Unary negation fragment with equivalence relations has the finite model property

Daniel Danielski and Emanuel Kieroński
Institute of Computer Science
University of Wrocław
Poland

Abstract

We consider an extension of the unary negation fragment of first-order logic in which arbitrarily many binary symbols may be required to be interpreted as equivalence relations. We show that this extension has the finite model property. More specifically, we show that every satisfiable formula has a model of at most doubly exponential size. We argue that the satisfiability (= finite satisfiability) problem for this logic is 2-ExpTime-complete. We also transfer our results to a restricted variant of the guarded negation fragment with equivalence relations.

CCS Concepts • **Theory of computation → Finite Model Theory**;

Keywords unary negation fragment, equivalence relations, satisfiability, finite satisfiability, finite model property

ACM Reference Format:
Daniel Danielski and Emanuel Kieroński. 2018. Unary negation fragment with equivalence relations has the finite model property. In *LICS '18: LICS '18: 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, July 9–12, 2018, Oxford, United Kingdom.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3209108.3209205

1 Introduction

A simple yet beautiful idea of restricting negation to subformulas with at most one free variable led ten Cate and Segoufin to a definition of an appealing fragment of first-order logic, called the unary negation fragment, UNFO [21]. UNFO turns out to have very nice algorithmic and model-theoretic properties, and, moreover, it has strong motivations from various areas of computer science. UNFO has the finite model property: every satisfiable formula has a finite model. This immediately implies the decidability of the satisfiability problem (does a given formula have a model?) and the finite satisfiability problem (does a given formula have a finite model?). To get tight complexity bounds one can, e.g., use another convenient property of UNFO, that every satisfiable formula has a tree-like model, and show that satisfiability is 2-ExpTime-complete. What is interesting, the lower bound holds even for bounded variable versions of this logic, and already the fragment with three variables is 2-ExpTime-hard. As several other seminal fragments of first-order logic, like the two variable fragment, FO^2 [16], the guarded fragment, GF [2], and the fluted fragment, FF [18], UNFO embeds propositional (multi)-modal logic, which opens connections to, e.g., such fields as verification of hardware and software or knowledge representation. Moreover, in contrast to the fragments mentioned above, UNFO can express unions of conjunctive queries, which makes it potentially attractive for the database community.

Similarly to most important decidable fragments of first order logic, including FO^2, GF and FF, UNFO has a drawback, which seriously limits its potential applications, namely, it cannot express transitivity of a binary relation, nor a related property of being an equivalence. This justifies studying formalisms, equipping the basic logics with some facilities allowing to express the above mentioned properties. The simplest way to obtain such formalisms is to divide the signature into two parts, a base part and a distinguished part, the latter containing only binary symbols, and impose explicitly some semantic constraints on the interpretations of the symbols from the distinguished part, e.g., require them to be interpreted as equivalences. Generally, the results are negative: both FO^2 and GF become undecidable with equivalences or with arbitrary transitive relations. More specifically, the satisfiability and the finite satisfiability problems for FO^2 and even for the two-variable restriction of GF, GF^2, with two transitive relations [10, 11] or three equivalences [13] are undecidable. Also the fluted fragment is undecidable when extended by equivalence relations [I. Pratt-Hartmann, W. Szwast, L. Tendera, *private communication*]. Positive results were obtained for FO^2 and GF only when the distinguished signature contains just one transitive symbol [17] or two equivalences [12], or when some further syntactic restrictions on the usage of distinguished symbols are imposed [14, 20].

UNFO turns out to be an exception here, since its satisfiability problem remains decidable in the presence of arbitrarily many equivalence or transitive relations. This can be shown by reducing the satisfiability problem for UNFO with equivalences to UNFO with arbitrary transitive relations (see Lemma 2.2). The decidability and 2-ExpTime-completeness of the satisfiability problem for the latter follow from two independent recent works, respectively by Jung et al. [9] and by Amarilli et al. [1]. In the first of them the decidability of UNFO with transitivity is stated explicitly, as a corollary from the decidability of the unary negation fragment with regular path expressions. The second shows decidability of the guarded negation fragment, GNFO, with transitive relations restricted to non-guard positions (for more about this logic see Section 5), which embeds UNFO with transitive relations.

Both the above mentioned decidability results are obtained by employing tree-like model properties of the logics and then using some automata techniques. Since tree-like unravellings of models are infinite, such approach works only for general satisfiability, and gives no insight into the decidability/complexity of the finite satisfiability problem.

LICS '18, July 9–12, 2018, Oxford, United Kingdom

ACM ISBN 978-1-4503-5583-4/18/07…$15.00
https://doi.org/10.1145/3209108.3209205

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 are finite. For example, models of programs have finite numbers of states and possible actions and real world databases contain finite sets of facts. Under such scenarios, an ability of solving only the general satisfiability problem may not be fully satisfactory.

In this paper we show that UNFO with arbitrarily many equivalence relations, UNFO+EQ, has the finite model property. It follows that the finite satisfiability and the general satisfiability problems for the considered logic coincide, and, due to the above mentioned reduction to UNFO with transitive relations, can be solved in 2-ExpTime. The corresponding lower bound can be obtained even for the two-variable version of the logic, in the presence of just two equivalence relations. We further transfer our results to the intersection of GNFO with equivalence relations on non-guard positions and the one-dimensional fragment [8]. A formula is *one-dimensional* if its every maximal block of quantifiers leaves at most one variable free. Moving from UNFO to this restricted variant of GNFO significantly increases the expressive power.

Studying equivalence relations may be seen as a step towards understanding finite satisfiability of UNFO or GNFO with arbitrary transitive relations. However, equivalence relations are also interesting on its own and in computer science were studied in various contexts. They play an important role in modal and epistemic logics, and were considered in the area of interval temporal logics [15]. Data words [4] and data trees [5], studied in the context of XML reasoning use an equivalence relation to compare data values, which may come from a potentially infinite alphabet; we remark, that, again, decidability results over such data structures are obtained only in the presence of a single equivalence relation, that is they allow to compare objects only with respect to a single parameter.

Related work. There are not too many decidable fragments of first-order logic whose finite satisfiability is known to remain decidable when extended by an unbounded number of equivalence relations. One exception is the two-variable guarded fragment with equivalence guards, GF^2+EG, a logic without the finite model property, whose finite satisfiability is NExpTime-complete [14]. GF^2+EG slightly differs in spirit from the mentioned decidable variant of GNFO with equivalence relations on non-guard positions, and thus also from UNFO+EQ which is a fragment of the latter. We remark however that these two approaches are not completely orthogonal. E.g., a GF^2+EG formula $\forall xy(E(x,y) \rightarrow (P(x) \wedge P(y)))$, in which atom $E(x,y)$ is used as a guard, when treated as a GNFO formula has $E(x,y)$ on a non-guard position; actually, it is a UNFO+EQ formula. Simply, guards play slightly different roles in GF and GNFO.

The decidability of the satisfiability problem for both GF^2+EG and UNFO+EQ can be shown relatively easily, by exploiting tree-based model properties for both logics. The analysis of the corresponding finite satisfiability problems is much more challenging. It turns out that the difficulties arising when considering GF^2+EG and UNFO+EQ are of different nature. The main problem in the case of GF^2+EG is that it allows, using guarded occurrences of inequalities $x \neq y$, to restrict some types of elements to appear at most once in every abstraction class of the guarding equivalence relation. This causes that some care is needed when performing surgery on models, and seems to require a global view at some of their properties. Indeed, the solution employs integer programming to describe some global constraints on models of the given formula. What is however worth remarking, in the case of GF^2+EG one can always construct models in which every pair of elements is connected by at most one equivalence. So, GF^2+EG does not allow for a real interaction among equivalence relations.

Inequalities $x \neq y$ are not allowed in UNFO+EQ, and indeed we do not have here any problems with duplicating elements of any type. On the other hand, UNFO+EQ allows for a non-trivial interaction among equivalences, and this seems to be the source of main obstacles for finite model constructions. Surprisingly, such obstacles are present already in the two-variable version of our logic. More intuitions about problems arising will be given later.

Our solution employs a novel (up to our knowledge) inductive approach to build a finite model of a satisfiable formula, starting from an arbitrary model. In the base of induction we construct some initial fragments in which none of the equivalences plays an important role. Such fragments are then joined into bigger and bigger structures, in which more and more equivalences become significant. This process eventually yields a finite model of the given formula.

Organization of the paper. Section 2 contains formal definitions and presents some basic facts. In Section 3 we show the finite model property for a restricted, two-variable variant of our logic, $UNFO^2$+EQ. We believe that treating this simpler setting first will help the reader to understand our ideas and techniques, since it allows them to be presented without some quite complicated technical details appearing in the general case. Then in Section 4 we describe the generalization of our construction working for full UNFO+EQ, pinpointing the main differences and additional difficulties arising in comparison to the two-variable case. In Section 5 we transfer our results to the one-dimensional guarded negation fragment with equivalences. Section 6 concludes the paper.

2 Preliminaries

2.1 Logics and structures

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

We work with purely relational signatures $\sigma = \sigma_{\text{base}} \cup \sigma_{\text{dist}}$ where σ_{base} is the *base signature* and σ_{dist} is the *distinguished signature*. All symbols from σ_{dist} are binary. Over such signatures we define the *unary negation fragment of first-order logic*, UNFO as in [21] by the following grammar:

$$\varphi = R(\bar{x}) \mid x = y \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \exists x\varphi \mid \neg\varphi(x)$$

where, in the last clause, φ has no free variables besides (at most) x.

A typical formula not expressible in UNFO is $x \neq y$. We formally do not allow universal quantification. However we will 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 $\forall xy\neg P(x,y)$ is in UNFO but $\forall xyP(x,y)$ is not.

The *unary negation fragment with equivalences*, UNFO+EQ is defined by the same grammar as UNFO. When satisfiability of its formulas is considered, we restrict the class of admissible models to those which interpret all symbols from σ_{dist} as equivalence relations. We also mention an analogous logic UNFO+TR in which the symbols from σ_{dist} are interpreted as (arbitrary) transitive relations.

2.2 Atomic types

An *atomic k-type* (or, shortly, a *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 will sometimes identify a k-type with the conjunction of its elements. Given a σ-structure $\mathfrak{A}$ and a tuple $a_1, \ldots, a_k \in A$ we denote by $\mathrm{atp}^{\mathfrak{A}}(a_1, \ldots, a_k)$ the atomic k-type *realized* by $a_1, \ldots, a_k$, that is the unique k-type $\alpha(x_1, \ldots, x_k)$ such that $\mathfrak{A} \models \alpha(a_1, \ldots, a_k)$.

2.3 Normal form and witness structures

We say that an UNFO+EQ formula is in Scott-normal form if it is of the shape

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

where each φ_i is an UNFO+EQ quantifier-free formula. This kind of normal form was introduced in the bachelor's thesis [6].

Lemma 2.1. *For any UNFO+EQ formula φ one can compute in polynomial time a normal form UNFO+EQ formula φ' over signature extended by some fresh unary symbols, such that any model of φ' is a model of φ and any model of φ can be expanded to a model of φ' by an appropriate interpretation of the additional unary symbols.*

The proof of Lemma 2.1 first converts φ into the so-called UN-normal form (see [21]) and then uses the standard Scott's technique [19] of replacing subformulas starting with blocks of quantifiers by unary atoms built out using fresh unary symbols, and appropriately axiomatizing the fresh unary relations.

Lemma 2.1 allows us, when dealing with decidability/complexity issues for UNFO+EQ, or when considering the size of minimal finite models of formulas, to restrict attention to normal form sentences.

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. For an element a and every conjunct φ_i choose a witness structure $\mathfrak{D}_i$. Then the structure $\mathfrak{D} = \mathfrak{A}\restriction\{D_1 \cup \ldots \cup D_m\}$ is called a *φ-witness structure* for a.

2.4 Basic facts

In FO^2 or in GF^2 extended by transitive relations one can enforce a transitive relation T to be an equivalence (it suffices to add conjuncts saying that T is reflexive and symmetric). The same is possible, by means of a simple trick (see [11]), even in the variant of GF^2 in which transitive relations can appear only as guards. It is however not possible in UNFO+TR. Indeed, it is not difficult to see that if $\mathfrak{A}$ is a model of an UNFO+TR formula φ in which all symbols from σ_{dist} are interpreted as equivalences then another model of φ can be constructed by taking two disjoint copies of $\mathfrak{A}$, choosing a symbol T from σ_{dist}, joining every element a from the first copy of $\mathfrak{A}$ with its isomorphic image in the second copy by the 2-type containing $T(x, y)$ as the only positive non-unary literal (in particular this 2-type contains $\neg T(y, x)$), and transitively closing T. In this model the interpretation of T is no longer an equivalence. However:

Lemma 2.2. *There is a polynomial time reduction from the satisfiability (finite satisfiability) problem for UNFO+EQ to the satisfiability (finite satisfiability) problem for UNFO+TR.*

Proof. Take an UNFO+EQ formula φ, convert it into normal form formula φ' and transform φ' into UNFO+TR formula φ'' in the following way: (i) replace in φ' every atom of the form $E(x, y)$ (for any variables x, y) by $E(x, y) \wedge E(y, x)$, (ii) add to φ'' the conjunct $\forall x E(x, x)$ for every distinguished symbol E. Now, any model of φ' is a model of φ''; and any model of φ'' can be transformed into a model of φ' by removing all non-symmetric transitive connections. □

The decidability and 2-ExpTime-completeness of UNFO+TR has been recently shown in [9]. Taking into consideration that even without equivalences/transitive relations UNFO is 2-ExpTime-hard we can state the following corollary.

Theorem 2.3. *The (general) satisfiability problem for UNFO+EQ is 2-ExpTime-complete.*

We recall that UNFO+TR is contained in *the base-guarded negation fragment* with transitivity, BGNFO+TR, in which transitive relations are allowed only at non-guard positions, and the latter logic has been recently shown decidable and 2-ExpTime-complete by Amarilli et al. in [1]. This gives an alternative argument for Thm. 2.3. We will return to BGNFO+TR in Section 5.

As said in Introduction both the decidability proof for UNFO+TR from [9] and the decidability proof for BGNFO+TR from [1] strongly rely on infinite tree-like unravellings of models, and thus they give no insight into the decidability/complexity of finite satisfiability.

Let us formulate now a simple but crucial observation on models of UNFO+EQ formulas.

Lemma 2.4. *Let $\mathfrak{A}$ be a model of a normal form UNFO+EQ formula φ. Let $\mathfrak{A}'$ be a structure in which all relations from σ_{dist} are equivalences such that*

1. *for every $a' \in A'$ there is a φ-witness structure for a' in $\mathfrak{A}'$.*
2. *for every tuple $a'_1, \ldots, a'_t$ (recall that t is the number of variables of the $\forall$-conjunct of φ) of elements of 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 $\mathfrak{A}' \models \varphi$.

Proof. Due to (1) all elements of $\mathfrak{A}'$ have the required witness structures for all $\forall\exists$-conjuncts. It remains to see that the $\forall$-conjunct is not violated. But since $\mathfrak{A} \models \neg\varphi_0(\mathfrak{h}(a_1), \ldots, \mathfrak{h}(a_t))$ and φ_0 is a quantifier-free formula in which only unary atoms may be negated, it is straightforward. □

The above observation leads in particular to a tree-like model property for UNFO+EQ. We define a *φ-tree-like unravelling* $\mathfrak{A}'$ of $\mathfrak{A}$ and a 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 put to level L_0 of A' an element a' such that $\mathrm{atp}^{\mathfrak{A}'}(a') = \mathrm{atp}^{\mathfrak{A}}(a)$; set $\mathfrak{h}(a') = a$. Having defined L_i repeat the following for every $a' \in L_i$. Choose in $\mathfrak{A}$ a φ-witness structure for $\mathfrak{h}(a')$. Assume it consists of $\mathfrak{h}(a'), a_1, \ldots, a_s$. Add a fresh copy a'_j of every a_j to L_{i+1}, make $\mathfrak{A}' \restriction \{a', a'_1, \ldots, a'_s\}$ isomorphic to $\mathfrak{A} \restriction \{\mathfrak{h}(a'), a_1, \ldots, a_s\}$ and set $\mathfrak{h}(a'_i) = a_i$. Complete the definition of $\mathfrak{A}'$ transitively closing all equivalences.

Lemma 2.5. *Let $\mathfrak{A}$ be a model of a normal form UNFO+EQ formula φ. Let $\mathfrak{A}'$ be a φ-tree-like unravelling of $\mathfrak{A}$. Then $\mathfrak{A}' \models \varphi$.*

Proof. It is readily verified that $\mathfrak{A}'$ meets the properties required by Lemma 2.4. In particular $\mathfrak{h}$ acts as the required homomorphism. □

Slightly informally, we say that a model of a normal form formula φ is *tree-like* if it has a shape similar to the structure $\mathfrak{A}'$ from the above lemma, that is: (i) it can be divided into levels, (ii) every element from level i has its φ-witness structure completed in level $i+1$, (iii) φ-witness structures for different elements from the same level are disjoint, (iv) only elements from the same witness structure may be joined by relations from σ_{base}, (v) the only σ_{dist}-connections among elements not belonging to the same witness structure are the result of closing transitively the equivalences in witness structures.

3 Small model theorem for UNFO^2+EQ

In this section we consider UNFO^2+EQ—the two-variable restriction of UNFO+EQ. We show the following theorem.

Theorem 3.1. *Every satisfiable UNFO^2+EQ formula φ has a finite model of size bounded doubly exponentially in $|\varphi|$.*

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 for FO^2 [19]:

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

where all φ_i are quantifier-free UNFO^2 formulas. Without loss of generality we assume that φ does not use relational symbols of arity greater than 2 (cf. [7]).

Let us fix a satisfiable normal form UNFO+EQ formula φ, and the finite relational signature $\sigma = \sigma_{\text{base}} \cup \sigma_{\text{dist}}$ consisting of those symbols that appear in φ. Enumerate the equivalence relation symbols as $\sigma_{\text{dist}} = \{E_1, \ldots, E_k\}$. Fix a (not necessarily finite) σ-structure $\mathfrak{A} \models \varphi$. We will show how to build a finite model of φ.

Generally, we will work in an expected way, starting from copies of some elements from $\mathfrak{A}$, adding for them fresh witnesses (using some patterns of connections extracted from $\mathfrak{A}$), then providing fresh witnesses for the previous witnesses, and so on. At some point, instead of producing new witnesses, we need a strategy of using only a finite number of them. It is perhaps worth explaining what are the main difficulties in such a kind of construction. A naive approach would be to unravel $\mathfrak{A}$ into a tree-like structure, like in Lemma 2.5, then try to cut each branch of the tree at some point a and look for witnesses for a among earlier elements. The problem is when we try to reuse an element b as a witness for a, and b is already connected to a by some equivalence relations. Then if a needs needs to join to b by some other equivalences then the resulting 2-type may become inconsistent with $\neg\varphi_0$. Another danger, similar in spirit, is that some b may be needed as a witness for several elements, $a_1, \ldots, a_s$. Then some of the a_is may become connected by some equivalences which, again, may be forbidden.

It seems to be a non-trivial task to find a safe strategy of providing witnesses using only finitely many elements and avoiding conflicts described above. This is why we employ a rather intricate inductive approach. We will produce substructures of the desired finite model in which some number of equivalences are total, using patterns extracted from corresponding substructures of the original model. Intuitively, knowing that an equivalence is total, we can forget about it in our construction. Roughly speaking, our induction goes on the number of equivalence relations which are not total in the given substructures. The constructed substructures will later become fragments of bigger and bigger substructures, which will eventually form the whole model. To enable composing bigger substructures from smaller ones in our inductive process we will additionally keep some information about the intended *generalized types* of elements in form of a pattern function pointing them to elements in the original model.

Let us turn to the details of the proof. Denote by $\boldsymbol{\alpha}$ the set of atomic 1-types realized in $\mathfrak{A}$. Note that $|\boldsymbol{\alpha}|$ is bounded exponentially in $|\sigma|$ and thus also in $|\varphi|$. In this section we will use (possibly decorated) symbol α to denote 1-types and β to denote 2-types.

We now introduce a notion of a generalized type which stores slightly more information about an element in a structure than its atomic 1-type. For a set S we denote by $\mathcal{P}(S)$ the powerset of S.

Definition 3.2. A *generalized type* (over σ) is a pair $(\alpha, \mathfrak{f})$ where α is an atomic 1-type, and $\mathfrak{f}$ is an *eq-visibility function*, that is a function of type $\mathcal{P}(\sigma_{\text{dist}}) \to \mathcal{P}(\boldsymbol{\alpha})$, such that, for every $\mathcal{E} \subseteq \sigma_{\text{dist}}$ we have $\alpha \in \mathfrak{f}(\mathcal{E})$, and for every $\mathcal{E}_1 \subseteq \mathcal{E}_2 \subseteq \sigma_{\text{dist}}$ we have $\mathfrak{f}(\mathcal{E}_2) \subseteq \mathfrak{f}(\mathcal{E}_1)$. Given a generalized type $\bar{\alpha}$ we will denote by $\bar{\alpha}.\mathfrak{f}$ its eq-visibility function. We say that an element $a \in A$ *realizes* a generalized type $\bar{\alpha} = (\alpha, \mathfrak{f})$ in $\mathfrak{A}$, and write $\text{gtp}^{\mathfrak{A}}(a) = \bar{\alpha}$ if (i) $\alpha = \text{atp}^{\mathfrak{A}}(a)$, (ii) for $\mathcal{E} \subseteq \sigma_{\text{dist}}$, $\bar{\alpha}.\mathfrak{f}(\mathcal{E}) = \{\text{atp}^{\mathfrak{A}}(b) : \mathfrak{A} \models E_i ab \text{ for all } E_i \in \mathcal{E}\}$. We say that a generalized type $\bar{\alpha}_1 = (\alpha_1, \mathfrak{f}_1)$ is a *safe reduction* of $\bar{\alpha}_2 = (\alpha_2, \mathfrak{f}_2)$ if $\alpha_1 = \alpha_2$ and for every $\mathcal{E} \subseteq \sigma_{\text{dist}}$ we have $\mathfrak{f}_1(\mathcal{E}) \subseteq \mathfrak{f}_2(\mathcal{E})$. We denote by $\bar{\boldsymbol{\alpha}}$ the set of generalized types realized in $\mathfrak{A}$, and for $B \subseteq A$ we denote by $\bar{\boldsymbol{\alpha}}[B]$ the subset of $\bar{\boldsymbol{\alpha}}$ consisting of the generalized types realized by elements of B.

We are ready to formulate our inductive lemma.

Lemma 3.3. *Let l_0 be a natural number $0 \le l_0 \le k$ and let $\mathcal{E}_0$ be a subset of σ_{dist} of size l_0. Denote by $\mathcal{E}_{tot}$ the set $\sigma_{dist} \setminus \mathcal{E}_0$, and by E^* the equivalence relation $\bigcap_{E_i \in \mathcal{E}_{tot}} E_i$.* [1] *Let $a_0 \in A$, let A_0 be the E^*-equivalence class of a_0 in $\mathfrak{A}$, and let $\mathfrak{A}_0$ be the induced substructure of $\mathfrak{A}$. Then there exists a finite structure $\mathfrak{B}_0$ and a function $\mathfrak{p} : B_0 \to A_0$ such that:*

- *(b1) All relations from $\mathcal{E}_{tot}$ are total in $\mathfrak{B}_0$.*
- *(b2) For every $b \in B_0$ if $\mathfrak{p}(b)$ has a witness w for $\varphi_i(x,y)$ in $\mathfrak{A}_0$ then there is $w' \in B_0$ such that $\text{atp}^{\mathfrak{A}_0}(\mathfrak{p}(b), w) = \text{atp}^{\mathfrak{B}_0}(b, w')$.*
- *(b3) For every $\mathcal{E} \subseteq \sigma_{dist}$, for every $b_1, b_2 \in B_0$ if for all $E_i \in \mathcal{E}$ $\mathfrak{B}_0 \models E_i(b_1, b_2)$ then $\text{gtp}^{\mathfrak{A}}(\mathfrak{p}(b_1)).\mathfrak{f}(\mathcal{E}) = \text{gtp}^{\mathfrak{A}}(\mathfrak{p}(b_2)).\mathfrak{f}(\mathcal{E})$.*
- *(b4) For every $b \in B_0$ we have that $\text{gtp}^{\mathfrak{B}_0}(b)$ is a safe reduction of $\text{gtp}^{\mathfrak{A}}(\mathfrak{p}(b))$.*
- *(b5) Every 2-type realized in $\mathfrak{B}_0$ is either also realized in $\mathfrak{A}_0$ or is obtained from a type realized in $\mathfrak{A}_0$ by removing from it all positive σ_{base}-binary atoms and possibly some equivalence connections and/or equalities.*
- *(b6) a_0 is in the image of $\mathfrak{p}$.*

$\mathfrak{B}_0$ may be seen as a small counterpart of $\mathfrak{A}_0$ in which every element b has witnesses for those φ_i-s for which $\mathfrak{p}(b)$ has witnesses in $\mathfrak{A}_0$. Intuitively, we may think that other witnesses required by b are *promised* by a link to $\mathfrak{p}(b)$ and will be provided in further steps.

Before we prove Lemma 3.3 let us see that it indeed implies the desired finite model property from Thm. 3.1. To this end, take as a_0 an arbitrary element of $\mathfrak{A}$ and consider $l_0 = k$. In this case $\mathcal{E}_0 = \{E_1, \ldots, E_k\}$, $\mathcal{E}_{tot} = \emptyset$, and $\mathfrak{A}_0 = \mathfrak{A}$. We claim that the structure $\mathfrak{B}_0$ produced now by an application of Lemma 3.3 is a model of φ. First, Condition (b2) ensures that all elements of $\mathfrak{B}_0$ have the required witnesses. Second, (b5) guarantees that for every pair of elements $b_1, b_2 \in B_0$ there is a homomorphism $\mathfrak{B}_0 \restriction \{b_1, b_2\} \to \mathfrak{A}$

[1] If $\mathcal{E}_{tot} = \emptyset$ then E^* is the total relation.

preserving the 1-types of elements; due to part (2) of Lemma 2.4 this implies that the conjunct $\forall xy \neg\varphi_0(x, y)$ is satisfied in $\mathfrak{B}_0$.

The rest of this section is devoted to a proof of Lemma 3.3. We proceed by induction over l_0. Consider the base of induction, $l_0 = 0$. In this case all equivalences in $\mathfrak{A}_0$ are total. Without loss of generality assume that $|A_0| = 1$. If this is not the case just add to σ_{dist} a fake symbol E_{k+1} and interpret it in $\mathfrak{A}$ as the identity relation. We take $\mathfrak{B}_0 = \mathfrak{A}_0$ and $\mathfrak{p}(a) = a$ for the only $a \in A_0$. Properties (b1)–(b6) are obvious.

Let us turn to the inductive step. Assume that Thm. 3.3 holds for some $l_0 = l - 1$, $0 < l < k$ and let us show that it also holds for $l_0 = l$. To this end let $\mathcal{E}_0$ be a subset of σ_{dist} of size l, $a_0 \in A$ and let $\mathcal{E}_{tot}$, E^* and $\mathfrak{A}_0$ be as in the statement of Thm. 3.3. Without loss of generality let us assume that $\mathcal{E}_0 = \{E_1, \ldots, E_l\}$.

To build $\mathfrak{B}_0$ we first prepare some basic building blocks for our construction, called *tree-like components*, or *components* for short.

3.1 Tree-like components

3.1.1 Informal description and the desired properties

A tree-like component is a finite structure whose universe is divided into levels $L_1, \ldots, L_{l+1}$. In each level L_i we additionally distinguish its *initial part*, L_i^{init}. L_1^{init} consists of a single element, called the *root* of the component. The elements of level L_{l+1} are called *leaves* of the component. It may happen that some L_i is empty. In such case also all levels L_j for $j > i$ are empty, in particular there are no leaves.

We define a *pattern component* for every generalized type from $\bar{\boldsymbol{\alpha}}[A_0]$. The pattern component constructed for $\bar{\alpha}$ will be denoted $\mathfrak{C}^{\bar{\alpha}}$. Along with the construction of $\mathfrak{C}^{\bar{\alpha}}$ we are going to define a function $\mathfrak{p}$ assigning elements of A_0 to elements of $C^{\bar{\alpha}}$. Later we take some number of copies of every pattern component and join them forming the desired structure $\mathfrak{B}_0$. The values of $\mathfrak{p}$ will be imported to B_0 from the pattern components.

Let us describe the properties which we are going to obtain during the construction of $\mathfrak{C}^{\bar{\alpha}}$:

- *(c1) All relations from $\mathcal{E}_{tot}$ are total in $\mathfrak{C}^{\bar{\alpha}}$.*
- *(c2) For every $c \in C^{\bar{\alpha}} \setminus L_{l+1}$ if $\mathfrak{p}(c)$ has a witness w for $\varphi_i(x, y)$ in $\mathfrak{A}_0$ then there is $w' \in C^{\bar{\alpha}}$ such that $\text{atp}^{\mathfrak{A}_0}(\mathfrak{p}(b), w) = \text{atp}^{\mathfrak{C}^{\bar{\alpha}}}(c, w')$.*
- *(c3) For every $\mathcal{E} \subseteq \sigma_{dist}$, for every $c_1, c_2 \in C^{\bar{\alpha}}$, if for all $E_i \in \mathcal{E}$ $\mathfrak{C}^{\bar{\alpha}} \models E_i(c_1, c_2)$ then $\text{gtp}^{\mathfrak{A}}(\mathfrak{p}(c_1)).\mathfrak{f}(\mathcal{E}) = \text{gtp}^{\mathfrak{A}}(\mathfrak{p}(c_2)).\mathfrak{f}(\mathcal{E})$.*
- *(c4) For every $c \in C^{\bar{\alpha}}$ we have that $\text{gtp}^{\mathfrak{C}^{\bar{\alpha}}}(c)$ is a safe reduction of $\text{gtp}^{\mathfrak{A}}(\mathfrak{p}(c))$.*
- *(c5) every 2-type realized in $\mathfrak{C}^{\bar{\alpha}}$ is either a type realized also in $\mathfrak{A}_0$ or is obtained from a type realized in $\mathfrak{A}_0$ by removing from it all σ_{base}-binary symbols and possibly some equivalences and/or equalities.*
- *(c6) If a pair of elements is joined by a relation from σ_{base} then they belong to the same level or to two consecutive levels.*
- *(c7) For $0 < i < l + 1$ the elements from L_i and L_{i+1} are not joined by relation E_i; hence the root is not connected to any leaf by any relation from $\mathcal{E}_0$.*

In particular, a component will satisfy almost all the properties required for $\mathfrak{B}_0$ by Thm. 3.3. What is missing are witnesses for leaves. A schematic view of a component is shown in Fig. 1.

3.1.2 Building a pattern component.

Let us turn to the details of construction. Let $\bar{\alpha}$ be a generalized type realized in $\mathfrak{A}$ by an element $r \in A_0$. If $\bar{\alpha}$ is the type of a_0 then assume $r = a_0$. We define a component $\mathfrak{C}^{\bar{\alpha}}$. To L_1^{init} we put r' which is a copy of r (that is, $\text{atp}^{\mathfrak{C}^{\bar{\alpha}}}(r') = \text{atp}^{\mathfrak{A}}(r)$), and set $\mathfrak{p}(r') = r$. The element r' is the root of $\mathfrak{C}^{\bar{\alpha}}$.

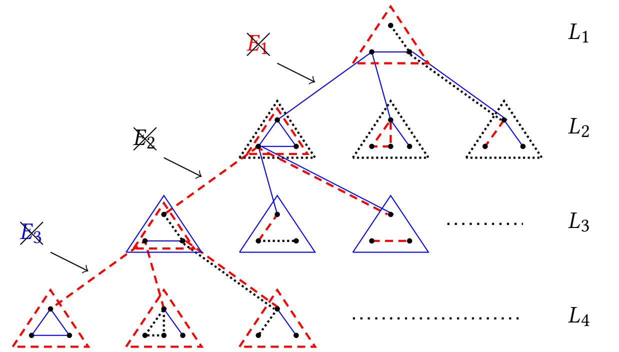

Figure 1. A component for $l = 3$. Triangles correspond to subcomponents. Dashed lines represent E_1, dotted are used for E_2 and solid for E_3. L_i and L_{i+1} are not joined by E_i.

Step 1: Subcomponents. Assume that we have defined $L_1, \ldots, L_{i-1}$, the initial part of L_i, and the structure of $\mathfrak{C}^{\bar{\alpha}}$ on $L_1 \cup \ldots \cup L_{i-1} \cup L_i^{\text{init}}$ for some $i \geq 1$. Assume that the values of $\mathfrak{p}$ on $L_1 \cup \ldots \cup L_{i-1} \cup L_i^{\text{init}}$ have also been defined. Let us explain how to construct the remaining part of level L_i. Take any element $c \in L_i^{\text{init}}$. Let $a_1 = \mathfrak{p}(c)$. Let $A_1 \subseteq A_0$ be the E_i-equivalence class of a_1 in $\mathfrak{A}_0$ (note that A_1 need not be the whole E_i-equivalence class of a_1 in $\mathfrak{A}$). Let $\mathcal{E}_1 = \mathcal{E}_0 \setminus \{E_i\}$. Note that all relations from $\sigma_{\text{dist}} \setminus \mathcal{E}_1$ are total in $\mathfrak{A}_1$, and $|\mathcal{E}_1| = l - 1$. Thus we can use the inductive assumption for $\mathcal{E}_1$, a_1 and $\mathfrak{A}_1$ and produce a structure $\mathfrak{B}_1$ and a function $\mathfrak{p} : B_1 \longrightarrow A_1$, satisfying properties listed in Thm. 3.3. We put to $L_i \setminus L_i^{\text{init}}$ a copy of each element from B_1 besides one element b_1 such that $\mathfrak{p}(b_1) = a_1$ (such element exists due to Condition (b6) of the inductive assumption). On the set consisting of c and all the elements added in this step we define the structure isomorphic to $\mathfrak{B}_1$, identifying c with b_1. We will further call such substructures of components *subcomponents*. We import the values of $\mathfrak{p}$ to the newly added elements from $\mathfrak{B}_1$. We repeat it independently for all $c \in L_i^{\text{init}}$. To complete the definition of the structure on $L_1 \cup \ldots \cup L_i$ we just transitively close all the equivalences.

Step 2. Adding witnesses. Having defined L_i, if $i < l + 1$ we now define L_{i+1}^{init}. Take any element $c \in L_i$. For every $1 \leq j \leq m$, if $\mathfrak{p}(c)$ has a witness $w \in A_0$ for $\varphi_j(x, y)$ then we want to reproduce such a witness for c. Let us denote $\beta = \text{atp}^{\mathfrak{A}}(\mathfrak{p}(c), w)$. If $E_i(x, y) \in \beta$ then by Condition (b2) of the inductive assumption c has an appropriate witness in the subcomponent added in the previous step. If $E_i(x, y) \notin \beta$ then we add a copy w' of w to L_{i+1}^{init}, join c with w' by β and set $\mathfrak{p}(w') = w$. Repeat this procedure independently for all $c \in L_i$. To complete the definition of the structure on $L_1 \cup \ldots \cup L_i \cup L_{i+1}^{\text{init}}$ we again transitively close all the equivalences.

The construction of the component is finished when L_{l+1} is defined. For further purposes let us enumerate the elements of L_{l+1} of the defined pattern component $\mathfrak{C}^{\bar{\alpha}}$ as $c_1^{\bar{\alpha}}, c_2^{\bar{\alpha}}, \ldots$

Let us see that we indeed obtain the desired properties.

Claim 3.4. *The constructed component satisfies the conditions below.*

- *(c1) Any pair of elements belonging to the same subcomponent is connected by all relations from $\mathcal{E}_{tot}$ by the inductive assumption;*

every 2-type used to connect an element from one subcomponent with its witness in another subcomponent is copied from $\mathfrak{A}_0$, and thus it contains all relations from $\mathcal{E}_{tot}$; from any element from the component one can reach every other element by connections inside subcomponents and by connections joining elements with their witnesses which means that the steps of transitively closing σ_{dist}-connections will make all pairs of elements connected by all relations from $\mathcal{E}_{tot}$.

(c2) *This is explicitly taken care in* Step: Adding witnesses. *A suspicious reader may be afraid that during the step of taking transitive closure of equivalences some additional equivalences may be added to a 2-type used to join an element with its witness. This however cannot happen. It follows from tree-like structure of components and from the inductive assumption.*

(c3) *If $\mathcal{E} \subseteq \mathcal{E}_{tot}$ then observe that $\mathfrak{p}(c_1)$ and $\mathfrak{p}(c_2)$ are connected by all relations from $\mathcal{E}$ since they both belong to A_0; this immediately implies the claim. If $\mathcal{E}$ contains $E_i \notin E_{tot}$ then by construction there is a sequence of elements $c_1 = d_1, d_2, \ldots, d_{2u-1}, d_{2u} = c_2$ such that (i) d_i is joined with d_{i+1} by all equivalences from $\mathcal{E}$, (ii) d_{2i-1} and d_{2i} belong to same subcomponent (it may happen that $d_{2i-1} = d_{2i}$), and (iii) d_{2i} and d_{2i+1} belong to two different subcomponents and d_{2i} was added as a witness for d_{2i+1} or vice versa. Now, by condition (b3) of the inductive assumption applied to subcomponents $\mathrm{gtp}^{\mathfrak{A}}(\mathfrak{p}(d_{2i-1})).\mathfrak{f}(\mathcal{E}) = \mathrm{gtp}^{\mathfrak{A}}(\mathfrak{p}(d_{2i})).\mathfrak{f}(\mathcal{E})$. By our construction $\mathrm{atp}^{\mathfrak{A}}(\mathfrak{p}(d_{2i}), \mathfrak{p}(d_{2i+1})) = \mathrm{atp}^{\mathfrak{C}^{\alpha}}(d_{2i}, d_{2i+1})$ and thus $\mathfrak{p}(d_{2i})$ and $\mathfrak{p}(d_{2i+1})$ are joined in $\mathfrak{A}$ by all equivalences from $\mathcal{E}$, which gives that $\mathrm{gtp}^{\mathfrak{A}}(\mathfrak{p}(d_{2i-1})).\mathfrak{f}(\mathcal{E}) = \mathrm{gtp}^{\mathfrak{A}}(\mathfrak{p}(d_{2i})).\mathfrak{f}(\mathcal{E})$. It follows that $\mathrm{gtp}^{\mathfrak{A}}(\mathfrak{p}(c_1)).\mathfrak{f}(\mathcal{E}) = \mathrm{gtp}^{\mathfrak{A}}(\mathfrak{p}(c_2)).\mathfrak{f}(\mathcal{E})$.*

(c4) *The equality of 1-types of c_1 and $\mathfrak{p}(c_1)$ follows from our choices of values of $\mathfrak{p}$. Take any $\mathcal{E} \subseteq \sigma_{dist}$ and let $\alpha' \in \mathrm{gtp}^{\mathfrak{C}^{\bar{\alpha}}}(c).\mathfrak{f}(\mathcal{E})$. This means that there exists an element $c' \in C^{\alpha}$ of 1-type α' joined with c by all relations from $\mathcal{E}$. By (c3) $\mathrm{gtp}^{\mathfrak{A}}(\mathfrak{p}(c)).\mathfrak{f}(\mathcal{E}) = \mathrm{gtp}^{\mathfrak{A}}(\mathfrak{p}(c')).\mathfrak{f}(\mathcal{E})$, and since all relations from $\mathcal{E}$ are equivalences $\alpha' \in \mathrm{gtp}^{\mathfrak{A}}(\mathfrak{p}(c')).\mathfrak{f}(\mathcal{E})$ and thus also $\alpha' \in \mathrm{gtp}^{\mathfrak{A}}(\mathfrak{p}(c)).\mathfrak{f}(\mathcal{E})$. This shows that $\mathrm{gtp}^{\mathfrak{C}^{\bar{\alpha}}}(c)$ is a safe reduction of $\mathrm{gtp}^{\mathfrak{A}}(\mathfrak{p}(c))$.*

(c5) *Take a 2-type β realized in $\mathfrak{C}^{\bar{\alpha}}$ by a pair c_1, c_2. If β is realized in a subcomponent then the claim follows by the inductive assumption applied to this substructure and the tree-like structure of $\mathfrak{C}^{\bar{\alpha}}$. If it joins an element from one subcomponent with its witness in another subcomponent then this 2-type is explicitly taken as a copy of a 2-type from $\mathfrak{A}_0$ (cf. also (c2)). Otherwise, the only positive non-unary atoms it may contain are equivalences added in one of the steps of taking transitive closures. Let $\mathcal{E}$ be the set of all equivalences belonging to β, and let α' be the 1-type of c_2. By (c4) $\mathrm{gtp}^{\mathfrak{C}^{\alpha}}(c_1)$ is a safe reduction of $\mathrm{gtp}^{\mathfrak{A}}(\mathfrak{p}(c_1))$, which means that $\alpha' \in \mathrm{gtp}^{\mathfrak{A}}(\mathfrak{p}(c_1)).\mathfrak{f}(\mathcal{E})$. Thus there is an element $a \in A$ of 1-type α' such that $\mathfrak{p}(c_1)$ is joined with a by all equivalences from $\mathcal{E}$. Observe now that $\mathrm{atp}^{\mathfrak{A}}(\mathfrak{p}(c_1), a)$ agrees with β on the 1-types it contains and contains all equivalences which are present in β. So the claim follows.*

(c6) *Follows directly from our construction.*

(c7) *Recall that level L_{i+1} contains witnesses for elements from L_i, but each such element is joined with its witness by a 2-type not containing E_i; any path from the root to a leaf must go through all levels, thus for each equivalence E_j, $1 \leq j \leq l$, there is a pair of consecutive elements on this path, not joined by E_j.*

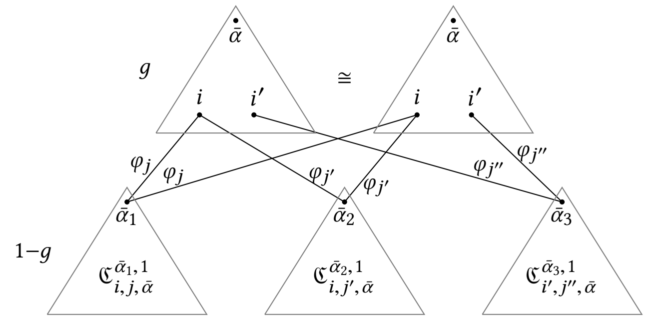

Figure 2. Joining the components.

3.2 Joining the tree-like components

In this step we are going to arrange a number of copies of our pattern tree-like components to obtain the desired structure $\mathfrak{B}_0$. We explicitly connect leaves of components with the roots of other components. We do it carefully, avoiding modifications to the internal structure of components, which could potentially result from transitivity of relations from $\sigma_{\rm dist}$. In particular, a pair of elements which are not connected by an equivalence $E_i \in \mathcal{E}_0$ in $\mathfrak{C}$ will not become connected by a chain of E_i-connections external to $\mathfrak{C}$.

Let *max* be the maximal number of elements in levels L_{l+1} over all pattern components constructed for types from $\bar{\boldsymbol{\alpha}}[A_0]$. For every $\bar{\alpha} \in \bar{\boldsymbol{\alpha}}[A_0]$ we take isomorphic copies $\mathfrak{C}^{\bar{\alpha},g}_{i,j,\bar{\alpha}'}$ of $\mathfrak{C}^{\bar{\alpha}}$, for $g = 0, 1$ (we will call g the *color* of a component), $i = 1, \ldots, max$, $j = 1, \ldots, m$, and every $\bar{\alpha}' \in \bar{\boldsymbol{\alpha}}[A_0]$. This constitutes the universe of a structure $\mathfrak{B}^+_0$, together with partially defined structure (on the copies of pattern components). A substructure of $\mathfrak{B}^+_0$ will be later taken as $\mathfrak{B}_0$. We import the values of $\mathfrak{p}$ from $\mathfrak{C}^{\bar{\alpha}}$ to all its copies. Let us denote the copy of element $c^{\bar{\alpha}}_s$ from $\mathfrak{C}^{\bar{\alpha},g}_{i,j,\bar{\alpha}'}$ as $c^{\bar{\alpha},g}_{s,(i,j,\bar{\alpha}')}$.

Our strategy is now as follows: if necessary, the root of $\mathfrak{C}^{\bar{\alpha}^*,g}_{i,j,\bar{\alpha}}$ will serve as a witness of type $\bar{\alpha}^*$ for $\varphi_j(x,y)$ and the i-th element from level L_{l+1} of all copies of $\mathfrak{C}^{\bar{\alpha}}$ of color $(1-g)$.

Formally, for every element $c^{\bar{\alpha},g}_{s,(i',j',\bar{\alpha}')}$, for every $1 \leq j \leq m$ if $\mathfrak{p}(c^{\bar{\alpha}}_s)$ has a witness w for $\varphi_j(x,y)$ in $\mathfrak{A}_0$ then, denoting $\bar{\alpha}^* = \mathrm{atp}^{\mathfrak{A}}(w)$ and $\beta = \mathrm{atp}^{\mathfrak{A}}(\mathfrak{p}(c^{\bar{\alpha}}_s), w)$, we join $c^{\bar{\alpha},g}_{s,(i',j',\bar{\alpha}')}$ with the root of $\mathfrak{C}^{\bar{\alpha}^*,1-g}_{s,j,\bar{\alpha}}$ using β. See Fig. 2. Transitively close all equivalences. This finishes the definition of $\mathfrak{B}^+_0$.

Finally, we choose any component $\mathfrak{C}$ whose root is mapped by $\mathfrak{p}$ to a_0 and remove from $\mathfrak{B}^+_0$ all the components which are not accessible from $\mathfrak{C}$ in the *graph of components*, formed by joining a pair of components iff the root of one of them serves as a witness for a leaf of another. We take the structure restricted to the remaining components as $\mathfrak{B}_0$.

3.3 Correctness of the construction

Claim 3.5. *The process of joining the tree-like components does not change the previously defined internal structure of any component.*

Proof. Potential changes could result only from closing transitively the equivalences which join leaves of some components with their witnesses—the roots of other components. Recall that by Condition (c7) the root of a component is not connected by any equivalence to any leaf of this component and note first that this condition cannot

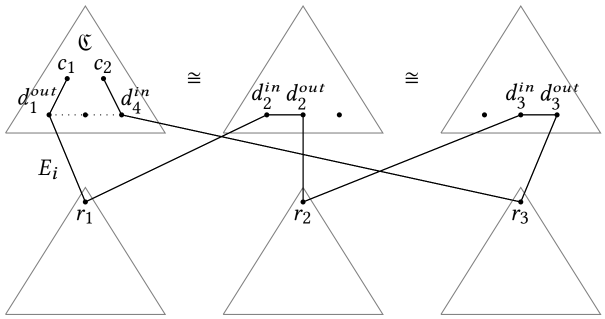

Figure 3. An E_i-path joining c_1 and c_2

be violated in the step of joining components. This is guaranteed by our strategy requiring leaves of components of color g to take as witnesses the roots of components of color $(1-g)$, for $g = 0, 1$.

Consider now any $E_i \in \mathcal{E}_0$ and elements c_1, c_2 belonging to the same component $\mathfrak{C}$. Assume that $\mathfrak{C} \not\models E_i(c_1, c_2)$, but $\mathfrak{B}_0 \models E_i(c_1, c_2)$. This means that during the process of providing witnesses for leaves, an E_i-path joining c_1 and c_2 was formed. Take such a path. Due to Condition (c7) such a path cannot enter a component through a leaf and leave it through the root. Thus, without loss of generality, we can assume that it is of the form $c_1, d_1^{out}, r_1, d_2^{in}, d_2^{out}, r_2, \ldots, d_{s-1}^{in}, d_{s-1}^{out}, r_{s-1}, d_s^{in}, c_2$, where the two elements of every pair (c_1, d_1^{out}), (d_2^{in}, d_2^{out}), …, (d_s^{in}, c_2) are members of the same component, all $d_i^{(\cdot)}$ are leaves, and each r_i is the root of a component used as a witness for d_i^{out} and d_{i+1}^{in}. See Fig. 3. Recalling our strategy, allowing a root to be used as a witness only for copies of the same leaf from some pattern component, we see that the components containing pairs (d_i^{in}, d_i^{out}), for $i = 2, \ldots, s-1$ and the component $\mathfrak{C}$ containing c_1 and c_2 are isomorphic to one another. Mapping isomorphically the E_i-edges joining d_i^{in} with d_i^{out} to $\mathfrak{C}$ we see that c_1 and c_2 were already connected E_i in $\mathfrak{C}$. Contradiction. ◻

Now, Conditions (b1)–(b6) can be shown using arguments similar to the ones used in the proofs of the above claim and (c1)–(c6).

This finishes the proof of Lemma 3.3 and thus also the proof of the finite model property for UNFO^2+EQ.

3.4 Size of models and complexity of UNFO^2+EQ

To complete the proof of Thm. 3.1 we need to estimate the size of finite models produced by our construction. This can be done by forming a recurrence relation for T_l—an upper bound on the size of structure $\mathfrak{B}_0$ constructed in the proof of Lemma 3.3 for $l_0 = l$. Note that the size of our final model is bounded by T_k. It is not difficult to see that $T_1 = 1$, and $T_{l+1} \leq twoexp(|\varphi|)T_l^{l+1}$, for a doubly exponential function $twoexp$. Solving this we get a doubly exponential bound on $|T_k|$.

We conclude this section with the following observation.

Theorem 3.6. *The satisfiability (= finite satisfiability) problem for UNFO^2+EQ is 2-ExpTime-complete.*

Proof. The upper bound follows from the finite model property and the upper bound for general satisfiability problem for UNFO+EQ formulated in Thm. 2.3. The lower bound can be shown by a routine adaptation of the proof of a 2-ExpTime-lower bound for the two-variable guarded fragment with two equivalence relations from [11]. A simple inspection of the properties needed to be expressed in that proof shows that they need only unary negations.

We also remark that a similar construction can be used to show that the doubly exponential upper bound on the size of models of satisfiable UNFO^2+EQ formulas is essentially optimal, that is UNFO^2+EQ it is possible to enforce models of at least doubly exponential size. ◻

4 Small model theorem for full UNFO+EQ

In this section we explain how to extend the small model theorem from the previous section to the case in which the number of variables is unbounded. The general approach is similar: given a pattern model we inductively rebuild it into a finite one. The first difference is that this inductive construction will be preceded by a pre-processing step producing from an arbitrary pattern model a model which has regular tree-like shape. Assuming such regularity will allow not only for a simpler description of the main construction, but, more importantly, for a simpler argument that the finite model we build satisfies part (2) from Lemma 2.4. Secondly, the number of levels of tree-like components we are going to construct needs to be increased with respect to the two-variable case. This time we not only require that the root of a component is not connected with any leaf by any (non-total) equivalence, but we ensure that there is no path from the root to a leaf built out of equivalence connection, on which the equivalences alternate less than t times (recall that t is the number of variables in the ∀-conjunct). The third difference we want to point out concerns the construction of witness structures. In the two-variable case a witness structure for a given element a and φ_i consisted of a and just one additional element and in the inductive process it was created at once. Now witness structures are bigger and in a single inductive step usually only a part of such structure is created (the part in which the appropriate equivalences are total), and the remaining part is completed in the higher levels of induction. Also, for simplicity, in a single step we will deal with φ-witness structures rather than with witness structures for various φ_i separately. Moreover, given a tree-like model we will speak about *the* φ-witness structure for an element, meaning the witness structure consisting of this element and its children, even if, accidentally, some other φ-witness structures for this element exist. Finally, generalized types from Section 3 will no longer be sufficient for our purposes and we will need some more complicated notions here.

4.1 Regular tree-like models

Lemma 4.1. *Every satisfiable UNFO normal form formula φ has a tree-like model $\mathfrak{A} \models \varphi$ with doubly exponentially many (with respect to $|\varphi|$) non-isomorphic subtrees.*

The proof starts from a tree-like model guaranteed by Lemma 2.5. Then, roughly speaking, some patterns which could possibly be extended to substructures falsifying the ∀-conjunct of φ are defined. A node of a tree-like model is assigned a *declaration*, that is the list of such patterns which do not appear in its subtree. We choose one node for every realized declaration and build a regular tree-like model out of copies of the chosen elements and their φ-witness structures. As the number of possible declarations is bounded doubly exponentially, the claim follows. We omit the details of the proof.

4.2 Main theorem

We are now ready to show the main result of this paper.

Theorem 4.2. *Every satisfiable UNFO+EQ formula φ has a model of size bounded doubly exponentially in $|\varphi|$.*

Let us fix a satisfiable normal form UNFO+EQ formula φ, and the finite relational signature $\sigma = \sigma_{\text{base}} \cup \sigma_{\text{dist}}$ consisting of all symbols appearing in φ. Enumerate equivalences as $\sigma_{\text{dist}} = \{E_1, \ldots, E_k\}$. Fix a regular tree-like σ-structure $\mathfrak{A} \models \varphi$ with at most doubly exponentially non-isomorphic subtrees, which exists due to Lemma 4.1. We show how to build a finite model of φ. We mimic the inductive approach and the main steps of a finite model construction for φ from the previous section. However, the details are more complicated.

Recall that in the two-variable case, we built our finite structure together with a function $\mathfrak{p}$ whose purpose was to assign to elements of the new model elements of the original model of similar generalized types. Intuitively, in the current construction the role of generalized types of elements will be played by the isomorphism types of subtrees of $\mathfrak{A}$.

To shorten notation we will denote by $[a]_E$ the E-equivalence class of an element a (the structure will be clear from the context). We denote by $\mathfrak{A}_a$ the subtree rooted at a (from now on such subtrees will be considered only in $\mathfrak{A}$). We state the counterpart of Lemma 3.3 as follows.

Lemma 4.3. *Let $\mathcal{E}_0 \subseteq \sigma_{dist}$, $\mathcal{E}_{tot} = \sigma_{dist} \setminus \mathcal{E}_0$, $E^* = \bigcap_{E_i \in \mathcal{E}_{tot}} E_i$, $a_0 \in A$, $\mathfrak{A}_0$ be the induced substructure of $\mathfrak{A}$ on $A_{a_0} \cap [a_0]_{E^*}$. Then there exists a finite structure $\mathfrak{A}'_0$, an element (called the root of $\mathfrak{A}'_0$) $a'_0 \in A'_0$ and a function $\mathfrak{f} : A'_0 \to A_0$ such that:*

(d1) $\mathfrak{f}(a'_0) = a_0$
(d2) E^* *is total in* $\mathfrak{A}'_0$
(d3) For each $a' \in A'_0$ it has a pre-φ-witness structure *in $\mathfrak{A}'_0$, that is an isomorphic copy of the restriction of the φ-witness structure for $\mathfrak{f}(a')$ in $\mathfrak{A}$ to A_0.*[2]
(d4) For each $a_1, \ldots, a_t =: \bar{a} \subseteq A'_0$ there exists a homomorphism $\mathfrak{h} : \bar{a} \to \mathfrak{A}_{a_0}$ such that for $1 \le i \le t$ we have $\mathfrak{A}_{\mathfrak{h}(a_i)} \cong \mathfrak{A}_{\mathfrak{f}(a_i)}$
(d5) Moreover if for some i $a_i = a'_0$ then $\mathfrak{h}$ can be chosen so that $\mathfrak{h}(a_i) = a_0$

The proof goes by induction on $l = |\mathcal{E}_0|$. The base of induction, $l = 0$, can be treated as in the two-variable case. For the inductive step, suppose that theorem holds for $l - 1$. We show that it holds for l. Without loss of generality let $\mathcal{E}_0 = \{E_1, \ldots, E_l\}$.

4.3 Components

In the two variable case we created a single type of a building block for the finite model construction for every generalized type realized in substructure $\mathfrak{A}_0$ of the original model. Now we create one type of a building block for every isomorphism type of a subtree rooted at a node of $\mathfrak{A}_0$. We denote by $\boldsymbol{\gamma}[A_0]$ the set of such isomorphism types. Let γ_{a_0} be the type of $\mathfrak{A}_{a_0}$.

Take $\gamma \in \boldsymbol{\gamma}[A_0]$ and the root $a \in A_0$ of a subtree of type γ. We explain how to construct a finite *tree-like component* (or shortly: a *component*) $\mathfrak{C}^\gamma$. The main steps of this construction are similar to the ones in the two-variable case. The tree-like component will have $tl + 1$ levels $L_1, \ldots, L_{tl+1}$. We start the construction of level L_i defining its initial part, L_i^{init}, and then expand it to full level.

[2]Note that this condition implies that $\mathfrak{f}$ preserves 1-types ($\text{atp}^{\mathfrak{A}'_0}(a) = \text{atp}^{\mathfrak{A}_0}(\mathfrak{f}(a))$)

We set $L_1^{init} = \{a'\}$ to consist of a copy of element a, i.e., we set $\text{atp}^{\mathfrak{C}^\gamma}(a') := \text{atp}^{\mathfrak{A}_0}(a)$. Put $\mathfrak{f}(a') = a$. We call a' the root of $\mathfrak{C}^\gamma$.

Suppose we have defined levels $L_1, \ldots, L_{i-1}$ and L_i^{init}, and the structure and the values of $\mathfrak{f}$ on $L_1 \cup \ldots \cup L_{i-1} \cup L_i^{init}$. We now explain how to define L_i and L_{i+1}^{init}. Let $s = 1 + (i - 1 \mod l)$.

Step 1: Subcomponents. Take any element $c \in L_i^{init}$. From the inductive assumption we have a structure $\mathfrak{B}_0$ with $E^* \cap E_s$ total on it, its root $b_0 \in B_0$ and a function $\mathfrak{f}_b : B_0 \to A_{\mathfrak{f}(c)} \subseteq A_0$ with $\mathfrak{f}_b(b_0) = \mathfrak{f}(c)$. The substructures obtained owing to the inductive assumption are called *subcomponents*. We identify b_0 with c, add isomorphically $\mathfrak{B}_0$ to L_i, extending function $\mathfrak{f}$ so that $\mathfrak{f} \restriction B_0 = \mathfrak{f}_b$. We do this independently for all $c \in L_i^{init}$. In contrast to the two-variable case we do not apply the transitive closure to relations from σ_{dist} at this moment. We remark, that taking the transitive closures would not affect the correctness of the construction, but not doing this will allow us for a simpler presentation of the correctness proof.

Step 2: Providing witnesses. This step is slightly different compared to its two-variable counterpart. For $i < tl + 1$ we now define L_{i+1}^{init}. Take $c \in L_i$. Let $\mathfrak{D}$ be the φ-witness structure for $\mathfrak{f}(c)$ in $\mathfrak{A}$. Let $\mathfrak{E}$ be the restriction of $\mathfrak{D}$ to $[\mathfrak{f}(c)]_{E^* \cap E_s}$. Let $\mathfrak{F}$ be the isomorphic copy of $\mathfrak{E}$ created for c in $\mathfrak{B}_0$, which exists due to (d3). Let $\mathfrak{E}' = \mathfrak{D} \restriction [\mathfrak{f}(c)]_{E^*}$. We add F''—a copy of $E' \setminus E$ to L_{i+1}^{init}, and isomorphically copy the structure of $\mathfrak{E}'$ to $F \cup F''$. Note that this operation is consistent with the previously defined structure on $\mathfrak{F}$. We set $\mathfrak{f} \restriction F''$ naturally choosing isomorphic copies from $E \setminus E'$ of the newly added elements. We repeat this step independently for all for all $c \in L_i$.

When L_{tl+1} is created the construction of $\mathfrak{C}^\gamma$ is completed.

4.4 Joining the components

As in the case of UNFO^2+EQ, this step consists in joining some leaves with some roots of components. To deal with the additional condition (d5) we will simply define a'_0 in such a way that it will not be used as a witness for any leaf. As promised above we create components for all types from $\boldsymbol{\gamma}[A_0]$. They are called *pattern components*. Let *max* be the maximal number of leaves over all pattern components, and let max_w be the maximal size of a φ-witness structure used in pattern components. We create components $\mathfrak{C}^{\gamma,g}_{i,j,\gamma'}$ for all $\gamma, \gamma' \in \boldsymbol{\gamma}[A_0]$, $g \in \{0, 1\}$ (g is sometimes called a *color* of the component), $1 \le i \le max$, $1 \le j \le max_w$, as isomorphic copies of $\mathfrak{C}^\gamma$. We also create an additional component $\mathfrak{C}^{\gamma_{a_0},0}_{\perp,\perp,\perp}$ as a copy of $\mathfrak{C}^{\gamma_{a_0}}$, and define a'_0 to be its root.

Now we provide a φ-witness structure for the i-th leaf c_i of any $\mathfrak{C}^{\gamma,g}_{(\cdot,\cdot,\cdot)}$. Take the φ-witness structure for $\mathfrak{f}(c_i)$ in $\mathfrak{A}$ consisting of the successors of $\mathfrak{f}(c_i)$ and let $\mathfrak{D}$ be its restriction to $\mathfrak{A}_0$. Let $\gamma_1, \ldots, \gamma_u$ be the types of subtrees rooted at elements of this restriction other than $\mathfrak{f}(c_i)$. Isomorphically copy $\mathfrak{D}$ choosing as witnesses the roots of structures $\mathfrak{C}^{\gamma_j,1-g}_{i,j,\gamma}$.

Finally, we take as $\mathfrak{A}^0_0$ the structure restricted to the components accessible in the graph of components from $\mathfrak{C}^{\gamma_{a_0},0}_{\perp,\perp,\perp}$. Recall that the graph of components is formed by joining a pair of components iff the root of one of them serves as a witness for a leaf of another.

We now define $\mathfrak{A}'_0$ as a copy of $\mathfrak{A}^0_0$ with transitively closed equivalences. Recall that in structure $\mathfrak{A}^0_0$ we, exceptionally, do not transitively close σ_{dist}-connections, and thus allow the interpretations of the symbols from σ_{dist} not to be transitive (we will keep using superscript 0 for auxiliary structures of this kind).

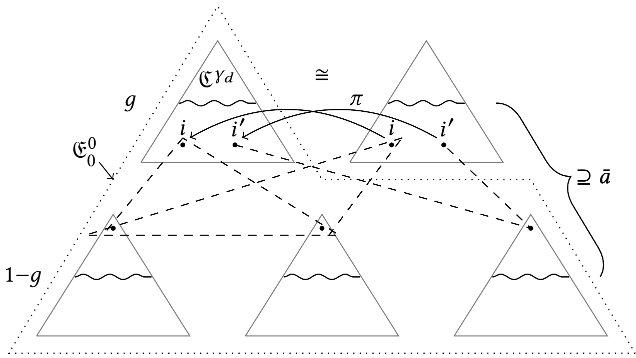

Figure 4. Reductions 1 and 2

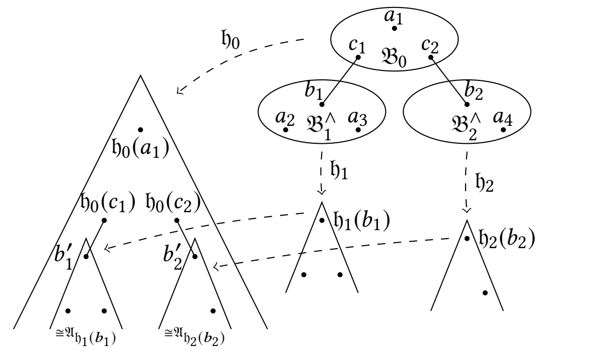

Figure 5. Joining homomorphisms

4.5 Correctness of the construction

Now we proceed to the proof that $\mathfrak{A}_0'$ satisfies conditions (d1)–(d5):

(d1) Follows directly from definition of L_1^{init} in $\mathfrak{C}^{\gamma_{a_0}}_{\bot,\bot,\bot} \subseteq \mathfrak{A}_0'$.

(d2) In every step in which a substructure of $\mathfrak{A}_0^0$ is defined, E^* is total in it. Then we choose a connected fragment of A_0^0 and close in it E^* transitively. Thus in the obtained structure $\mathfrak{A}_0'$ E^* is total.

(d4) This is the key part of our argumentation. The proof consists of several homomorphic reductions performed in order to show that we can restrict attention to a structure built as a tree-like component but twice as high.

Reduction 0. Take $\bar{a} \subseteq A_0'$. Let $\bar{a}_1, \bar{a}_2, \dots, \bar{a}_K$ be the connected fragments contained in $\bar{a}$ in the Gaifman-graph of $\mathfrak{A}_0^0$ with eliminated $\mathcal{E}_{tot}$-relations. If we have homomorphisms $\mathfrak{h}_i : \bar{a}_i \to \mathfrak{A}_0$, it is sufficient to put $\mathfrak{h} = \bigcup \mathfrak{h}_i$ as the desired homomorphism, since E^* is total on $\mathfrak{A}_0$ and for $a \in \bar{a}_i$ we also have $\mathfrak{A}_{\mathfrak{h}(a)} = \mathfrak{A}_{\mathfrak{h}_i(a)} \cong \mathfrak{A}_{\mathfrak{f}(a)}$. So we can restrict attention to connected $\bar{a}$.

Reduction 1. Since the choice of the number of levels in components is such that in each component there is no path from the root to a leaf having less than t alternations of equivalences, $\bar{a}$ is contained "on the boundary of two layers". See Fig. 4. More precisely, there is $g \in \{0,1\}$ such that there is no equivalence path in structure $\mathfrak{A}_0^0$ joining two elements in $\bar{a}$ which contains some connection between a root of color g and a leaf of color $1-g$. Let $\mathfrak{D}_0^0$ be a structure obtained from $\mathfrak{A}_0^0$ by removing all direct connections between roots of color g and leaves of color $1-g$ and $\mathfrak{D}_0'$ its minimal extension in which equivalences are transitively closed. We have just proved that the identity $\mathrm{id} : \bar{a} \to \mathfrak{D}_0'$ is a homomorphism, and since for all $a \in \bar{a}$, $\mathfrak{A}_{\mathfrak{f}(a)} = \mathfrak{A}_{\mathfrak{f}(\mathrm{id}(a))}$, we can restrict attention to $\bar{a} \subseteq \mathfrak{D}_0'$.

Reduction 2. Note that there is at most one type γ_d of components of color g chosen in the previous reduction containing some element from $\bar{a}$. See Fig. 4. Now we can naturally "project" all elements of $\bar{a}$ of color g on one chosen component $\mathfrak{C}^{\gamma_d} := \mathfrak{C}^{\gamma_d, g}_{(i,j,\gamma_e)}$. Call this projection π. Then we remove from $\mathfrak{D}_0^0$ all components of color g other than $\mathfrak{C}^{\gamma_d}$ and all components of color $1-g$ of form other than $\mathfrak{C}^{\cdot, 1-g}_{\cdot, \cdot, \gamma_d}$ obtaining structure $\mathfrak{E}_0^0$. Let $\mathfrak{E}_0'$ be created by closing transitively all equivalences in $\mathfrak{E}_0^0$. We claim that π is a homomorphism from $\bar{a}$ to $\mathfrak{E}_0'$. Indeed such projection can be applied to paths in $\mathfrak{D}_0^0$ to get corresponding paths in $\mathfrak{E}_0^0$. Since for all $a \in \bar{a}$ we have $\mathfrak{A}_{\mathfrak{f}(a)} = \mathfrak{A}_{\mathfrak{f}(\pi(a))}$, we may restrict attention to $\bar{a} \subseteq \mathfrak{E}_0'$.

Final step. From the construction of $\mathfrak{A}_0'$ we can see that $\mathfrak{E}_0^0$ can be considered as a component of height $2(tl+1)$ and such component can be viewed, slightly informally speaking, as a tree T whose nodes are subcomponents. We will build a homomorphism $\mathfrak{h} : \bar{a} \to \mathfrak{A}_{a_0}$ inductively using a bottom-up approach on tree T. For a subcomponent $\mathfrak{B}$ denote $\mathfrak{B}^\wedge$ the substructure of $\mathfrak{E}_0'$ being its restriction to the set of all elements of components belonging to the subtree rooted at $\mathfrak{B}$ in T.

Claim 4.4. *For every subcomponent $\mathfrak{B}_0 \in T$ with root b_0 and for every $\bar{a} \subseteq \mathfrak{B}_0^\wedge$, $|\bar{a}| \leq t$ there exists a homomorphism $\mathfrak{h} : \bar{a} \to \mathfrak{A}_{\mathfrak{f}(b_0)}$ such that for all $a \in \bar{a}$, $\mathfrak{A}_{\mathfrak{f}(a)} \cong \mathfrak{A}_{\mathfrak{h}(a)}$ and if $b_0 \in \bar{a}$ then $\mathfrak{h}(b_0) = \mathfrak{f}(b_0)$.*

Proof. Bottom-up induction on subtrees.

Base of induction. If $\bar{a} \subseteq \mathfrak{B}_0$ the claim follows from the inductive assumption of Lemma 4.3.

Inductive step. Let $\mathfrak{B}_1, \dots, \mathfrak{B}_K$ be the list of those children of $\mathfrak{B}_0$ in T for which $\mathfrak{B}_i^\wedge$ contains some elements from $\bar{a}$; denote by b_i the root of $\mathfrak{B}_i$ and let $c_i \in B_0$ be such that b_i is a witness chosen by c_i in the step of providing witnesses/joining components. If $K = 0$ we are done (see the proof of the base of induction). If $K = 1$ and $\bar{a} \subseteq \mathfrak{B}_1^\wedge$ the thesis follows from the inductive assumption of this claim. Otherwise, by the inductive assumption of this claim applied to $(\bar{a} \cap \mathfrak{B}_i^\wedge) \cup \{b_i\}$ we have homomorphisms $\mathfrak{h}_i : (\bar{a} \cap \mathfrak{B}_i^\wedge) \cup \{b_i\} \to \mathfrak{A}_{\mathfrak{f}(b_i)}$ and from the inductive assumption of Lemma 4.3 a homomorphism $\mathfrak{h}_0 : (\bar{a} \cap \mathfrak{B}_0) \cup \{c_i : i \leq K\} \to \mathfrak{A}_{\mathfrak{f}(b_0)}$. Because $\mathfrak{A}$ is a regular model, homomorphisms $\mathfrak{h}_0, \dots, \mathfrak{h}_K$ can be joined together into $\mathfrak{h} : \bar{a} \cup \{b_i : i \leq K\} \cup \{c_i : i \leq K\} \to \mathfrak{A}_{\mathfrak{f}(b_0)}$ (see Fig. 5) in such a way that for all $a \in \mathrm{dom}(\mathfrak{h}_i)$ we have $\mathfrak{A}_{\mathfrak{h}(a)} \cong \mathfrak{A}_{\mathfrak{h}_i(a)} \cong \mathfrak{A}_{\mathfrak{f}(a)}$ (the last isomorphism follows from the inductive assumption of this claim and Lemma 4.3). Indeed, when joining $\mathfrak{h}_0$ with $\mathfrak{h}_i$ we can see that some child b_i' of $\mathfrak{f}(c_i)$ satisfies $\mathfrak{A}_{\mathfrak{f}(b_i)} \cong \mathfrak{A}_{b_i'}$ and we can treat $\mathfrak{h}_i$ as a homomorphism into $\mathfrak{A}_{b_i'}$. It follows from the construction that $\mathfrak{h}$ has the following property: if $b_0 \in \bar{a}$ then $\mathfrak{h}(b_0) = \mathfrak{h}_0(b_0) = \mathfrak{f}(b_0)$. Recalling the tree-like structure on T we get that $\mathfrak{h}$ is a homomorphism. To finish the proof of the inductive step, we restrict $\mathfrak{h}$ to $\bar{a}$. □

(d5) Easily follows from inspection of the proof of (d3).

(d3) It is taken care explicitly during the step of providing witnesses (non-leaves) and the step of joining components (leaves). There are no additional connections inside witness structures, since equivalence classes for $E \in \mathcal{E}_0$, as in (d4) part of the proof, lie "on boundary of two colors"' and, similarly as in the proof of (d4) one can consider the graph on subcomponents. The approach here is similar to one employed in the proof of (c2) and Claim 3.5.

Let us show, how Lemma 4.3 implies the finite model property for UNFO+EQ. Take $\mathcal{E}_0 = \sigma_{\text{dist}}$, a_0 - the root of $\mathfrak{A}$. We apply Lemma 4.3 and get a structure $\mathfrak{A}'_0$ and a function $\mathfrak{f} : A'_0 \to A_0$. Note, that $\mathfrak{A}_0 = \mathfrak{A}$. Let us see that $\mathfrak{A}'_0$ satisfies the conditions of Lemma 2.4. Indeed, (1) follows from (d3). Condition (2) follows from (d3) (1-type remark) and (d4). So $\mathfrak{A}'_0 \models \varphi$.

4.6 Size of models and complexity

As in the two-variable case we can show that the size of our final model is bounded doubly exponentially in the size of the formula. The finite model property and Thm. 2.3 allow us to conclude.

Theorem 4.5. *The finite satisfiability problem for UNFO+EQ is 2-ExpTime-complete.*

5 Towards guarded negation with equivalences

We observe now that our small model construction can be adapted even for a slightly bigger logic. The *guarded negation fragment* of first-order logic, GNFO, is defined in [3] by the following grammar:

$$\varphi = R(\bar{x}) \mid x = y \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \exists x \varphi \mid \gamma(\bar{x}, \bar{y}) \wedge \neg\varphi(\bar{y}),$$

where γ is an atomic formula. Since equality statements of the form $x = x$ can be used as guards, GNFO may be viewed as an extension of UNFO. However, the satisfiability problem for GNFO with equivalences is undecidable. It follows from the fact that even the two-variable guarded fragment, which is contained in GNFO, becomes undecidable when extended by equivalences [11].

To regain decidability we consider the *base-guarded negation fragment with equivalences*, BGNFO+EQ, analogous to the base-guarded negation fragment with transitive relations, BGNFO+TR, investigated in [1]. In these variants all guards must belong to σ_{base}, and all symbols from σ_{dist} must be interpreted as equivalences/transitive relations. Recall that the general satisfiability problem for BGNFO+TR was shown decidable in [1], and as explained in Section 2.4 this implies decidability of the general satisfiability problem for BGNFO+EQ. In this paper we do not solve the finite satisfiability problem for full BGNFO+EQ. We however solve this problem for its one-dimensional restriction.

We say that a first-order formula is *one-dimensional* if its every maximal block of quantifiers leaves at most one variable free. E.g., $\neg\exists yzR(x, y, z))$ is one-dimensional, and $\neg\exists zR(x, y, z))$ is not. By *one-dimensional guarded negation fragment*, GNFO_1 we mean the subset of GNFO containing its all one-dimensional formulas. Not all UNFO formulas are one-dimensional, but they can be easily converted to the already mentioned UN-normal form [21], which contains only one-dimensional formulas. The cost of this conversion is linear. This allows us to view UNFO as a fragment of GNFO_1.

We can define the one-dimensional restriction BGNFO_1+EQ of BGNFO+EQ in a natural way. We note that moving from UNFO+EQ to BGNFO_1+EQ significantly increases the expressive power. An example formula which is in BGNFO_1+EQ but is not expressible in UNFO+EQ is $\neg\exists xy(R(x, y) \wedge \neg E_1(x, y))$, which says that $R \subseteq E_1$. Observe, however, that since guards must belong to σ_{base} we are not able to express the containment of one equivalence relation in another equivalence, or in a relation from σ_{base}.

Our proof from Section 4 can be adapted to cover the case of BGNFO_1+EQ. The adaptation is not difficult. What is crucial is that in the current construction, during the step of providing witnesses, we build isomorphic copies of whole witness structures, which means that we preserve not only positive atoms but also their negations. Thus, we preserve witness structures for BGNFO_1+EQ.

Theorem 5.1. *BGNFO_1+EQ has a doubly exponential finite model property, and its satisfiability (= finite satisfiability) problem is 2-ExpTime-complete.*

6 Conclusion

We proved the finite model property for UNFO with equivalence relations and for the one-dimensional restriction of GNFO with equivalences outside guards. There are two natural directions in which our work can be extended. Currently we are working on the finite satisfiability problem for UNFO with arbitrary transitive relations. Note that it lacks the finite model property fails, as the formula $\forall x \exists y T(x, y) \wedge \forall x \neg T(x, x)$, with transitive T, is satisfiable only in infinite models. Another open question is the decidability of finite satisfiability of full BGNFO+EQ.

Acknowledgments

This work is supported by Polish National Science Centre grant No 2016/21/B/ST6/01444.

References

[1] A. Amarilli, M. Benedikt, P. Bourhis, and M. Vanden Boom. 2016. Query Answering with Transitive and Linear-Ordered Data. In *International Joint Conference on Artificial Intelligence, IJCAI 2016*. 893–899.
[2] H. Andréka, J. van Benthem, and I. Németi. 1998. Modal Languages and Bounded Fragments of Predicate Logic. *J. Philosophical Logic* 27 (1998), 217–274.
[3] V. Bárány, B. ten Cate, and L. Segoufin. 2015. Guarded Negation. *J. ACM* 62, 3 (2015), 22.
[4] M. Bojańczyk, C. David, A. Muscholl, T. Schwentick, and L. Segoufin. 2011. Two-variable logic on data words. *ACM Trans. Comput. Log.* 12, 4 (2011), 27.
[5] M. Bojanczyk, A. Muscholl, T. Schwentick, and L. Segoufin. 2009. Two-variable logic on data trees and XML reasoning. *J. ACM* 56, 3 (2009).
[6] M. Dzieciolowski. 2017. *Satisfability issues for unary negation logic*. Bachelor's Thesis. University of Wrocław.
[7] E. Grädel, P. Kolaitis, and M. Y. Vardi. 1997. On the decision problem for two-variable first-order logic. *B. Symb. Log.* 3, 1 (1997), 53–69.
[8] L. Hella and A. Kuusisto. 2014. One-dimensional Fragment of First-order Logic. In *Advances in Modal Logic, AIML 2014)*. 274–293.
[9] J. Ch. Jung, C. Lutz, M. Martel, and T. Schneider. 2018. Querying the Unary Negation Fragment with Regular Path Expressions. In *International Conference on Database Theory, ICDT 2018*. 15:1–15:18.
[10] Y. Kazakov. 2006. *Saturation-based decision procedures for extensions of the guarded fragment*. Ph.D. Dissertation. Universität des Saarlandes, Saarbrücken, Germany.
[11] E. Kieroński. 2005. Results on the Guarded Fragment with Equivalence or Transitive Relations. In *Computer Science Logic (LNCS)*, Vol. 3634. Springer, 309–324.
[12] E. Kieroński, J. Michaliszyn, I. Pratt-Hartmann, and L. Tendera. 2014. Two-variable first-order logic with equivalence closure. *SIAM J. Comput.* 43, 3 (2014), 1012–1063.
[13] E. Kieroński and M. Otto. 2012. Small Substructures and Decidability Issues for First-Order Logic with Two Variables. *J. Symb. Log.* 77 (2012), 729–765.
[14] E. Kieroński and L. Tendera. 2018. Finite Satisfiability of the Two-Variable Guarded Fragment with Transitive Guards and Related Variants. *ACM Trans. Comput. Logic* 19, 2, Article 8 (2018), 8:1–8:34 pages.
[15] A. Montanari, M. Pazzaglia, and P. Sala. 2016. Adding one or more equivalence relations to the interval temporal logic. *Theor. Comput. Sci.* 629 (2016), 116–134.
[16] M. Mortimer. 1975. On languages with two variables. *Zeitschr. f. Math. Logik und Grundlagen d. Math.* 21 (1975), 135–140.
[17] I. Pratt-Hartmann. 2018. The Finite Satisfiability Problem for Two-Variable, First-Order Logic with one Transitive Relation is Decidable. *To appear in Math. Log. Q.* (2018).
[18] I. Pratt-Hartmann, Wieslaw Szwast, and Lidia Tendera. 2016. Quine's Fluted Fragment is Non-Elementary. In *Computer Science Logic, CSL 2016*. 39:1–39:21.
[19] D. Scott. 1962. A decision method for validity of sentences in two variables. *J. Symb. Log.* 27 (1962), 477.
[20] W. Szwast and L. Tendera. 2004. The guarded fragment with transitive guards. *Ann. Pure Appl. Logic* 128 (2004), 227–276.
[21] B. ten Cate and L. Segoufin. 2013. Unary negation. *Log. Meth. Comput. Sci.* 9, 3 (2013).