant regular languages.

Undecidable extensions Our results show that UNFO and UNFP are well behaved logics. One may ask if there are extensions that are still well behaved. Inequalities are a minimal form of negation not supported by UNFO. Unfortunately, extending UNFO with inequalities leads to undecidability. Let us denote by $\mathrm{UNFO}^{\neq}$ the extension of UNFO with inequalities, and with $\mathrm{UNFO}^{\neg}$ the extension of UNFO with negative relational atomic formulas. Recall that a fragment of first-order logic is called a *conservative reduction class* if there a computable map from arbitrary first-order formulas to formulas in the fragment, which preserves (un)satisfiability as well as finite (un)satisfiability.

▶ **Theorem 12.** $\mathrm{UNFO}^{\neq}$ *and* $\mathrm{UNFO}^{\neg}$ *are conservative reduction classes, and hence undecidable for satisfiability on arbitrary structures and on finite structures.*

Also, in the fixed point case, one can wonder whether the restriction to *monadic* least fixed-points was necessary. Indeed, this question naturally arises since it is known that the *guarded fragment* of first-order logic is decidable even when extended with (guarded) fixed point operators of arbitrary arity. However, it turns out that in our setting allowing non-monadic fixed points operators makes the logic undecidable.

▶ **Theorem 13.** *The extension of* UNFP *with non-monadic fixed point operators is undecidable for satisfiability on arbitrary structures and on finite structures.*

Further work We have already mentioned that UNFO and GFO are incomparable. It would be nice to come up with a logic that generalizes both UNFO and GFO. A step in this direction is the recent work of [5] showing that (finite) satisfiability of a Boolean combination of guarded and CQ formulas is decidable. With Vince Bárány we are currently investigating the *guarded negation fragment* of FO which allows negations of the form $R(\overline{x}) \wedge \neg\phi(\overline{x})$. This fragment, and its fixed point extension, generalize both UNFO and GFO, while apparently retaining their good properties, including robust decidability.

References

1 H. Andréka, J. van Benthem, and I. Németi. Modal languages and bounded fragments of predicate logic. *Journal of Philosophical Logic*, 27:217–274, 1998.

2 F. Baader, D. Calvanese, D.L. McGuinness, D. Nardi, and P.F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.

3 J. Bailey, G. Dong, and A.W. To. Logical queries over views: Decidability and expressiveness. *Transactions on Computational Logic*, 11(2):8, 2010.

4 V. Bárány and M. Bojańczyk. Personal communication.

5 V. Bárány, G. Gottlob, and M. Otto. Querying the guarded fragment. In *Proc. of Logic in Computer Science (LICS)*, 2010.

6 J. van Benthem. *Modal Logic and Classical Logic*. Bibliopolis, 1983.

7 D. Berwanger and E. Grädel. Games and model checking for guarded logics. In *LPAR*, pages 70–84, 2001.

8 M. Bojańczyk. Two-way alternating automata and finite models. In *Intl. Coll. on Automata, Languages and Programming (ICALP)*, pages 833–844, 2002.

9 S.R. Buss and H. Louise. On truth-table reducibility to sat. *Inf. Comput.*, 91(1):86–102, 1991.

10 B. ten Cate and M. Marx. Navigational XPath: calculus and algebra. *SIGMOD Record*, 36(2):19–26, 2007.

11 E.M. Clarke, O. Grumberg, and D.A. Peled. *Model checking*. MIT Press, 1999.

12 S. Cosmadakis, H. Gaifman, P. Kanellakis, and M. Vardi. Decidable optimization problems for database logic programs. In *Proc. of ACM Symposium on Theory of Computing (STOC)*, 1988.

13 C. David. *Analyse de XML avec données non-bornées*. PhD thesis, Université Paris Diderot - Paris 7 (LIAFA), 2009.

14 G. Gottlob. NP trees and Carnap's modal logic. *Journal of the ACM*, 42(2):421–457, 1995.

15 E. Grädel. Why are modal logics so robustly decidable? In *Current Trends in Theoretical Computer Science*, pages 393–408. 2001.

16 E. Grädel and I. Walukiewicz. Guarded fixed point logic. In *LICS*, pages 45–54, 1999.

17 W. Hodges. *Model Theory*. Cambridge University Press, 1993.

18 E. Hoogland and M. Marx. Interpolation and definability in guarded fragments. *Studia Logica*, 70(3):373–409, 2002.

19 L. Libkin. *Elements of Finite Model Theory*. Springer, 2004.

20 M. Marx and M. de Rijke. Semantic characterizations of navigational XPath. *SIGMOD Record*, 34(2):41–46, 2005.

21 M. Otto. Bisimulation invariance and finite models. In *Lecture Notes in Logic, Logic Colloquium 2002*, pages 276–298. 2006.

22 E. Rosen. Modal logic over finite structures. *Journal of Logic, Language, and Computation*, 6(4):427–439, 1997.

23 P. Schnoebelen. Oracle circuits for branching-time model checking. In *International Conference on Automata, Languages and Programming*, pages 187–187, 2003.

24 M.Y. Vardi. Why is modal logic so robustly decidable? In *Descriptive Complexity and Finite Models*, pages 149–184, 1996.

25 K.W. Wagner. More Complicated Questions about Maxima and Minima, and some Closures of NP. *Theoretical Computer Science*, 51(1-2):53 – 80, 1987.