Finite Satisfiability of Unary Negation Fragment with Transitivity

Daniel Danielski
University of Wrocław, Poland

Emanuel Kieroński
University of Wrocław, Poland
kiero@cs.uni.wroc.pl

Abstract

We show that the finite satisfiability problem for the unary negation fragment with an arbitrary number of transitive relations is decidable and 2-ExpTime-complete. Our result actually holds for a more general setting in which one can require that some binary symbols are interpreted as arbitrary transitive relations, some as partial orders and some as equivalences. We also consider finite satisfiability of various extensions of our primary logic, in particular capturing the concepts of nominals and role hierarchies known from description logic. As the unary negation fragment can express unions of conjunctive queries, our results have interesting implications for the problem of finite query answering, both in the classical scenario and in the description logics setting.

2012 ACM Subject Classification Theory of computation → Logic

Keywords and phrases unary negation fragment, transitivity, finite satisfiability, finite open-world query answering, description logics

Digital Object Identifier 10.4230/LIPIcs.MFCS.2019.17

Related Version A full version of this paper is available at `https://arxiv.org/abs/1809.03245`.

Funding Supported by Polish National Science Centre grant No 2016/21/B/ST6/01444.

1 Introduction

Decidable fragments and unary negation. Searching for attractive fragments of first-order logic is an important theme in theoretical computer science. Successful examples of such fragments, with numerous applications, are modal and description logics. They have their own syntax, but naturally translate to first-order logic, via the *standard translation.* Several seminal decidable fragments of first-order logic were identified by preserving one particular restriction obeyed by this translation and dropping all the others. Important examples of such fragments are two-variable logic, FO^2, [25], the guarded fragment, GF, [2], and the fluted fragment, FF, [24, 22]. They restrict, respectively, the number of variables, the quantification pattern and the order of variables in which they appear as arguments of predicates. A more recent proposal [27] is the *unary negation fragment*, UNFO. This time we restrict the use of negations, allowing them only in front of subformulas with at most one free variable. UNFO turns out to retain many good algorithmic and model theoretic properties of modal logic, including the finite model property, a tree-like model property and the decidability of the satisfiability problem. We remark here that UNFO and GF have a common decidable generalization, *the guarded negation fragment*, GNFO, [5].

To justify the attractiveness of UNFO let us look at one of the crucial problems in database theory, open-world query answering. Given an (incomplete) set of facts $\mathfrak{D}$, a set of constraints $\mathcal{T}$ and a query q, check if $\mathfrak{D} \wedge \mathcal{T}$ entails q. Generally, this problem is undecidable, and to make it decidable one needs to restrict the class of queries and constraints. Widely investigated class of queries are (unions of) *conjunctive queries* – (disjunctions of) sentences

44th International Symposium on Mathematical Foundations of Computer Science (MFCS 2019).
Editors: Peter Rossmanith, Pinar Heggernes, and Joost-Pieter Katoen; Article No. 17; pp. 17:1–17:15
LIPICS Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

of the form $\exists \bar{x}\psi(\bar{x})$ where ψ is a conjunction of atoms. An important class of constraints are *tuple generating dependencies*, TGDs, of the form $\forall \bar{x}\bar{y}(\psi(\bar{x},\bar{y}) \to \exists \bar{z}\psi'(\bar{y},\bar{z}))$, where ψ and ψ' are, again, conjunctions of atoms. Conjunctive query answering against arbitrary TGDs is still undecidable (see, e.g., [6]), so TGDs need to be restricted further. Several classes of TGDs making the problem decidable have been proposed. One interesting such class are *frontier-one* TGDs, in which the *frontier* of each dependency, $\bar{y}$, consists just of a single variable [4]. Frontier-one TGDs are a special case of frontier-guarded TGDs [3]. Checking whether $\mathfrak{D}$ and $\mathcal{T}$ entail q boils down to verifying (un)satisfiability of the formula $\mathfrak{D} \wedge \mathcal{T} \wedge \neg q$. It turns out that if $\mathcal{T}$ is a conjunction of frontier-one TGDs and q is a disjunction of conjunctive queries then the resulting formula belongs to UNFO.

Transitivity. A serious weakness of the expressive power of UNFO is that it cannot express transitivity of a binary relation, nor related properties like being an equivalence, a partial order or a linear order. This limitation becomes particularly important when database or knowledge representation applications are considered, as transitivity is a natural property in many real-life situations. Just consider relations like *greater-than* or *part-of*. This weakness is shared by FO^2, GF and FF. Thus, it is natural to think about their extensions, in which some distinguished binary symbols may be explicitly required to be interpreted as transitive relations. It turns out that FO^2, GF and FF do not cope well with transitivity, and the satisfiability problems for the obtained extensions are undecidable [15, 13, 23] (see also [10, 18, 17]). Some positive results were obtained for FO^2, GF and FF only when one transitive relation is available [21, 18, 23] or when some further syntactic restrictions are imposed [26].

UNFO is an exception here, since its satisfiability problem remains decidable in the presence of arbitrarily many transitive relations. This has been explicitly stated in [16], as a corollary from a stronger result that UNFO is decidable when extended by regular path expressions. Independently, the decidability of UNFO with transitivity, UNFO+$\mathcal{S}$, follows from [1], which deals with the decidability of a richer logic, the guarded negation fragment with transitive relations restricted to non-guard positions, which embeds UNFO+$\mathcal{S}$. From both papers the 2-ExpTime-completeness of UNFO+$\mathcal{S}$ can be inferred.

Our main results. A problem related to satisfiability is *finite satisfiability*, in which we ask about the existence of *finite* models. In computer science, the importance of decision procedures for finite satisfiability arises from the fact that most objects about which we may want to reason using logic, e.g., databases, are finite. Thus the ability of solving only general satisfiability may not be fully satisfactory. Both the above-mentioned decidability results implying the decidability of UNFO+$\mathcal{S}$ are obtained by employing tree-like model properties of the logics and then using automata techniques. Since tree-like unravelings of models are infinite, this approach works only for general satisfiability, and gives little insight into the decidability/complexity of finite satisfiability. In this paper we consider the finite satisfiability problem for UNFO+$\mathcal{S}$. Actually, we made a step in this direction already in our previous paper [7] (see [8] for its longer version) where we proved a related result that UNFO with equivalence relations, UNFO+EQ, has the finite model property and thus that its satisfiability and finite satisfiability problems coincide, both being 2-ExpTime-complete. Some ideas developed in [7] are extended and applied also here, even though UNFO+$\mathcal{S}$ does not have the finite model property which becomes evident when looking at the following formula with transitive T, $\forall x \exists y Txy \wedge \forall x \neg Txx$, satisfiable only in infinite models.

Our main contribution is demonstrating the decidability of finite satisfiability for UNFO+$\mathcal{S}$ and establishing its 2-ExpTime-completeness. En route we obtain a triply exponential bound on the size of minimal models of finitely satisfiable UNFO+$\mathcal{S}$ formulas. Actually, our results hold for a more general setting, in which some relations may be required to be interpreted as equivalences, some as partial orders, and some just as arbitrary transitive relations. Returning to database motivations, we get this way the decidability of the *finite* open-world query answering for unions of conjunctive queries against frontier-one TGDs with equivalences, partial orders and arbitrary transitive relations. By *finite open-world query answering* we mean the question if for given $\mathfrak{D}$, $\mathcal{T}$ and q, $\mathfrak{D}$ and $\mathcal{T}$ entail q over *finite* structures.

To the best of our knowledge, UNFO+$\mathcal{S}$ is the first logic which allows one to use arbitrarily many transitive relations, and, at the same time, to speak non-trivially about relations of arbitrary arities, whose finite satisfiability problem is shown decidable. In the case of related logics of this kind, like the guarded fragment with transitive guards [26], and the guarded negation fragment with transitive relations outside guards [1], the decidability was shown only for general satisfiability, and its finite version is open. (Finite satisfiability was shown decidable only for the *two-variable* guarded fragment with transitive guards [20]).

We believe that moving from UNFO+EQ from [7] to UNFO+$\mathcal{S}$ is an important improvement. Besides the fact that this requires strengthening our techniques and employing some new ideas, general transitive relations have stronger motivations than equivalences. In particular, it opens natural connections to the realm of description logics, DLs.

UNFO and expressive description logics. UNFO, via the above-mentioned standard translation, embeds the DL $\mathcal{ALC}$, as well as its extension by inverse roles ($\mathcal{I}$) and role intersections ($\sqcap$). Thus, having the ability of expressing conjunctive queries, we can use our results to solve the so-called *(finite) ontology mediated query answering* problem, (F)OMQA, for some DLs. This problem is a counterpart of (finite) open-world query answering: given a conjunctive query (or a union of conjunctive queries) and a knowledge base specified in a DL, check whether the query holds in every (finite) model of this knowledge base.

While there are quite a lot of results for OMQA, not much is known about FOMQA. In particular, for DLs with transitive roles ($\mathcal{S}$) the only positive results we are aware of are the ones obtained recently in [12], where the decidability and 2-ExpTime-completeness of FOMQA for the logics $\mathcal{SOI}$, $\mathcal{SIF}$ and $\mathcal{SOF}$ is shown. This is orthogonal to our results described above, since UNFO+$\mathcal{S}$ captures neither nominals ($\mathcal{O}$) nor functional roles ($\mathcal{F}$). On the other hand, we are able to express any positive boolean combinations of roles, including their intersection ($\sqcap$), which allows us to solve FOMQA, e.g., for the logic $\mathcal{SI}^{\sqcap}$. Moreover we can use non-trivially relations of arity greater than two.

It is an interesting question if our decidability result can be extended to capture some more expressive DLs. Unfortunately, we cannot hope for *number restrictions* ($\mathcal{Q}$ or $\mathcal{N}$) or even *functional roles* ($\mathcal{F}$), as satisfiability and finite satisfiability of UNFO (even without transitive relations) and two binary functional relations are undecidable. This is implicit in [27] (see the full version of this paper for an explicit proof). On the positive side, we show the decidability and 2-ExpTime-completeness of finite satisfiability of UNFO+$\mathcal{SOH}$, extending UNFO+$\mathcal{S}$ by constants (corresponding to *nominals* ($\mathcal{O}$)) and inclusions of binary relations (capturing *role hierarchies* ($\mathcal{H}$)). This is sufficient, in particular, to imply the decidability of FOMQA for the description logic $\mathcal{SHOI}^{\sqcap}$, which, up to our knowledge, is a new result.

Towards guarded negation fragment. We propose also another decidable extension of our basic logic, the *one-dimensional base-guarded negation fragment* with transitive relations on non-guard positions, BGNFO$_1$+$\mathcal{S}$. This is a non-trivial fragment of the already mentioned logic from [1]. After some rather easy adjustments, our constructions cover this bigger logic, however, it becomes undecidable when extended with inclusions of binary relations.

Organization of the paper. The rest of this paper is organized as follows. Section 2 contains definitions, basic facts and a high-level description of our decidability proof. As our constructions are rather complex, in the main body of the paper, Section 3, we explicitly process the restricted, two-variable case of our logic, for which our ideas can be presented more transparently. In Section 4 we just formulate the remaining results, leaving the details for the full version of this paper, which also contains the missing proofs from Sections 2 and 3. In Section 5 we conclude the paper.

2 Preliminaries

2.1 Logics, structures, types and functions

We employ standard terminology and notation from model theory. We refer to structures using Fraktur capital letters, and their domains using the corresponding Roman capitals. For a structure $\mathfrak{A}$ and $A' \subseteq A$ we use $\mathfrak{A}{\restriction}A'$ or $\mathfrak{A}'$ to denote the restriction of $\mathfrak{A}$ to A'.

The *unary negation fragment of first-order logic*, UNFO is defined by the following grammar [27]: $\varphi = B\bar{x} \mid x = y \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \exists x \varphi \mid \neg\varphi(x)$, where, in the first clause, B represents any relational symbol, and, in the last clause, φ has no free variables besides (at most) x. An example formula not expressible in UNFO is $x \neq y$. We formally do not have universal quantification. However we allow ourselves to use $\forall \bar{x} \neg \varphi$ as an abbreviation for $\neg \exists \bar{x} \varphi$, for an UNFO formula φ. Note that frontier-one TGDs $\forall \bar{x} y (\psi(\bar{x}, y) \rightarrow \exists \bar{z} \psi'(y, \bar{z}))$ are in UNFO as they can be rewritten as $\neg \exists \bar{x} y (\psi(\bar{x}, y) \wedge \neg \exists \bar{z} \psi'(y, \bar{z}))$.

We mostly work with purely relational signatures (admitting constants only in some extensions of our main results) of the form $\sigma = \sigma_{\text{base}} \cup \sigma_{\text{dist}}$, where σ_{base} is the *base signature*, and σ_{dist} is the *distinguished signature*. We assume that $\sigma_{\text{dist}} = \{T_1, \ldots, T_{2k}\}$, with all the T_u binary, and intension that T_{2u} is interpreted as the inverse of T_{2u-1}. For every $1 \leq u \leq k$ we sometimes write T_{2u}^{-1} for T_{2u-1}, and T_{2u-1}^{-1} for T_{2u}. We say that a subset $\mathcal{E}$ of σ_{dist} is *closed under inverses* if, for every $1 \leq u \leq 2k$, we have $T_u \in \mathcal{E}$ iff $T_u^{-1} \in \mathcal{E}$. Note that $\mathcal{E}$ is closed under inverses iff $\sigma_{\text{dist}} \setminus \mathcal{E}$ is closed under inverses. Given a formula φ we denote by σ_φ the signature induced by φ, *i.e.*, the minimal signature, with its distinguished part closed under inverses, containing all symbols from φ.

The *unary negation fragment with transitive relations*, UNFO+$\mathcal{S}$, is defined by the same grammar as UNFO, however when satisfiability of its formulas is considered, we restrict the class of admissible models to those that interpret all symbols from σ_{dist} as transitive relations and, additionally, for each u, interpret T_{2u} as the inverse of T_{2u-1}. The latter condition is intended to simplify the presentation, and is imposed without loss of generality. In our constructions we sometimes consider some auxiliary structures in which symbols from σ_{dist} are not necessarily interpreted as transitive relations (but the pairs T_{2u-1}, T_{2u} are always interpreted as inverses of each other).

An *(atomic) k-type* over a signature σ is a maximal satisfiable set of literals (atoms and negated atoms) over σ with variables $x_1, \ldots, x_k$. We often identify a k-type with the conjunction of its elements. We are mostly interested in 1- and 2-types. Given a σ-structure $\mathfrak{A}$ and $a, b \in A$ we denote by $\mathrm{atp}^{\mathfrak{A}}(a)$ the 1-type *realized* by a, that is the unique 1-type $\alpha(x_1)$ such that $\mathfrak{A} \models \alpha(a)$, and by $\mathrm{atp}^{\mathfrak{A}}(a, b)$ the unique 2-type $\beta(x_1, x_2)$ such that $\mathfrak{A} \models \beta(a, b)$.

We use various functions in our paper. Given a function $f : A \rightarrow B$ we denote by $\mathrm{Rng} f$ its range, by $\mathrm{Dom} f$ its domain, and by $f{\restriction}A_0$ the restriction of f to $A_0 \subseteq A$.

2.2 Normal form, witnesses and basic facts

We say that an UNFO+$\mathcal{S}$ formula is in Scott-normal form if it is of the shape

$$\forall x_1, \ldots, x_t \neg\varphi_0(\bar{x}) \wedge \bigwedge_{i=1}^{m} \forall x \exists \bar{y} \varphi_i(x, \bar{y}) \tag{1}$$

where each φ_i is a UNFO+$\mathcal{S}$ quantifier-free formula and φ_0 is additionally in negation normal form (NNF). A similar normal form for UNFO was introduced in the bachelor's thesis [9]. By a straightforward adaptation of Scott's translation for FO^2 [25] one can translate in polynomial time any UNFO+$\mathcal{S}$ formula to a formula in normal form, in such a way that both are satisfiable over the same domains. This allows us, when dealing with decidability/complexity issues for UNFO+$\mathcal{S}$, or when considering the size of minimal finite models of formulas, to restrict attention to normal form formulas.

Given a structure $\mathfrak{A}$, a normal form formula φ as in (1) and elements $a, \bar{b}$ of A such that $\mathfrak{A} \models \varphi_i(a, \bar{b})$ we say that the elements of $\bar{b}$ are *witnesses* for a and φ_i and that $\mathfrak{A}{\restriction}\{a, \bar{b}\}$ is a *witness structure* for a and φ_i. Fix an element a. For every φ_i choose a witness structure $\mathfrak{W}_i$. Then the structure $\mathfrak{W} = \mathfrak{A}{\restriction}\{W_1 \cup \ldots \cup W_m\}$ is called a φ-*witness structure* for a.

We are going to present a construction which given an arbitrary finite model of a normal form UNFO+$\mathcal{S}$ formula φ builds a finite model of φ of a bounded size. The construction goes via several intermediate steps in which some tree-like models are produced. To argue that that they are still models of φ we use the following basic observation (we recall that t is the number of variables of the $\forall$-conjunct of φ).

▶ **Lemma 1.** *Let $\mathfrak{A}$ be a model of a normal form UNFO+$\mathcal{S}$ formula φ. Let $\mathfrak{A}'$ be a structure in which all symbols from σ_{dist} are interpreted as transitive relations, such that*
(a1) *for every $a' \in A'$ there is a φ-witness structure for a' in $\mathfrak{A}'$,*
(a2) *for every tuple $a'_1, \ldots, a'_t \in A'$ there is a homomorphism $\mathfrak{h} : \mathfrak{A}'{\restriction}\{a'_1, \ldots, a'_t\} \to \mathfrak{A}$ which preserves 1-types of elements.*
Then $\mathfrak{A}' \models \varphi$.

2.3 Plan of the small model construction

Our main goal is to show that finite satisfiability of UNFO+$\mathcal{S}$ formulas can be checked in 2-ExpTime. To this end we will introduce a natural notion of tree-like structures and a measure associating with transitive paths of such structures their so-called *ranks*. Intuitively, for a transitive relation T_i and a T_i-path π, the T_i-rank of π is the number of one-directional T_i-edges in π (a precise definition is given in Section 3.1). Then we show that having the following forms of models is equivalent for a normal form formula φ:
(f1) finite;
(f2) tree-like, with bounded ranks of transitive paths;
(f3) tree-like, with ranks of transitive paths bounded doubly exponentially in $|\varphi|$;
(f4) tree-like, with ranks of paths bounded doubly exponentially in $|\varphi|$, and regular (with doubly exponentially many non-isomorphic subtrees);
(f5) finite of size triply exponential in $|\varphi|$.
We will make the following steps: (f1) $\rightsquigarrow$ (f2), (f2) $\rightsquigarrow$ (f3), (f3) $\rightsquigarrow$ (f4), (f4) $\rightsquigarrow$ (f5). The step closing the circle, (f5) $\rightsquigarrow$ (f1) is trivial. In the two-variable case, we will omit the form (f4) and directly show (f3) $\rightsquigarrow$ (f5). Our 2-ExpTime-algorithm will look for models of the form (f3). Showing transitions leading from (f3) to (f5) justifies that its answers coincide indeed with the existence of *finite* models.

This scheme is similar to the one we used to show the finite model property for UNFO+EQ in [7]. In the main part of the construction from [7] we build bigger and bigger substructures in which some equivalence relations are total. The induction goes, roughly speaking, by the number of non-total equivalences in the substructure. Here we extend this approach to handle one-way transitive connections. It may be useful to briefly compare the case of UNFO+$\mathcal{S}$ and the case of UNFO+EQ.

First of all, if a given formula φ is from UNFO+EQ then we can start our constructions leading to a small finite model of φ from its arbitrary model, while if φ is in UNFO+$\mathcal{S}$ we start from a *finite* model of φ. A very simple step (f1) $\rightsquigarrow$ (f2) in both papers is, essentially, identical. The counterpart of step (f3) $\rightsquigarrow$ (f4) in the case of equivalences is slightly simpler, but the main differences lie in steps (f2) $\rightsquigarrow$ (f3) and (f4) $\rightsquigarrow$ (f5). The former, clearly, is not present at all in [7]. While the general idea in this step is quite standard, as we just use a kind of tree pruning, the details are rather delicate due to possible interactions among different transitive relations, and this step is, by no means, trivial. We refine here, in particular, the apparatus of *declarations* introduced in [7]. Regarding step (f4) $\rightsquigarrow$ (f5), the main construction there, in its single inductive step, has two phases: building the so-called *components* and then arranging them into a bigger structure. It is this first phase which is more complicated than in the corresponding step in [7]. Having components prepared we join them similarly as in [7].

3 The two-variable case

As in the case of unbounded number of variables we can restrict attention to normal form formulas, which in the two-variable case simplify to the standard Scott-normal form [25]:

$$\forall xy \neg\varphi_0(x,y) \wedge \bigwedge_{i=1}^{m} \forall x \exists y \varphi_i(x,y), \tag{2}$$

where all φ_i are quantifier-free UNFO^2+$\mathcal{S}$ formulas (in this restricted case it is not important whether φ_0 is in NNF or not). As is typical for two-variable logics we assume that formulas do not use relational symbols of arity greater than 2 (cf. [14]).

3.1 Tree pruning in the two-variable case

We use a standard notion of a (finite or infinite) rooted tree and related terminology. Additionally, any set consisting of a node and all its children is called a *family*. Any node b, except for the root and the leaves, belongs to two families: the one containing its parent, and the one containing its children, the latter called the *downward family of* b.

We say that a structure $\mathfrak{A}$ over a signature consisting of unary and binary symbols is a *light tree-like structure* if its nodes can be arranged into a rooted tree in such a way that if $\mathfrak{A} \models Baa'$ for some non-transitive relation symbol B then one of three conditions holds: $a = a'$, a is the parent of a' or a is a child of a', and if $\mathfrak{A} \models T_u aa'$ for some T_u then either $a = a'$ or there is a sequence of distinct nodes $a = a_0, a_1, \ldots, a_k = a'$ such that a_i and a_{i+1} are joined by an edge of the tree and $\mathfrak{A} \models T_u a_i a_{i+1}$. In other words, distant nodes in a light tree-like structure can be joined only by transitive connections, moreover, these transitive connections are just the transitive closures of connections inside families. For a light tree-like structure $\mathfrak{A}$ and $a \in A$ we denote by A_a the set of all nodes in the subtree rooted at a and by $\mathfrak{A}_a$ the corresponding substructure.

Let $\mathfrak{A}$ be a light tree-like structure. A sequence of nodes $a_1, \ldots, a_N \in A$ is a *downward path* in $\mathfrak{A}$ if for each i a_{i+1} is a child of a_i. A *downward-T_u-path* is a downward path such that for each i we have $\mathfrak{A} \models T_u a_i a_{i+1}$. The *$T_u$-rank* of a downward-$T_u$-path $\vec{a}$, $\mathfrak{r}_u^{\mathfrak{A}}(\vec{a})$, is the cardinality of the set $\{i : \mathfrak{A} \models \neg T_u a_{i+1} a_i\}$. The T_u-*rank* of an element $a \in A$ is defined as $\mathfrak{r}_u^{\mathfrak{A}}(a) = \sup\{\mathfrak{r}_u^{\mathfrak{A}}(\vec{a}) : \vec{a} = a, a_2, \ldots, a_N; \vec{a}$ is a downward-T_u-path$\}$. For an integer M, we say that $\mathfrak{A}$ has downward-T_u-paths *bounded by* M when for all $a \in A$ we have $\mathfrak{r}_u^{\mathfrak{A}}(a) \leq M$, and that $\mathfrak{A}$ has *transitive paths bounded by* M if it has downward-T_u-paths bounded by M for all u. Note that a downward-T_u-path bounded by M may have more than M nodes, as the symmetric T_u-connections do not increase the rank.

Given an arbitrary model $\mathfrak{A}$ of a normal form UNFO^2+$\mathcal{S}$ formula φ we can simply construct its light tree-like model of degree bounded by $|\varphi|$. We define a *light-φ-tree-like unraveling* $\mathfrak{A}'$ of $\mathfrak{A}$ and an associated function $\mathfrak{h} : A' \to A$ in the following way. $\mathfrak{A}'$ is divided into levels $L_0, L_1, \ldots$. Choose an arbitrary element $a \in A$ and add to level L_0 of A' an element a' such that $\mathrm{atp}^{\mathfrak{A}'}(a') = \mathrm{atp}^{\mathfrak{A}}(a)$; set $\mathfrak{h}(a') = a$. The element a' will be the only element of L_0 and will become the root of $\mathfrak{A}'$. Having defined L_i repeat the following for every $a' \in L_i$. For every j, if $\mathfrak{h}(a')$ is not a witness for φ_j and itself then choose in $\mathfrak{A}$ a witness b for $\mathfrak{h}(a')$ and φ_j. Add a fresh copy b' of b to L_{i+1}, make $\mathfrak{A}' \restriction \{a', b'\}$ isomorphic to $\mathfrak{A} \restriction \{\mathfrak{h}(a'), b\}$ and set $\mathfrak{h}(b') = b$. Complete the definition of $\mathfrak{A}'$ transitively closing all relations from σ_{dist}.

▶ **Lemma 2** ((f1) $\rightsquigarrow$ (f2), light)**.** *Let $\mathfrak{A}$ be a finite model of a normal form UNFO^2+$\mathcal{S}$ formula φ. Let $\mathfrak{A}'$ be a light-φ-tree-like unraveling of $\mathfrak{A}$. Then $\mathfrak{A}' \models \varphi$ and $\mathfrak{A}'$ is a light tree-like structure of degree bounded by $|\varphi|$, and transitive paths bounded by $|A|$.*

Our next task is making the transition (f2) $\rightsquigarrow$ (f3). For this purpose we introduce a notion of light declarations. It is closely related to a notion of declarations which will be used in the general case, but simpler than the latter. Fix a signature and let $\boldsymbol{\alpha}$ be the set of 1-types over this signature.

For $\mathcal{T} \subseteq \{T_1, \ldots, T_{2k}\}$ we write $\mathfrak{A} \models \mathcal{T}ab$ iff $\mathfrak{A} \models T_u ab$ for all $T_u \in \mathcal{T}$. A *light declaration* is a function of type $\mathcal{P}(\{T_1, \ldots, T_{2k}\}) \to \mathcal{P}(\boldsymbol{\alpha})$. Given a light tree-like structure $\mathfrak{A}$ and its node a we say that a *respects* a light declaration $\mathfrak{d}$ if for every $\mathcal{T}$, for every $\alpha \in \mathfrak{d}(\mathcal{T})$ there is **no** node $b \in A$ of 1-type α such that $\mathfrak{A} \models \mathcal{T}ab$. We denote by $\mathrm{ldec}^{\mathfrak{A}}(a)$ the maximal light declaration respected by a. Formally, for every $\mathcal{T} \subseteq \{T_1, \ldots, T_{2k}\}$, $\mathrm{ldec}^{\mathfrak{A}}(a)(\mathcal{T}) = \{\alpha :$ for every node b of type α we have $\neg\mathfrak{A} \models \mathcal{T}ab\}$. Intuitively, $\mathrm{ldec}^{\mathfrak{A}}(a)$ says, for any combination of transitive relations, which 1-types have **no** realizations to which a is connected by this combination in $\mathfrak{A}$. Note that if a respects a light declaration $\mathfrak{d}$ then for any $\mathcal{T}$ we have $\mathfrak{d}(\mathcal{T}) \subseteq \mathrm{ldec}^{\mathfrak{A}}(a)(\mathcal{T})$. We remark that it would be equivalent to define the light declarations without the negations, listing the 1-types that a given node **is** connected with, however we choose a version with negations to make them uniform with the corresponding (more complicated) notion in the general case, where negations are more convenient.

Now we define the *local consistency conditions (LCCs)* for a system of light declarations $(\mathfrak{d}_a)_{a \in A}$ assigned to all nodes of a tree-like structure $\mathfrak{A}$. Let F be the downward family of some node a. We say that the system *satisfies LCCs at a* if for every $a_1, a_2 \in F$ and for every $\mathcal{T}$ such that $\mathfrak{A} \models \mathcal{T}a_1a_2$ the following two conditions hold: **(ld1)** for every $\alpha \in \boldsymbol{\alpha}$, if $\alpha \in \mathfrak{d}_{a_1}(\mathcal{T})$ then $\alpha \in \mathfrak{d}_{a_2}(\mathcal{T})$, **(ld2)** $\mathrm{atp}^{\mathfrak{A}}(a_2) \notin \mathfrak{d}_{a_1}(\mathcal{T})$. Given a light tree-like structure $\mathfrak{A}$ we say that a system of light declarations $(\mathfrak{d}_a)_{a \in A}$ is *locally consistent* if it satisfies LCCs at each $a \in A$ and is *globally consistent* if $\mathfrak{d}_a(\mathcal{T}) \subseteq \mathrm{ldec}^{\mathfrak{A}}(a)(\mathcal{T})$ for each $a \in A$ and each $\mathcal{T}$. Note that the global consistency means that all nodes a respect their light declarations $\mathfrak{d}_a$. It is not difficult to see that local and global consistency play along in the following sense.

▶ **Lemma 3** (Local-global, light)**.** *Let $\mathfrak{A}$ be a light tree-like structure. Then, (i) if a system of light declarations $(\mathfrak{d}_a)_{a \in A}$ is locally consistent then it is globally consistent; and (ii) the* canonical *system of light declarations, $(\mathrm{ldec}^{\mathfrak{A}}(a))_{a \in A}$, is locally consistent.*

Given a light tree-like structure $\mathfrak{A}$, by the *generalized type* of a node a of $\mathfrak{A}$ we will mean a pair $(\mathrm{ldec}^{\mathfrak{A}}(a), \mathrm{atp}^{\mathfrak{A}}(a))$, and denote it as $\mathrm{gtp}^{\mathfrak{A}}(a)$. We introduce a concept of top-down tree pruning. Let $\mathfrak{A}$ be a light tree-like structure. A *top-down tree pruning process* on $\mathfrak{A}$ has countably many steps $0, 1, 2, \ldots$, each of them producing a new light tree-like structure by removing some nodes from the previous one and naturally stitching together the surviving nodes. We emphasise that the universes of all structures build in this process are subsets of the universe of the original structure $\mathfrak{A}$. More specifically, we take $\mathfrak{A}_0 := \mathfrak{A}$, and having constructed $\mathfrak{A}_i$, $i \geq 0$ construct $\mathfrak{A}_{i+1}$ as follows. For every node a of $\mathfrak{A}_i$ of depth $i+1$ (we assume that the root has depth 0) either leave the subtree rooted at a untouched or replace it by a subtree rooted at some descendant b of a having in the original structure $\mathfrak{A}$ the same generalized type as a, and then transitively close all transitive relations. The result of the process is a naturally defined limit structure $\mathfrak{A}'$, in which the pair of elements a, b, of depth d_a and d_b respectively, has its 2-type taken from $\mathfrak{A}_{\max(d_a, d_b)}$. Note that this 2-type is not modified in the subsequent structures, so the definition is sound.

▶ **Lemma 4** (Tree-pruning, light). *Let $\mathfrak{A}$ be a light tree-like structure. Let $(\mathfrak{d}_a)_{a \in A}$ be the canonical system of light declarations on $\mathfrak{A}$, $\mathfrak{d}_a := \mathrm{ldec}^{\mathfrak{A}}(a)$. Let $\mathfrak{A}'$ be the result of a top-town tree pruning process on $\mathfrak{A}$. Then (i) the system of light declarations $(\mathfrak{d}_a)_{a \in A'}$ (the canonical declarations from $\mathfrak{A}$ of the nodes surviving the pruning process) in $\mathfrak{A}'$ is locally consistent, (ii) for any pair of elements $a, a' \in A'$ there is a homomorphism $\mathfrak{A}\restriction\{a, a'\} \to A$ preserving the 1-types; it also follows that (iii) for a normal form φ, if $\mathfrak{A}$ is a model of φ such that any node a has all its witnesses in its downward family then $\mathfrak{A}' \models \varphi$.*

It is not difficult to devise a strategy of top-down tree pruning leading to a model with short transitive paths in a simple scenario where only one transitive relation is present. With several transitive relations, however, a quite intricate strategy seems to be required. The main obstacle is that when decreasing the T_u-rank of an element a, for some u, we may accidentally increase the T_v-rank of a for some $v \neq u$. Nevertheless, an appropriate strategy exists (see the full version of this paper), which allows us to state:

▶ **Lemma 5** ((f2) ⇝ (f3), light). *Let φ be a normal form UNFO^2+$\mathcal{S}$ formula. Let $\mathfrak{A} \models \varphi$ be a light tree-like structure over signature σ_φ, with transitive paths bounded by some natural number M, such that each element has all the required witnesses in its downward family. Then φ has a light tree-like model with transitive paths bounded doubly exponentially in $|\varphi|$.*

3.2 Finite model construction in the two-variable case

In this section we show the following small model property. To this end, in particular, we will make the transition (f3) ⇝ (f5).

▶ **Theorem 6.** *Every finitely satisfiable two-variable UNFO+$\mathcal{S}$ formula φ has a finite model of size bounded triply exponentially in $|\varphi|$.*

Let us fix a finitely satisfiable normal form UNFO+$\mathcal{S}$ formula φ over a signature $\sigma_\varphi = \sigma_{\text{base}} \cup \sigma_{\text{dist}}$ for $\sigma_{\text{dist}} = \{T_1, \ldots, T_{2k}\}$. Denote by $\boldsymbol{\alpha}$ the set of 1-types over this signature. Fix a light tree-like model $\mathfrak{A} \models \varphi$, with linearly bounded degree and doubly exponentially bounded transitive paths (in this section we denote this bound by $\hat{M}_\varphi$), as guaranteed by Lemma 5. We show how to build a "small" finite model $\mathfrak{A}' \models \varphi$. For a set $\mathcal{E} \subseteq \sigma_{\text{dist}}$, closed under inverses, and $a \in A$ we denote by $[a]_{\mathcal{E}}$ the set consisting of a and all elements $b \in A$ such that $\mathfrak{A} \models T_u ab$ for all $T_u \in \mathcal{E}$. Note that $[a]_{\mathcal{E}}$ is either a singleton or each of the $T_u \in \mathcal{E}$ is total on $[a]_{\mathcal{E}}$, that is, for each $b_1, b_2 \in [a]_{\mathcal{E}}$ we have $\mathfrak{A} \models T_u b_1 b_2$ for all $T_u \in \mathcal{E}$. We note that $[a]_\emptyset = A$.

In our construction we inductively produce finite fragments of $\mathfrak{A}'$ corresponding to some (potentially infinite) classes $[a]_\mathcal{E}$ of $\mathfrak{A}$. Essentially, the induction goes downward on the size of $\mathcal{E}$. Intuitively, if a relation is total then it plays no important role, so we may forget about it during the construction. Every such fragment will be obtained by an appropriate arrangement of some number of basic building blocks, called *components*. Each of the components is obtained by some number of applications of the inductive assumption to situations in which a new pair of relations T_{2u-1}, T_{2u} is added to $\mathcal{E}$.

Let us formally state our inductive lemma. In this statement we do not explicitly include any bound on the size of promised finite models, but such a bound will be implicit in the proof and will be presented later. Recall that $\mathfrak{A}$ is the model fixed at the beginning of this subsection.

▶ **Lemma 7** (Main construction, light). *Let $a_0 \in A$ and let $\mathcal{E}_0 \subseteq \sigma_{dist}$ be closed under inverses, let $\mathcal{E}_{tot} := \sigma_{dist} \setminus \mathcal{E}_0$. Let $\mathfrak{A}_0 = \mathfrak{A}_{a_0}\restriction[a_0]_{\mathcal{E}_{tot}}$. Then there exist a finite structure $\mathfrak{A}'_0$, a function $\mathfrak{p} : A'_0 \to A_0$ and an element $a'_0 \in A'_0$, called the* origin *of $\mathfrak{A}'_0$, such that*

(b1) *A'_0 is a singleton or every symbol from $\mathcal{E}_{tot}$ is interpreted as the total relation on $\mathfrak{A}'_0$.*

(b2) *$\mathfrak{p}(a'_0) = a_0$.*

(b3) *For each $a' \in A'_0$ and each i, if $\mathfrak{p}(a')$ has a child being its witness for φ_i in $\mathfrak{A}_0$ then a' has a witness for φ_i in $\mathfrak{A}'_0$. Moreover, $\mathrm{atp}^{\mathfrak{A}'_0}(a') = \mathrm{atp}^{\mathfrak{A}_0}(\mathfrak{p}(a'))$.*

(b4) *For every pair $a', b' \in A'_0$ there exists a homomorphism $\mathfrak{h} : \mathfrak{A}'_0\restriction\{a', b'\} \to \mathfrak{A}$ preserving 1-types such that $\mathfrak{h}(a') = \mathfrak{p}(a')$, and for any 1-type α and $\mathcal{T} \subseteq \{1, \ldots, 2k\}$, if $\mathfrak{A}'_0 \models \mathcal{T}a'b'$ and $\alpha \notin \mathrm{ldec}^{\mathfrak{A}}(\mathfrak{p}(b'))(\mathcal{T})$ then $\alpha \notin \mathrm{ldec}^{\mathfrak{A}}(\mathfrak{p}(a'))(\mathcal{T})$.*

Observe first that Lemma 7 indeed allows us to build a particular finite model of φ. Apply it to $\mathcal{E}_0 = \sigma_{\mathrm{dist}}$ (which means that $\mathcal{E}_{tot} = \emptyset$ and $[a_0]_{\mathcal{E}_{tot}} = A$) and a_0 being the root of $\mathfrak{A}$ (which means that $\mathfrak{A}_0 = \mathfrak{A}$) and use Lemma 1 to see that the obtained structure $\mathfrak{A}'_0$ is a model of φ. Indeed, Condition (a1) of Lemma 1 follows directly from Condition (b3), as in this case $\mathfrak{p}(a')$ has all witnesses in $\mathfrak{A}_0$. Condition (a2) is directly implied by Condition (b4).

The proof of Lemma 7 goes by induction on l, where $l = |\mathcal{E}_0|/2$. In the base of induction, $l = 0$, we have $\mathcal{E}_{tot} = \sigma_{\mathrm{dist}}$. Without loss of generality we may assume that the classes $[a]_{\mathcal{E}_{tot}}$ are singletons for all $a \in A$. (If this is not the case, we just add artificial transitive relations T_{2k+1} and T_{2k+2} both interpreted as the identity in $\mathfrak{A}$.) We simply take $\mathfrak{A}'_0 := \mathfrak{A}_0 = \mathfrak{A}\restriction\{a_0\}$ and set $\mathfrak{p}(a_0) = a_0$. It is readily verified that the conditions (b1)–(b4) are then satisfied.

For the inductive step assume that Lemma 7 holds for arbitrary $\mathcal{E}_0$ closed under inverses, of size $2(l-1) < 2k$. We show that then it holds for $\mathcal{E}_0$ of size $2l$. Take such $\mathcal{E}_0$, and assume, w.l.o.g., that $\mathcal{E}_0 = \{T_1, \ldots, T_{2l}\}$. In the next two subsections we present a construction of $\mathfrak{A}'_0$. We argue that it is correct in the full version of this paper. Finally we estimate the size of the produced models and establish the complexity of the finite satisfiability problem.

3.2.1 Pattern components

We plan to construct $\mathfrak{A}'_0$ out of basic building blocks called *components*. Each component will be an isomorphic copy of some pattern component.

Let $\boldsymbol{\gamma}[A_0]$ be the set of the generalized types realized in $\mathfrak{A}_0$. For every $\gamma \in \boldsymbol{\gamma}[A_0]$ we construct two pattern structures, a *pattern component* $\mathfrak{C}^\gamma$ and an *extended pattern component* $\mathfrak{G}^\gamma$. $\mathfrak{C}^\gamma$ is a finite structure whose universe is divided into $2l$ *layers* $L_1, \ldots, L_{2l}$. $\mathfrak{G}^\gamma$ extends $\mathfrak{C}^\gamma$ by an additional, *interface layer*, denoted L_{2l+1}. See the left part of Fig. 1. We now define $\mathfrak{G}^\gamma$, obtaining then $\mathfrak{C}^\gamma$ just by the restriction of $\mathfrak{G}^\gamma$ to non-interface layers.

Each non-interface layer L_i is further divided into *sublayers* $L_i^1, L_i^2, \ldots, L_i^{\hat{M}_\varphi+1}$. Additionally, in each sublayer L_i^j its initial part $L_i^{j,init}$ is distinguished. In particular, $L_1^{1,init}$ consists

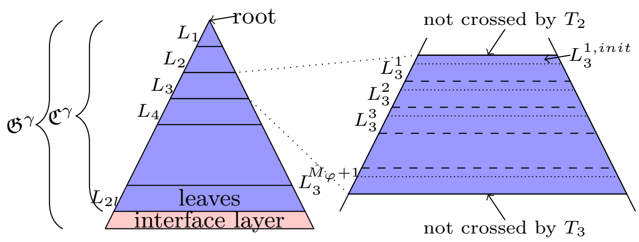

Figure 1 A schematic view of a component in the two-variable case.

of a single element called the *root*. The interface layer L_{2l+1} has no internal division but, for convenience, is sometimes referred to as $L_{2l+1}^{1,init}$. The elements of L_{2l} are called *leaves* and the elements of L_{2l+1} are called *interface elements*. See Fig. 1.

$\mathfrak{G}^\gamma$ will have a shape resembling a tree, with structures obtained by the inductive assumption as nodes, though it will not be tree-like in the sense of Section 3.1 (in particular, the internal structure of nodes may be complicated). All elements of $\mathfrak{G}^\gamma$, except for the interface elements, will have appropriate witnesses (those required by (b3)) provided. The crucial property we want to enforce is that the root of $\mathfrak{G}^\gamma$ will not be joined to its interface elements by any transitive path.

We remark that during the process of building a pattern component we do not yet apply the transitive closure to the distinguished relations. Postponing this step is not important from the point of view of the correctness of the construction, but will allow us for a more precise presentation of the proof of this correctness. Given a component $\mathfrak{C}$ (extended component $\mathfrak{G}$) we will sometimes denote by $\mathfrak{C}_+$ ($\mathfrak{G}_+$) the structure obtained from $\mathfrak{C}$ ($\mathfrak{G}$) by applying all the appropriate transitive closures.

The role of every non-interface layer L_u is, speaking informally, to *kill* T_u, that is to ensure that there will be no T_u-connections from L_u to L_{u+1}. See the right part of Fig. 1. The role of sublayers of L_u, on the other hand, is to decrease the T_u-rank of the patterns of elements. The purpose of the interface layer, L_{2l+1}, will be to connect the component with other components.

If γ is the generalized type of a_0 then take $a := a_0$; otherwise take as a any element of A_0 of generalized type γ. We begin the construction of $\mathfrak{G}^\gamma$ by defining $L_1^{1,init} = \{a'\}$ for a fresh a', setting $\text{atp}^{\mathfrak{G}^\gamma}(a') = \text{atp}^{\mathfrak{A}}(a)$ and $\mathfrak{p}(a') = a$.

Construction of a layer. Let $1 \leq u \leq 2l$. Assume we have defined layers $L_1, \ldots, L_{u-1}$, the initial part of sublayer L_u^1, $L_u^{1,init}$, and both the structure of $\mathfrak{G}^\gamma$ and the values of $\mathfrak{p}$ on $L_1 \cup \ldots \cup L_{u-1} \cup L_u^{1,init}$. We are going to kill T_u. We now expand $L_u^{1,init}$ to a full layer L_u.

Step 1: Subcomponents. Assume that we have defined sublayers $L_u^1, \ldots, L_u^{j,init}$, and both the structure of $\mathfrak{G}^\gamma$ and the values of $\mathfrak{p}$ on $L_1 \cup \ldots \cup L_{u-1} \cup L_u^1 \cup \ldots \cup L_u^{j,init}$. For each $b \in L_u^{j,init}$ perform independently the following procedure. Apply the inductive assumption to $\mathfrak{p}(b)$ and the set $\mathcal{E}_0 \setminus \{T_u, T_u^{-1}\}$ obtaining a structure $\mathfrak{B}_0$, its origin b_0 and a function $\mathfrak{p}_b : B_0 \to A_{\mathfrak{p}(b)} \cap [\mathfrak{p}(b)]_{\mathcal{E}_{tot} \cup \{T_u, T_u^{-1}\}} \subseteq A_0$ with $\mathfrak{p}_b(b_0) = \mathfrak{p}(b)$. Identify b_0 with b and add the remaining elements of $\mathfrak{B}_0$ to L_u^j, retaining the structure. Substructures $\mathfrak{B}_0$ of this kind will be called *subcomponents* (note that all appropriate relations are transitively closed in subcomponents). Extend $\mathfrak{p}$ so that $\mathfrak{p}{\restriction}B_0 = \mathfrak{p}_b$. This finishes the definition of L_u^j.

Step 2: Providing witnesses. For each $b \in L_u^j$ and $1 \leq s \leq m$ independently perform the following procedure. Let $\mathfrak{B}_0$ be the subcomponent created inductively in *Step 1*, such that $b \in B_0$. If $\mathfrak{p}(b)$ has a witness for $\varphi_s(x, y)$ in A_0 then we want to reproduce such a witness for b. Choose one such witness c (being a child of $\mathfrak{p}(b)$) for $\mathfrak{p}(b)$. Let us denote $\beta = \mathrm{atp}^{\mathfrak{A}}(\mathfrak{p}(b), c)$. If $\{T_u xy, T_u^{-1} xy\} \subseteq \beta$ then by Condition (b3) of the inductive assumption b already has an appropriate witness in the subcomponent $\mathfrak{B}_0$. So we do nothing in this case. If $T_u xy \in \beta$ and $T_u^{-1} xy \notin \beta$ then we add a copy c' of c to $L_u^{j+1,init}$; if $T_u xy \notin \beta$ then we add a copy c' of c to $L_{u+1}^{1,init}$. We join b with c' by β and set $\mathfrak{p}(c') = c$.

An attentive reader may be afraid that when adding witnesses for elements of the last sublayer $L_u^{\hat{M}_\varphi+1}$ of L_u we may want to add one of them to the non-existing layer $L_u^{\hat{M}_\varphi+2}$. There is however no such danger, which follows from the following claim.

▷ Claim 8. (i) Let $b \in L_u^{j,init}$ and let $\mathfrak{B}_0$ be the subcomponent created for b in *Step 1*. Then for all $b' \in \mathfrak{B}_0$ we have $\mathfrak{r}_u^{\mathfrak{A}}(\mathfrak{p}(b)) \geq \mathfrak{r}_u^{\mathfrak{A}}(\mathfrak{p}(b'))$. (ii) Let $b \in L_u^j$ and let $c' \in L_u^{j+1}$ be a witness created for b in *Step* 2. Then $\mathfrak{r}_u^{\mathfrak{A}}(\mathfrak{p}(b)) > \mathfrak{r}_u^{\mathfrak{A}}(\mathfrak{p}(c'))$.

Hence, when moving from L_u^j to L_u^{j+1} the T_u-ranks of pattern elements for the elements of these sublayers strictly decrease. Since these ranks are bounded by $\hat{M}_\varphi$, then, even if the T_u-ranks of the patterns of some elements of L_u^1 are equal to $\hat{M}_\varphi$, then, if $L_u^{\hat{M}_\varphi+1}$ is non-empty, the T_u-ranks of the patterns of its elements must be 0, which means that they cannot have witnesses connected to them one-directionally by T_u.

The construction of $\mathfrak{G}^\gamma$ is finished when layer L_{2l} is fully processed. We have added some elements to the interface layer, L_{2l+1}. Recall that it has only its “initial part”.

3.2.2 Joining the components

In this section we take some number of copies of pattern components and arrange them into the desired structure $\mathfrak{A}_0'$, identifying interface elements of some components with the roots of some other. Some care is needed in this process in order to avoid any modifications of the internal structure of closures $\mathfrak{C}_+$ of components $\mathfrak{C}$, which could potentially result from the transitivity of relations. In particular we need to ensure that if for some u a pair of elements of a component $\mathfrak{C}$ is not connected by T_u inside $\mathfrak{C}$, then it will not become connected by a chain of T_u-edges external to $\mathfrak{C}$.

We create a pattern component $\mathfrak{C}^\gamma$ and its extension $\mathfrak{G}^\gamma$ for every $\gamma \in \boldsymbol{\gamma}[A_0]$. Let γ_{a_0} be the generalized type of a_0. Let *max* be the maximal number of interface elements across all the $\mathfrak{G}^\gamma$. For each $\mathfrak{G}^\gamma$ arbitrarily number its interface elements from 1 up to, maximally, *max*.

For each γ we take copies $\mathfrak{G}_{i,\gamma'}^{\gamma,g}$ of $\mathfrak{G}^\gamma$ for $g \in \{0, 1\}$, $1 \leq i \leq max$ and $\gamma' \in \boldsymbol{\gamma}[A_0]$. The parameter g is sometimes called a *color* (*red* or *blue*); it is convenient to think that the non-interface elements of $G_{i,\gamma'}^{\gamma,g}$ are of color g, but its interface elements have color $1-g$, cf. the left part of Fig. 1, as the latter will be later identified with the roots of some components of color $1 - g$. We import the numbering of the interface elements to these copies. We also take an additional copy $\mathfrak{G}_{\perp,\perp}^{\gamma_{a_0},0}$ of $\mathfrak{G}^{\gamma_{a_0}}$. Its root will become the origin of the whole $\mathfrak{A}_0'$. By $\mathfrak{C}_{i,\gamma'}^{\gamma,g}$ we denote the restriction of $\mathfrak{G}_{i,\gamma'}^{\gamma,g}$ to its non-interface elements.

For each γ, g consider extended components of the form $\mathfrak{G}_{\cdot,\cdot}^{\gamma,g}$, where the placeholders $\cdot$ can be substituted with any combination of proper indices. Perform the following procedure for each $1 \leq i \leq max$. Let b be the i-th interface element of any such extended component, let γ' be the generalized type of $\mathfrak{p}(b)$. Identify the i-th interface elements of all $\mathfrak{G}_{\cdot,\cdot}^{\gamma,g}$ with the root c_0 of $\mathfrak{G}_{i,\gamma}^{\gamma',1-g}$. Note that the values of $\mathfrak{p}(c_0)$ and $\mathfrak{p}(b)$ may differ. However, by construction, they have identical generalized types γ'. For the element c^* obtained in this identification step we define $\mathfrak{p}(c^*) = \mathfrak{p}(c_0)$.

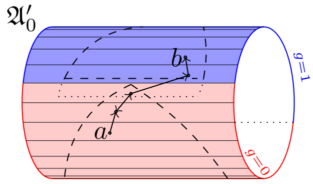

Figure 2 Viewing $\mathfrak{A}^0_0$ and $\mathfrak{A}'_0$ as placed on a cylindrical surface.

Define the graph of components used in the above construction, G^{comp}, by joining two components by an edge iff we identified an interface element of the extended version of one of them with the root of the other. Let A^0_0 be the union of the components accessible from $\mathfrak{C}^{\gamma_{a_0},0}_{\perp,\perp}$ in G^{comp} and let $\mathfrak{A}^0_0$ be the induced structure. Note that in $\mathfrak{A}^0_0$ we still do not take the transitive closures of relations. We define $\mathfrak{A}'_0$ by transitively closing all relations from σ_{dist} in $\mathfrak{A}^0_0$. Finally, we choose as the origin a'_0 of $\mathfrak{A}'_0$ the root of the pattern component $\mathfrak{C}^{\gamma_0,0}_{\perp,\perp}$.

We remark that it is sufficient to take as the universe of $\mathfrak{A}'_0$ the union of the universes of some components $\mathfrak{C}^{\cdot,\cdot}_{\cdot,\cdot}$, and not of their extended versions $\mathfrak{G}^{\cdot,\cdot}_{\cdot,\cdot}$ from which we started our construction, since the interface elements from these extended components were identified with some roots of other components.

For the correctness proof of our construction see the full version of this paper. In this proof it is helpful to think about $\mathfrak{A}^0_0$ and $\mathfrak{A}'_0$ as the structures placed on a cylindrical surface and divided into $4l$ *levels*, see Fig. 2. What is crucial, any transitive path in $\mathfrak{A}^0_0$ can cross at most one of the two borders between colors.

3.2.3 Size of models and complexity

By a rather routine calculation we can show that models produced in the proof of Lemma 7 are of size bounded triply exponentially in the length of input formulas. This finishes the proof of Thm. 6, which immediately gives the decidability of the finite satisfiability problem for UNFO^2+$\mathcal{S}$ and suggests a simple 3-NExpTime-procedure: guess a finite structure of size bounded triply exponentially in the size of input φ and verify that it is indeed a model of φ. We can however do better and show a doubly exponential upper bound matching the known complexity of the general satisfiability problem. For this we design an alternating exponential space algorithm searching for models of the form (f3). The lower bound can be obtained for the two-variable UNFO^2+$\mathcal{S}$ in the presence of one transitive relation by a straightforward adaptation of the lower bound proof for GF^2 with transitive guards [19].

▶ **Theorem 9.** *The finite satisfiability problem for UNFO^2+$\mathcal{S}$ is 2-*ExpTime*-complete.*

4 The general case and its further extensions

In the full version of this paper we generalize the ideas from Section 3 to show:

▶ **Theorem 10.** *The finite satisfiability problem for UNFO+$\mathcal{S}$ is 2-*ExpTime*-complete.*

We also obtain a triply exponential upper bound on the size of minimal finite models of finitely satisfiable formulas. The structure of the proofs is similar to the two-variable case, though some details are more complicated. In particular, we need to go through form (f4) of models: regular trees with bounded ranks of transitive paths. We also explain that in addition to general transitive relations we can use also equivalences and partial orders.

We further extend Thm. 10 by considering an extension, UNFO+$\mathcal{SOH}$, of UNFO+$\mathcal{S}$ by constants and inclusion of binary relations of the form $B_1 \subseteq B_2$, interpreted in a natural way: $\mathfrak{A} \models B_1 \subseteq B_2$ iff $\mathfrak{A} \models \forall xy(B_1xy \rightarrow B_2xy)$.

▶ **Theorem 11.** *The finite satisfiability problem for UNFO+$\mathcal{SOH}$ is 2-*ExpTime*-complete.*

As mentioned in the Introduction, UNFO+$\mathcal{SOH}$ captures several interesting description logics. This implies that we can solve FOMQA problem for them. In particular, we have the following corollary, which, up to our knowledge is the first decidability result for FOMQA in the case of a description logic with both transitive roles and role hierarchies.

▶ **Corollary 12.** *Finite ontology mediated query answering, FOMQA, for the description logic $\mathcal{SHOI}^{\sqcap}$ is decidable and 2-*ExpTime*-complete.*

$\mathcal{SHOI}^{\sqcap}$ and some related logics are considered, e.g., in [11]. For more about FOMQA for description logics with transitivity see [12]. For more about OMQA for description logics see, e.g., references in [12].

Somewhat orthogonally to the extensions motivated by description logics we consider the *base-guarded negation fragment with transitivity*, BGNFO+$\mathcal{S}$, for which the general satisfiability problem was shown decidable in [1]. We do not solve its finite satisfiability problem here, but, analogously to the extension with equivalence relations, UNFO+EQ [7], we are able to lift our results to its one-dimensional restriction, BGNFO$_1$+$\mathcal{S}$, admitting only formulas in which every maximal block of quantifiers leaves at most one variable free.

▶ **Theorem 13.** *The finite satisfiability problem for BGNFO$_1$+$\mathcal{S}$ is 2-*ExpTime*-complete.*

Surprisingly, in contrast to UNFO+$\mathcal{S}$, BGNFO$_1$+$\mathcal{S}$ becomes undecidable when extended by inclusions of binary relations.

5 Conclusions

We proved that the finite satisfiability problem for the unary negation fragment with transitive relations, UNFO+$\mathcal{S}$, is decidable and 2-ExpTime-complete, complementing this way the analogous result for the general satisfiability problem for this logic implied by two other papers. Further, we identified some decidable extensions of our base logic capturing the concepts of nominals and role hierarchies from description logics. We noted that our work has some interesting implications on the finite query answering problem both under the classical (open-world) database scenario as well as in the description logics setting.

One open question is the decidability of the finite satisfiability problem for the full logic BGNFO+$\mathcal{S}$ from [1]. We made a step in this direction here, by solving this problem for the one-dimensional restriction of that logic. Another question is if our techniques can be adapted to a setting in which we do not assert that some distinguished relations are transitive but where we can talk about the transitive closure of the binary relations, or, more generally, to the extension of UNFO with regular path expressions from [16].

We finally remark that we do not know if our small model construction, producing finite models of size bounded triply exponentially in the size of the input formulas, is optimal with respect to the size of models. The best we can do for the lower bound is to enforce models of doubly exponential size (actually, this can be done in UNFO even without transitive relations).

References

1 A. Amarilli, M. Benedikt, P. Bourhis, and M. Vanden Boom. Query Answering with Transitive and Linear-Ordered Data. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence, IJCAI 2016*, pages 893–899, 2016.

2 H. Andréka, J. van Benthem, and I. Németi. Modal Languages and Bounded Fragments of Predicate Logic. *Journal of Philosophical Logic*, 27:217–274, 1998.

3 J.-F. Baget, M. LeClere, and M.-L. Mugnier. Walking the Decidability Line for Rules with Existential Variables. In *Proceedings of the 12th International Conference on Principles of Knowledge Representation and Reasoning, KR 2010*, pages 466–476, 2010.

4 J.-F. Baget, M. Leclère, M.-L. Mugnier, and E. Salvat. Extending Decidable Cases for Rules with Existential Variables. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence, IJCAI 2009*, pages 677–682, 2009.

5 V. Bárány, B. ten Cate, and L. Segoufin. Guarded Negation. *J. ACM*, 62(3):22, 2015.

6 A. Calì, G. Gottlob, and M. Kifer. Taming the Infinite Chase: Query Answering under Expressive Relational Constraints. *J. Artif. Intell. Res.*, 48:115–174, 2013.

7 D. Danielski and E. Kieroński. Unary negation fragment with equivalence relations has the finite model property. In *33rd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2018*, pages 285–294, 2018.

8 D. Danielski and E. Kieronski. Unary negation fragment with equivalence relations has the finite model property. *CoRR*, abs/1802.01318, 2018. `arXiv:1802.01318`.

9 M. Dzieciołowski. Satisfability issues for unary negation logic. Bachelor's thesis, University of Wrocław, 2017.

10 H. Ganzinger, Ch. Meyer, and M. Veanes. The Two-Variable Guarded Fragment with Transitive Relations. In *14th Annual IEEE Symposium on Logic in Computer Science, LICS 1999*, pages 24–34, 1999.

11 B. Glimm and Y. Kazakov. Role Conjunctions in Expressive Description Logics. In *Logic for Programming, Artificial Intelligence, and Reasoning, 15th International Conference, LPAR 2008*, pages 391–405, 2008.

12 T. Gogacz, Y. A. Ibáñez-García, and F. Murlak. Finite Query Answering in Expressive Description Logics with Transitive Roles. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Sixteenth International Conference, KR 2018.*, pages 369–378, 2018.

13 E. Grädel. On The Restraining Power of Guards. *J. Symb. Log.*, 64(4):1719–1742, 1999.

14 E. Grädel, P. Kolaitis, and M. Y. Vardi. On the decision problem for two-variable first-order logic. *Bulletin of Symbolic Logic*, 3(1):53–69, 1997.

15 E. Grädel, M. Otto, and E. Rosen. Undecidability results on two-variable logics. *Archiv für Mathematische Logik und Grundlagenforschung*, 38(4-5):313–354, 1999.

16 J. Ch. Jung, C. Lutz, M. Martel, and T. Schneider. Querying the Unary Negation Fragment with Regular Path Expressions. In *International Conference on Database Theory, ICDT 2018*, pages 15:1–15:18, 2018.

17 Y. Kazakov. *Saturation-based decision procedures for extensions of the guarded fragment.* PhD thesis, Universität des Saarlandes, Saarbrücken, Germany, 2006.

18 E. Kieroński. Results on the Guarded Fragment with Equivalence or Transitive Relations. In *Computer Science Logic*, volume 3634 of *LNCS*, pages 309–324. Springer, 2005.

19 E. Kieroński. On the complexity of the two-variable guarded fragment with transitive guards. *Inf. Comput.*, 204(11):1663–1703, 2006.

20 E. Kieroński and L. Tendera. Finite Satisfiability of the Two-Variable Guarded Fragment with Transitive Guards and Related Variants. *ACM Trans. Comput. Logic*, 19(2):8:1–8:34, 2018.

21 I. Pratt-Hartmann. The Finite Satisfiability Problem for Two-Variable, First-Order Logic with one Transitive Relation is Decidable. *Mathematical Logic Quarterly*, 2018.

22 I. Pratt-Hartmann, W. Szwast, and L. Tendera. The Fluted Fragment Revisited. *Journal of Symbolic Logic*, Forthcoming, 2019.

23 I. Pratt-Hartmann and L. Tendera. The Fluted Fragment with Transitivity. In *44th International Symposium on Mathematical Foundations of Computer Science, MFCS 2019*, pages 18:1–18:15, 2019.

24 W. V. Quine. On the limits of decision. In *Proceedings of the 14th International Congress of Philosophy*, volume III, pages 57–62, 1969.

25 D. Scott. A decision method for validity of sentences in two variables. *Journal Symbolic Logic*, 27:477, 1962.

26 W. Szwast and L. Tendera. The guarded fragment with transitive guards. *Annals of Pure and Applied Logic*, 128:227–276, 2004.

27 B. ten Cate and L. Segoufin. Unary negation. *Logical Methods in Comp. Sc.*, 9(3), 2013.