Two-variable Logic with Counting and Trees

Witold Charatonik, Piotr Witkowski
Institute of Computer Science
University of Wroclaw, Poland
Email: \{wch,pwit\}@ii.uni.wroc.pl

Abstract—We consider the two-variable logic with counting quantifiers (C²) interpreted over finite structures that contain two forests of ranked trees. This logic is strictly more expressive than standard C² and it is no longer a fragment of the first order logic. In particular, it can express that a structure is a ranked tree, a cycle or a connected graph of bounded degree. It is also strictly more expressive than the first-order logic with two variables and two successor relations of two finite linear orders. We give a decision procedure for the satisfiability problem for this logic. The procedure runs in NEXPTIME, which is optimal since the satisfiability problem for plain C² is NEXPTIME-complete.

I. INTRODUCTION

A. Two-variable logics

The first-order logic with two variables (for short, the two-variable logic, FO²) is, besides the guarded fragment, the most prominent decidable fragment of first-order logic. It is important in computer science because of its decidability and connections with other formalisms like modal, temporal or description logics or applications in XML or ontology reasoning.

The satisfiability of the two-variable logic was proved to be decidable in [23], [17] and NEXPTIME-complete in [7]. Since the fragment has a limited expressive power, a lot of effort was put to extend it beyond the first-order logic while preserving decidability.

Many extensions of this logic quickly lead to undecidability [9], [10]. One of few known decidable extensions is the two-variable fragment with counting quantifiers [8], but this is still a fragment of the first-order logic. Decidable extensions that go beyond the first-order logic include FO² over restricted classes of structures where some relation symbols are interpreted as orders or equivalence relations [18], [13], [3], [14], [22] or FO² with restricted transitive closure [11] or equivalence closure [12].

The two-variable fragment with counting quantifiers (C²) is the set of first-order formulas with at most two variables, but with the counting quantifiers of the form \( \exists^k \), \( \exists^\geq k \) or \( \exists^= k \). The (finite) satisfiability problem for C² is the problem of whether a given C² formula has a (finite) model. The two problems of satisfiability and finite satisfiability for C² (which are two different problems as C² does not have a finite model property) were proved to be decidable in [8]. Another solution of the satisfiability problem together with NEXPTIME-completeness result under unary encoding of numbers in counting quantifiers can be found in [19]. Pratt-Hartmann in [20] established NEXPTIME-completeness of both satisfiability and finite satisfiability under binary encoding of numbers in counting quantifiers. The algorithms used in proving upper bounds for C² in these papers are quite sophisticated, a significant simplification can be found in [21].

B. Our contribution

In this paper we study the two-variable logic with counting quantifiers interpreted over finite structures that contain two (possibly empty) forests of ranked trees of degree bounded by a given number \( d \). This logic is strictly more expressive than standard C² and it is no longer a fragment of the first order logic. It allows to express that the underlying structure is a ranked tree or that it is or it contains a finite set of disjoint ranked trees.

The main result of this paper is the algorithm that decides satisfiability problem for the extended logic. The method we use is an extension of the one from [21]. We simplified a bit the technique used there and extended it to handle structures where distinguished predicate symbols are restricted to represent trees. The algorithm runs in NEXPTIME, which is optimal since the satisfiability problem for plain C² is NEXPTIME-complete.

Our original motivation stems from the area of program verification. In [4] the authors define a decidable extension of the universal fragment of the two-variable fragment of the first-order logic (FO²) with datalog programs and use it in a bounded model-checking procedure for programs manipulating dynamic pointer structures. The idea is that the universal two-variable fragment can express error conditions and is closed under weakest preconditions, and the extension with datalog can express recursively defined data structures on heap like lists or trees. In [5] we extended the logic introduced in [4] to C² with datalog and proved NEXPTIME-completeness of the obtained satisfiability problem. The decidability proof for C² with datalog was done by a reduction to C²—this shows that the extended logic was only a syntactic extension of C² and consequently still a fragment of the first-order logic, and that C² is expressive enough to encode recursive data structures (there is a subtlety here: C² cannot express that a given structure is a tree but it can express that a substructure rooted at a given point is a tree, which is enough to model trees on a heap). With the current extension we have strictly more expressive logic: not only universal fragment, but also counting quantifiers, and much more expressive power in modeling data.
structures — using two forests one can express e.g. trees whose leaves are joined in a single list.

Although originally motivated by verification, we believe the result is of independent interest because of many applications of two-variable logics. Our logic is the first decidable extension of \( C^2 \) that we are aware of and that goes beyond the first-order logic.

C. Related work

We use the technique introduced by Pratt-Hartmann in [20], [21]. A more detailed comparison of similarities and differences between his and our approach is given in Section III-A.

Our research falls into the area of extensions of the two-variable fragment with signatures containing special binary relations, which has been extensively studied recently. It is known that the finite satisfiability problem is undecidable for \( FO^2 \) with two transitive relations [11], with three equivalence relations [13], with one transitive and one equivalence relation [14], with three linear orders [10]. The problem is decidable in \( 3NEXPTIME \) for \( FO^2 \) over words [3] or trees [2] with one equivalence relation; it is \( NEXPTIME \)-complete with one linear order and one equivalence relation [3], \( EXPSPACE \)-complete with two linear orders [22].

The work that is closest to our result is on \( FO^2 \) with two successor relations on two linear orders [16], [6]. The problem was claimed to be decidable in \( 2NEXPTIME \) in [16], but, as explained in [6], the proof there was flawed; the main result in [6] shows that the problem is in \( NEXPTIME \). The same problem with three (or more) successor relations of finite linear orders remains open. Our result subsumes the one in [6]. First, we have not only \( FO^2 \), but also \( C^2 \). Second, we allow signatures with an arbitrary number of binary relations, not only the two special ones. Third, we have not only linear orders, but arbitrary forests (the two successor relations from [6] correspond to two total trees of degree one).

Complexity of two-variable logics over trees was recently studied in [1], but the results there are incomparable with ours. On one hand, for similar reasons as above, namely lack of arbitrary binary predicates and counting quantifiers as well as inability to express forests. On the other hand, the logics in [1] work on unranked trees and may use transitive closure of the successor relation and the sibling relation, which is not expressible in our case.

II. Preliminaries

We start by giving a few examples showing the expressive power of our logic.

A. Expressive power

It is well known (see e.g., Corollary 3.19 in [15]) that first-order logic cannot express connectivity of graphs. Similarly, it is not possible (see e.g., Proposition 3.20 in [15]) to test by a first-order sentence if a graph is a tree. In fact, this even not possible for graphs of degree bounded by 2. The following examples show that \( C^2 \) interpreted over classes of structures with forests can express some non first-order properties. The formula \( forest(root,\ tree,\ edge) \) used in the examples below is formally defined in Section III-B; it is used there for a different purpose, here we only use it to make sure that the unary predicates \( root \) and \( tree \) are interpreted as roots and nodes of a forest defined by the binary predicate \( edge \).

Example 1. Consider an undirected graph \( G \) with nodes of degree bounded by \( d \) and edges represented by a symmetric relation \( E(\cdot,\cdot) \). Such a graph can be defined by the formula \( \varphi_G = \forall x\forall y.(E(x,y) \leftrightarrow E(y,x)) \land \forall x\exists y.E(x,y) \). Consider the conjunction of \( \varphi_G \) with \( forest(root,\ tree,\ edge) \) and the formula \( \exists^=1x.root(x) \land \forall x.tree(x) \land \forall x,y.(edge(x,y) \rightarrow E(x,y)) \). This formula interpreted in the class of structures where the predicate \( edge \) defines a forest of degree \( d \) expresses that \( G \) has a spanning tree, which is equivalent to \( G \) being connected.

By using more than one root we may also express that a graph is disconnected; note however that this is not a simple negation of the formula expressing connectedness.

Example 2. Consider a directed simple graph \( G \) with edges represented by a relation \( E \) and a \( C^2 \) formula \( \exists^=1x.root(x) \land \forall x.tree(x) \land \forall x,y.(edge(x,y) \rightarrow E(x,y)) \). This formula in conjunction with \( forest(root,\ tree,\ edge) \) expresses that \( G \) is a rooted directed tree (of degree bounded by \( d \)). Additional conjunct \( \forall x.\exists^=1y.edge(x,y) \) would say that each node in the tree has degree at most 3. In the same way one may express that each non-leaf node has the same degree (like in binary trees) or has a lower bound on its degree (like in 2-3-trees). Similarly one can express that the structure is a path or a cycle.

Our next example shows that counting quantifiers together with the fact that the vocabulary \( \Sigma \) is not a priori bounded, give the logic ability to express some cardinality constraints.

Example 3. We will encode the property that a structure contains a substructure consisting of two disjoint trees of degree bounded by \( d \), one of which has twice as many elements as the other. To express that there are two trees it is enough to add the conjunct \( \exists^=2x.root(x) \) to \( forest(root,\ tree,\ edge) \). To distinguish between elements of both trees we introduce two fresh unary predicates, \( first(\cdot) \) and \( second(\cdot) \). The formula

\[
\exists x.root(x) \land first(x) \land \forall x,y.first(x) \land edge(x,y) \rightarrow first(y)
\]

ensures that all nodes of the first tree are labeled with the \( first(\cdot) \) predicate. In a similar way we express that the element of the second tree are labeled with \( second(\cdot) \) predicate. Then we add conjuncts expressing that these predicates are disjoint and label only nodes of trees. Finally, to say that the second tree has twice more elements than the first one, we introduce a fresh predicate \( match(\cdot,\cdot) \) and the formula

\[
(\forall x.first(x) \rightarrow \exists^=2y.match(x,y) \land second(y)) \land
(\forall y.second(y) \rightarrow \exists^=1x.match(x,y) \land first(x)).
\]

In the last example we want to show that the logic can express two successor relations of linear orders. This time we
use two forests, one defined by relation $\text{edge}_1$ and the other by $\text{edge}_2$.

**Example 4.** If the symbol $\text{edge}_1$ is interpreted as a successor relation in a tree, then the formula $\exists^1 x. \text{root}_1(x) \wedge \forall x \cdot \text{tree}_1(x) \wedge \exists^1 x. \forall y. \neg \text{edge}_1(x, y)$, in conjunction with forest($\text{root}_1, \text{tree}_1, \text{edge}_1$), expresses that there is exactly one tree in a forest, every node in the underlying structure is a node in the tree and that the tree has exactly one leaf, so the edge_1 relation is a successor relation of some linear order. Using the same formula with predicates root2, tree2 and edge2, we may express the second successor relation of a linear order.

Note that if a finite structure $A$ is a model of a $\Sigma^2$ formula $\phi$ over a signature $\Sigma$ and $\text{edge}_1, \text{edge}_2$ are fresh binary predicate symbols, then $A$ can be extended to a model of $\phi$ in the class of structures with two forests spread over predicates $\text{edge}_1$ and $\text{edge}_2$, so we retain the full expressive power of $\Sigma^2$.

**B. Notation**

Whenever possible, we follow the notation and terminology from [21]. Let $\Sigma$ be a relational vocabulary (that is, a vocabulary not containing function symbols). We assume that $\Sigma$ contains only predicates of arity 1 and 2. The formulas of $\Sigma^2$ are first-order formulas built over the vocabulary $\Sigma$, extended with counting quantifiers of the form $\exists^k x$, $\exists^k y$ or $\exists^k $, and employing only the variables $x$ and $y$. The numbers in counting quantifiers as well as the number $d$ bounding the degree of trees are encoded in binary.

**Definition 1** (Class $\mathbb{F}_d(\Sigma, s_1, s_2)$). Let $\Sigma$ be a relational vocabulary containing at least predicates $s_1(\cdot, \cdot)$ and $s_2(\cdot, \cdot)$. The class $\mathbb{F}_d(\Sigma, s_1, s_2)$ is the class of finite structures $A$ over $\Sigma$ where, for $i \in \{1, 2\}$, the predicate $s_i$ is interpreted as a forest of degree at most $d$ (that is, a directed acyclic graph such that the indegree of each non-root node is 1 and the outdegree of each node is bounded by $d$).

In the following we solve the satisfiability problem for $\Sigma^2$ in the class $\mathbb{F}_d(\Sigma, s_1, s_2)$, i.e., the problem whether for a given formula $\varphi$ and a number $d$ there exists a structure $A \in \mathbb{F}_d(\Sigma, s_1, s_2)$ such that $A \models \varphi$.

A 1-type over the signature $\Sigma$ is a maximal propositionally consistent conjunction of atomic and negated atomic formulas over $\Sigma$ involving only the variable $x$. Similarly, a 2-type is a maximal propositionally consistent conjunction of atomic and negated atomic formulas over $\Sigma$ involving only the variables $x$ and $y$ and containing $\neg x = y$. For complexity reasons we will identify a 1-type $\tau$ (a 2-type $\tau$) with the set of positive atomic formulas occurring in $\tau$ (in $\tau$).

Each 2-type $\tau(x, y)$ uniquely determines two 1-types of $x$ and $y$, respectively, that we denote $tp_1(\tau)$ and $tp_2(\tau)$. For a 2-type $\tau$ the 2-type obtained by swapping the variables $x$ and $y$ is denoted $\tau^{-1}$. Symbols $\Pi(\Sigma)$ and $\Phi(\Sigma)$ denote the set of 1-types and the set of 2-types over $\Sigma$ respectively.

For a structure $A$ over the signature $\Sigma$ and an element $e \in A$, $\text{tp}^A(e)$ denotes a unique $1$-type $\pi \in \Pi(\Sigma)$ such that $A \models \pi(e)$. Similarly, for $e_1, e_2 \in A$, $\text{tp}^A(e_1, e_2)$ is the unique 2-type $\tau \in \Phi(\Sigma)$ such that $A \models \tau(e_1, e_2)$. Symbol $\Pi(A)$ denotes the set of 1-types over $\Sigma$ realized in $A$.

If $\Sigma$ is a relational signature and $\tuple{f} = f_1, \ldots, f_m$ is a sequence of distinct binary predicates in $\Sigma$, then the pair $\langle \Sigma, \tuple{f} \rangle$ is called a classified signature. Let $\langle \Sigma, \tuple{f} \rangle$ be a classified signature and let $\tau(x, y)$ be a 2-type over $\Sigma$. We say that $\tau$ is a message type over $\langle \Sigma, \tuple{f} \rangle$ if $f(x, y) \in \tau(x, y)$ for some distinguished predicate $f$ in $\tuple{f}$. If $\tau(x, y)$ is a message type such that $\tau^{-1}$ is also a message type then we say that $\tau$ is an invertible message type. If $\tau$ or $\tau^{-1}$ are message types then $\tau$ is called a noisy type. On the other hand, if $\tau$ is a 2-type such that neither $\tau$ nor $\tau^{-1}$ is a message type, then we say that $\tau$ is a silent type. Let $\mathcal{A}$ be a relational structure over $\langle \Sigma, \tuple{f} \rangle$ and let $c$ be a natural number. Structure $\mathcal{A}$ is said to be $c$-bounded if $\mathcal{A} \models \forall x \exists^c y, (f(x, y) \wedge x \neq y)$ for every $f \in \tuple{f}$.

For a relational structure $\mathcal{A}$ over a vocabulary containing at least binary predicate $s$ denote by $\mathcal{A}_s$ the substructure of $\mathcal{A}$ induced by the relation $s$. Similarly, if there are two binary predicates $s_1$ and $s_2$, denote by $\mathcal{A}_{s_1, s_2}$ the substructure of $\mathcal{A}$ induced by $s_1 \cup s_2$.

Given a structure $\mathcal{A}$ over a classified signature $\langle \Sigma, \tuple{f} \rangle$ and a message type $\tau$, if $\mathcal{A} \models \tau(e_1, e_2)$ then $e_2$ is called a witness for $\tau$ in $\mathcal{A}$. It follows that if $\tau$ is an invertible message type then also $e_1$ is a witness for $e_2$. It is because $\mathcal{A} \models \tau^{-1}(e_2, e_1)$ and $\tau^{-1}$ is an (invertible) message type.

Since we consider predicates of arity at most 2, a structure $\mathcal{A}$ can be seen as a complete directed graph, where nodes are labeled by 1-types and edges are labeled by 2-types. Thus to define such a structure it is enough to define 1-types of its elements and 2-types of all pairs of elements provided that the projections of 2-types onto 1-types coincide with these 1-types and that for each pair $(e_1, e_2)$ of elements connected by a 2-type $\mu$ the pair $\langle e_2, e_1 \rangle$ is connected by the 2-type $\mu^{-1}$.

We say that a signature $\Sigma$ extends a signature $\Sigma'$ if $\Sigma \subseteq \Sigma'$. Let $\tau$ be a 2-type over $\Sigma_e$. Denote by $\eta|_{\Sigma}$ the restriction of the type $\tau$ to the vocabulary $\Sigma_e$, that is the unique 2-type over $\Sigma_e$ obtained by removing $(\Sigma_e \setminus \Sigma)$-literals from $\tau$. Similarly, for a 1-type $\pi$ over $\Sigma_e$ by $\pi|_{\Sigma_e}$ we denote the restriction of the the type $\pi$ to the vocabulary $\Sigma_e$. A (1- or 2-type $\tau$ over the signature $\Sigma_e$ extends a type $\tau'$ over $\Sigma$ if $\tau|_{\Sigma} = \tau'$. If $\mathcal{B}$ is any structure over $\Sigma_e$ then $B|_{\Sigma}$ is the restriction of $B$ to the vocabulary $\Sigma$ (and $B$ is an extension of $B|_{\Sigma}$). If $B$ is a set of (1- or 2-)types over $\Sigma_e$, then we define $B|_{\Sigma} = \{ \eta|_{\Sigma} \mid \eta \in B \}$. A 1-type $\kappa \in \Pi(\Sigma)$ that has only one realization in a structure $\mathcal{A}$ is said to be a king 1-type in $\mathcal{A}$. If an element $e \in \mathcal{A}$ realizes a king 1-type then it is said to be a king in $\mathcal{A}$. Any structure may have multiple kings.

**III. Satisfiability of $\Sigma^2$ in $\mathbb{F}_d(\Sigma, s_1, s_2)$**

Our aim is to check if any formula $\varphi \in \Sigma^2$ has a model in $\mathbb{F}_d(\Sigma, s_1, s_2)$. We start with a high-level description of our method.

A. Overview of the decision procedure

The first crucial observation is that although $\Sigma^2$ is not able to express that a structure is a tree, it is able to express...
something very close to this property, namely that a structure consists of a tree and some number of cycles disconnected from this tree. Therefore, by simply adding some conjuncts to the input formula, we may be sure that the considered structure consists of a forest with several cycles as on Figure 1.

![Fig. 1. A model of the formula forest(r, t, s) with a tree-like cycle. Each node in the structure is labeled by predicate t(·). In addition the only root is labeled by r(·). Unlabeled edges denote the predicate s(·, ·).](image)

The second crucial observation is that if a cycle contains an edge labeled \( \tau \) (a 2-type \( \tau \)) and the same 2-type occurs in a tree, then a simple model surgery allows to implant the cycle into the tree as shown on Figure 2. That is, given a (partial) model of the input formula that contains a tree and a cycle with some 2-type occurring in both of them, we may construct a model with a bigger tree and a smaller number of cycles.

![Fig. 2. Breaking a tree-like cycle. Up: the cycle on elements \( e_1, e_2, e_3 \) can be broken, because edge \( \langle e_1, e_2 \rangle \) realizes an invertible type \( \tau \), which is also realized by edge \( \langle h_1, h_2 \rangle \). Down: the tree obtained after model surgery.](image)

The third step is to make sure that some tree in the considered structure contains the 2-types needed in the previous step. To achieve this, we introduce extended types (represented by predicates \( \text{type}_{e\Phi} \) in the next section). Intuitively, with the extended type \( \Phi \) of a node we want to collect all the 2-types of all edges in a subtree rooted at this node. Having this, by only local consistency checks (by checking the extended type of a node and looking only at the 2-types of the edges originating at this node and at the extended types of immediate successors) we may collect in the root of a tree the set of all 2-types of edges present in the tree. Then we may simply require that the extended types of roots in a structure contain all 2-types of forest edges present in the structure. Having this, we may implant any cycle to some tree.

The fourth crucial observation is that collecting all this information is too costly and we cannot afford it, but we do not need it. Specifically, assume that we need a 2-type \( \tau \) to occur in the extended type of a root \( r \) of some tree \( t \) and that \( r \) has two immediate successors \( e_1 \) and \( e_2 \). Now, if \( \tau \) occurs in the extended types of both \( e_1 \) and \( e_2 \) then it is redundant and we may safely remove it from the extended type of one of these nodes. Therefore, in the following construction an extended type \( \Phi \) of a node \( e \) does not collect all the 2-types in the subtree rooted at \( e \) — it only promises that all 2-types from \( \Phi \) occur in this subtree. This simple observation decreases the complexity by one exponent.

The fifth crucial observation is that extended types of roots in a forest, understood as promises, are preserved by the model surgery from the second step. This is simply because model surgery increases the trees, so all 2-types present in a subtree before the surgery remain present afterwards. On the other hand, the extended types of internal nodes may not be preserved. This is because promises in cycles are not well founded (each node in a cycle may delegate everything it promises to the successor), so the promises from a cycle need not be fulfilled in the tree after the surgery. For this reason we keep the information about extended types in a separate signature and at the time of surgery this information is erased from the constructed structure.

The sixth crucial observation is that there is still some redundancy. Namely, in the second step we required that some edge of a cycle matches some edge in a tree, but in the construction we guarantee that all edges of a cycle match some edges in some trees. Exploiting this redundancy we may handle structures with two forests. Let us refer to the two forests as the black one and the green one. Suppose that the structure contains a cycle \( C \) made of black edges. If some of the edges in the cycle is not green, we may choose this one for the model surgery and we may be sure that this surgery does not affect anything in the green forest. On the other hand, if all edges in the cycle are both black and green then the considered cycle is both black and green at the same time and the model surgery implants the cycle into a black tree thus decreasing the number of black cycles; in this case we also break the corresponding green cycle and either we implant it into a green tree as shown on Figure 3, or we implant it into another, existing green cycle (to see this, add a green edge from the green successor of \( h_1 \) to \( e'_2 \) on Figure 3), or we create a new green cycle that contains all nodes of \( C \) but is not a black cycle (to see this, add a green edge from \( e_2 \) to \( e'_1 \)). In all cases the total number of cycles present in the structure decreases.

On top of these six observations we now have to put the Pratt-Hartmann construction from [21]. Let us briefly recall this construction. First, the input formula is brought into a version of Scott normal form where all existential quantifiers occur in a context of the form \( \forall x \exists ! y \ldots \). The class of considered models is also simplified so that only so-
We use chromatized and silent types. All edges appearing in the forest are invertible message types, but we have to take some additional care about kings. Establishing silent types is the last step of the construction. Unfortunately the whole construction is quite sophisticated and because of space limits some parts of it had to be moved to the appendix.

### B. Detailed construction

In this section we demonstrate that a $C \leq 2$ formula $\varphi$ is satisfiable in $F_d(\Sigma, s_1, s_2)$ if and only if there exists an exponential (in $c! + \log d$) description of some of its models. For one direction, starting from an arbitrary model $A$ of $\varphi$ in $F_d(\Sigma, s_1, s_2)$, we expand $A$ by interpreting some fresh predicates; we also introduce a vocabulary $\Sigma_c$ that extends $\Sigma$ and use predicates from the extended vocabulary to ensure that the constructed structure is well-typed wrt. extended types. Then we define the description as a frame and a system of inequalities (see Definitions 12 and 15). For the other direction, given such a description we construct a model $A$ that fits to the description.

In the first order logic it is not possible to define structures $A$ such that $A_\alpha$ is a (finite) forest of trees. However, it is possible to state that some nodes are roots of trees in $A_\alpha$. Consider the formula $\text{forest}(\text{root}, \text{tree}, s)$ defined below.

**Definition 2 (Forest Formula).** Let $s(\cdot, \cdot)$ be a binary predicate and let $\text{root}(\cdot)$, $\text{tree}(\cdot)$ be unary predicates. Define forest formula $\text{forest}(\text{root}(\cdot), \text{tree}(\cdot), s)$ to be the conjunction of the following $C \leq 2$ formulas: $\forall y. \text{root}(y) \rightarrow \text{tree}(y)$, $\forall y. \neg s(y, y)$, $\forall x \forall y.s(x, y) \rightarrow \neg \text{root}(y) \land \text{tree}(x) \land \text{tree}(y)$ and $\forall x. \text{tree}(x) \land \neg \text{root}(x) \rightarrow \exists\overline{y}.s(y, x)$.

Define $\text{forest}(r_1, t_1, s_1, r_2, t_2, s_2) = \text{forest}(r_1, t_1, s_1) \land \text{forest}(r_2, t_2, s_2)$. To shorten the notation we will often write $\text{forest}$ without parameters to denote $\text{forest}(r_1, t_1, s_1, r_2, t_2, s_2)$. Since the forest formula is true on all structures in $F_d(\Sigma, s_1, s_2)$ if the additional unary predicates $r_1, t_i$ are interpreted appropriately, for all $C \leq 2$ formulas $\varphi$ the formulas $\varphi$ and $\varphi \land \text{forest}$ are equisatisfiable on structures in $F_d(\Sigma, s_1, s_2)$. Therefore, without loss of generality we can assume that all considered formulas $\varphi$ have $\text{forest}$ as a conjunct. Consider any finite model $A$ of $\text{forest}(r, t, s)$. Then all elements labeled with $r$ are roots of some trees in $A_\alpha$, but the graph $A_\alpha$ may contain cycles as shown on Figure 1. Notice that each node of $A_\alpha$ occurs in at most one simple cycle. Call a maximal connected substructure in $A_\alpha$ containing a cycle a *tree-like cycle*.

For a natural number $n$ denote by $\mathbf{m}$ the set $\{1, \ldots, n\}$. We will assume that the input $C \leq 2$ formula $\varphi$ is in a normal form

$$\varphi = \forall x \forall y. (\alpha(x, y) \lor x = y) \land \bigwedge_{h \in \mathbf{m}} \forall x \exists d_h.C_h.f_h(x, y) \land (x \neq y)$$

where $\alpha$ is a quantifier-free formula with unary and binary predicate symbols, $\leq h \in \{\leq \cdot \}$ for $h \in \mathbf{m}$, $C_1, \ldots, C_m$ are positive integers encoded in binary, and $f_1, \ldots, f_m$ are distinguished binary predicates. Let $c = \max\{C_h | h \in \mathbf{m}\}$. It is well known (see Lemma 1. in [20]) that each $C \leq 2$ formula $\varphi$ can be transformed to a formula in normal form that is equivalent to $\varphi$ over structures of cardinality $\geq c$, and uses
only bounds of the form $C_h$ in counting quantifiers. Here we relax the normal form and additionally use bounds of the form $\leq C_h$ to assure that in every model $M \in F_d(\Sigma, s_1, s_2)$ of all edges of $\mathcal{M}_{s_0, s_1}$ realize invertible message types. Such structures $M$ will be called structures with rigid forests. Observe that each element in a model of the formula $\varphi$ has at most $\sum_{k=1}^{m} C_h$ different witnesses, which is a number bounded by $mc$.

Now, to simplify the reasoning, we modify the formula $\varphi$ and restrict the class of models that we consider. Let $F_d(s_1, s_2)$ be $\{ \square F_d(s_1, s_2) \}$. Let $i \in \mathbb{Z}$. A 2-type $\tau$ is called a tree-$i$ 2-type if it is a 2-type of some edge in the forest no. $i$, that is if $\tau \models s_i(x, y)$. It is called a tree 2-type if it is tree-1 or tree-2 2-type.

**Definition 3.** A structure $A \in F_d(s_1, s_2)$ is said to be
- a structure with rigid forests if every tree 2-type $\tau$ realized in $A$ is an invertible message type,
- chromatic if distinct elements connected by a chain of one or two invertible message types always have distinct 1-types,
- $Z$-differentiated, where $Z$ is a natural number, if any 1-type realized in $A$ has either one or more than $Z$ realizations.

In any structure in $F_d(\Sigma, s_1, s_2)$, by introducing new binary predicates $\text{parent}_i$ and adding to the initial formula some conjuncts stating that $\text{parent}_i$ is the reverse of $s_i$, we can easily make both forests rigid. By Lemmas 2 and 3 in [20], using a small number of additional unary predicates, we can make the structure chromatized and $Z$-differentiated (see Appendix A for the details of this construction). Together with earlier considerations concerning $C^2$ formulas this allows to restrict our attention to chromatic and $3\text{mc}$-differentiated structures with rigid forests that belong to $F_d(\Sigma, s_1, s_2)$, and formulas in normal form (1) that entail forest. From now on we assume that a formula $\varphi$ and a classified signature $(\Sigma, \overline{T})$ are fixed.

We now define an extended signature $\Sigma_{\text{ext}}$. In the following construction this signature will be used to store the information about extended types of elements in a structure. We introduce unary predicates of the form $\text{type}_{\varphi}(\cdot)$, where $\Phi$ is a subset of the set of all invertible 2-types $\tau$ over $(\Sigma, \overline{f})$, and let $\Sigma_{\text{ext}} = \Sigma \cup \{ \text{type}_{\varphi}(\cdot) \mid \Phi \subseteq \Phi(\Sigma) \land i \in \mathbb{Z} \}$. Unfortunately this signature is too big and for complexity reasons we will have to work with some signature $\Sigma_c \subseteq \Sigma_{\text{ext}}$: intuitively $\Sigma_c$ describes only extended types that do occur in the considered structure. Note that we do not define $\Sigma_{\text{ext}}$ at this point, so the following definitions work for any fixed $\Sigma_c \subseteq \Sigma_{\text{ext}}$.

Let $A$ be any structure in $F_d(\Sigma, s_1, s_2)$. For $i \in \mathbb{Z}$ denote by $T_i(e)$ the subtree of $A_{s_i}$ rooted in $e$. By the type of forest no. $i$ in $A$, in symbols $\Phi^i(A)$ we mean the set of all tree, 2-types realized in $A_i$.

**Definition 4.** Let $B$ be a structure in $F_d(\Sigma_c, s_1, s_2)$ and $i \in \mathbb{Z}$. We say that a node $e \in B_{s_i}$ is well typed in forest $i$ if there exists exactly one subset $\Phi$ of the set $\Phi^i(B|_{\Sigma_c})$ such that $B \models \text{type}_{\varphi}(e)$ and
- either $e$ is a leaf in $B_{s_i}$ and $\Phi = \emptyset$, or
- $e$ is a non-leaf node in $B_{s_i}$ with immediate successors $\{ e_j \}_{j=1}^k$ and
  $$\Phi \subseteq \{ \tau_1 \ldots \tau_k \} \cup \bigcup_{j=1}^k \Phi_j,$$
  where $\tau_j$ is the 2-type connecting $e$ with $e_j$, restricted to the signature $\Sigma_c$, $B \models \text{type}_{\varphi}(e_j)$ and $e_j$ is well typed in forest $i$, for $j \in \mathbb{Z}$.

Intuitively, this definition says that the theorem made in the node $e$ is fulfilled: all 2-types from $\Phi$ do occur in the tree rooted at $e$.

**Definition 5.** Let $B$ be a structure in $F_d(\Sigma_c, s_1, s_2)$ and $i \in \mathbb{Z}$. Let $\{ e_j \}_{j=1}^k$ be the sequence of all tree roots in $B_{s_i}$. We say that the forest $i$ is well typed in $B$ if for all $j \in \mathbb{Z}$ the root $e_j$ is well typed in forest $i$ with $B \models \text{type}_{\varphi}(e_j)$, and
  $$\Phi^i(B|_{\Sigma_c}) \subseteq \bigcup_{j=1}^k \Phi_j.$$  

Finally, a structure $B$ is well typed if both forests 1 and 2 are well typed in $B$.

Intuitively, the above definition says that all 2-types of edges in the forest occur in labels of the roots of the forest, that is, each tree, 2-type in $A$ is realized in some tree in $B_{s_i}$. The following lemma says that all structures in $F_d(\Sigma, s_1, s_2)$ can be well typed using a small number of type predicates.

**Lemma 1.** Let $A \in F_d(\Sigma, s_1, s_2)$ be a chromatic and $3\text{mc}$-differentiated structure over $(\Sigma, \overline{f})$. There exists an extension $B$ of the structure $A$ to a signature $(\Sigma_c, \overline{f})$ where $\Sigma_c \subseteq \Sigma_{\text{ext}}$ such that $B$ is well typed. Moreover $|\Sigma_c \setminus \Sigma| \leq 2(\Phi^1(A)) + |\Phi^2(A)| + 1$.

**Definition 6 (Decorated Structure).** Let $B$ be a relational structure over $(\Sigma_c, \overline{f})$, where $\Sigma_c$ extends $\Sigma$. We say that $B$ is $3\text{mc}$-decorated if it is well typed and $B|_{\Sigma}$ is chromatic and $3\text{mc}$-differentiated structure with rigid forests.

For the rest of this section we assume that $\varphi$ is a $C^2$ formula in normal form that entails the forest formula $\varphi$ and that some model of $\varphi$ is a $3\text{mc}$-decorated structure.

One of the building blocks of our descriptions are sets of 1-types, over an extended signature $\Sigma_c$, that should describe king 1-types over an initial signature $\Sigma$. For this reason we need a technical property of extended types that we call compatibility with $\Sigma$. A subset $K_e$ of $\Pi(\Sigma_c)$ is called compatible with $\Sigma$ if $\kappa_1|\Sigma = \kappa_2|\Sigma$ implies $\kappa_1 = \kappa_2$, for all $\kappa_1, \kappa_2 \in K_e$. Intuitively, compatibility is a requirement that a king (which has a unique type in the considered structure) has a unique extended type. Of course in any decorated structure $B$ over $\Sigma_c$, the set $\{ \pi \mid \pi|_{\Sigma} \text{ is a king type in } B|_{\Sigma} \}$ is compatible with $\Sigma$.

Given a structure $B$ over a signature $(\Sigma_c, \overline{f})$ and an element $a \in B$, we want to capture message types connecting $a$ to other
elements of $\mathcal{B}$ and all 2-types connecting $a$ to kings of $\mathcal{B}_{|\Sigma}$. We first define a set of all these 2-types. If $K_e$ is a set of 1-types over $\Sigma_e$ compatible with $\Sigma$, then denote by $\tau(K_e, \Sigma, \Sigma_e, \hat{f})$ the set of all 2-types $\mu$, such that $\mu$ is a message type over $(\Sigma_e, f)$ or $\text{tp}_2(\mu) \in K_e$.

We are now ready to introduce star types.

**Definition 7** (Star type in $\mathcal{A}$). Let $\mathcal{A}$ be a structure over $(\Sigma_e, f)$, where $\Sigma_e$ extends $\Sigma$, and let $a$ be an element of $\mathcal{A}$. Moreover, let $K_e = \{ \kappa \mid \kappa|\Sigma_e \text{ is a king type in } \mathcal{A}_{|\Sigma_e} \}$. Assume that $\mu_1, \ldots, \mu_{M_1}, \mu_{M_1+1}, \ldots, \mu_M$ is a standard enumeration of $\tau(K_e, \Sigma, \Sigma_e, \hat{f})$, where $\mu_1, \ldots, \mu_{M_1}$ are the invertible message types and $\mu_{M_1+1}, \ldots, \mu_M$ are the remaining types (silent types ending in $K_e$ and non-invertible message types). A star type of $a$ in $\mathcal{A}$, denoted $\text{st}^a(\pi)$, is a tuple $\pi = (\pi, v_1, \ldots, v_M)$ where $\pi = \text{tp}^a(a)$ and for all $j \in \mathcal{M}$ the number $v_j$ is the number of realizations of the 2-type $\mu_j$ originating in $a$, in symbols

$$v_j = |\{ b \in \mathcal{A} \mid \text{tp}^a(ab, \mu_j) \}|.$$

We denote the number $v_j$ by $\sigma[j]$ and the type $\pi$ by $\pi(a)$.

Observe that in the definition above $\pi$ satisfies the conditions

1. $v_j \geq 0$ for all $j \in M$,
2. $v_j > 0$ implies $\text{tp}_2(\mu_j) = \pi$ for all $j \in M$,
3. $\sum_{j \in \mathcal{M}} (v_j = 1)$ for all $\kappa \in K_e$ such that $\kappa \neq \pi$,
4. $\sum_{j \in \mathcal{M}} (v_j = 0)$ if $\pi \notin K_e$.

The first of these conditions is obvious. The second one says that all 2-types originating in $a$ have the same 1-type of the origin, namely the 1-type of $a$. The third one says that for all kings $k$ (in $A_{|\Sigma}$) the element $a$ is connected with $k$ by exactly one 2-type (provided that $k \neq a$). The last condition says that if $a$ is a king then it is not connected with itself by any 2-type (recall that 2-types connect different elements). Notice that the set $K_e$ in the Definition 7 is compatible with $\Sigma$.

**Definition 8**. A star type over the set of 2-types $\tau(K_e, \Sigma, \Sigma_e, \hat{f})$ is any tuple $(\pi, v_1, \ldots, v_M)$ satisfying conditions 1)–4) above. A structure $\mathcal{A}$ is said to realize a star type $\pi$ if $\text{st}^a(\pi) = \sigma$ for some $a \in \mathcal{A}$.

If $K$ is some subset of $\Pi(\Sigma)$ then a star type over the set of 2-types $\tau(K, \Sigma, \Sigma, \hat{f})$ is called a basic star type. Note that this definition is correct since $\Sigma$ trivially extends itself. Let $\mathcal{A}$ be a structure over $(\Sigma_e, f)$ and let $\mathcal{A}' = \mathcal{A}_{|\Sigma_e}$. If $\sigma = \text{st}^a(\pi)$ for some $a \in \mathcal{A}$ then we define $\sigma|_{\Sigma_e}$ as the basic star type $\text{st}^a(\pi)$. We say that $\sigma|_{\Sigma_e}$ is the restriction of $\sigma$ to vocabulary $\Sigma$. If $ST$ is a set of star types over $\tau(K_e, \Sigma, \Sigma_e, f)$ then we define $ST_{|\Sigma_e} = \{ \sigma|_{\Sigma_e} \mid \sigma \in ST \}$.

We say that a 2-type $\mu_j$ occurs in a star type $\pi$ if $\sigma[j] > 0$. We say that a star type $\pi$ is a tree, star type if $t_i(x)$ is a conjunct in $\pi(a)$. A star type is rigid if all tree 2-types occurring in $\pi$ are invertible. If 1-types of all the neighbors of a given element $a$ connected with it by invertible message types are pairwise distinct and distinct from $\text{tp}^a(a)$ then the star type $\text{st}^a(\pi)$ is said to be chromatic. Observe that a structure $\mathcal{A}$ is chromatic if and only if all star types realized in $\mathcal{A}$ are chromatic.

Checking if a given structure $\mathcal{A}$ satisfies the formula $\varphi$ can be done locally: it is enough to inspect every element $e$ of $\mathcal{A}$ along with the sequence of 2-types originating in $e$. The inspection checks if 2-types in the sequence entail the subformulas of $\varphi$ of the form $\forall x, y, \ldots$ and that the number of witnesses is correct. First, we inspect only 2-types connecting $e$ with its important neighbors, represented by the star type of $e$, as stated by the following definition. We postpone the full inspection until Definition 13.

**Definition 9** ($\sigma \vdash \varphi$). Let $\sigma = (\pi, v_1, \ldots, v_M)$ be a star type and let $\varphi$ be a $C^2$ formula in normal form (1). We say that $\sigma$ satisfies $\varphi$, in symbols $\sigma \vdash \varphi$ if:

- for each 2-type $\mu_i$ such that $\sigma[i] > 0$ the formula $\alpha$ is a consequence of $\mu_0$ and $\mu^0_1$, that is $\sigma = \mu_0 \rightarrow \alpha$, where $\mu_0$ is seen as conjunction of literals, and
- for each $h \in M$ we have

$$\left( \sum_{i \in \mu} \sigma[i] \right) \implies h C_h.$$

Note also that although forest is not a formula in normal form, it describes only local properties that can be checked by inspection of star types. The only interesting conjunct of forest$(r, t, s)$ that is not in normal form is $\forall x. t(x) \land \neg r(x) \rightarrow \exists y.s(y, x)$. To check if this conjunct is true in $\sigma$ it is enough to check that whenever $t(x) \in \pi(\sigma)$ and $r(x) \notin \pi(\sigma)$, there exists exactly one 2-type that occurs in $\sigma$ and contains $s(y, x)$ as a conjunct. Therefore, even if it is not an instance of the above definition, we feel free to say that some star types satisfy the forest formula.

For a given set of star types $ST$, by $\pi(ST)$ we denote the set $\{ \pi(\sigma) \mid \sigma \in ST \}$ and by $\tau(ST)$ — the set of 2-types occurring in star types from $ST$, that is the set $\{ \mu_j \mid \exists \sigma \in ST. j \in M \land \sigma[j] > 0 \}$. By type of forest $i$ in $ST$, in symbols $\Phi^i(ST)$ we mean the set of all tree, 2-types occurring in star types from $ST$, that is the set $\{ \tau_{|\Sigma} \mid s_i(x, y) \in \tau \land \exists \sigma \in ST. \tau \text{ occurs in } \sigma \}$.

The following definition describes a local consistency check for extended types inside star types: a node $e$ promises that all 2-types from a set $\Phi$ occur in the subtree rooted at $e$, and we check that these 2-types either connect $e$ with its immediate successors or are promised by these successors.

**Definition 10**. Let $i \in 2$ and let $\sigma$ be a rigid tree, star type over $\tau(K_e, \Sigma, \Sigma_e, f)$. Let $\{ \tau_j \}_{j=1}^i$ be the sequence of all tree, 2-types occurring in $\sigma$. Then $\sigma$ is called well typed in forest $i$ if there exists exactly one subset $\Phi$ of the set $\Phi(\Sigma)$ such that $\text{type}^i_{\Phi}(x) \in \pi(\sigma)$, and

- either $k = 0$ and $\Phi = \emptyset$, or
- $k > 0$, $\Phi \subseteq \{ \tau_1, \ldots, \tau_k \} \cup \bigcup_{j=1}^i \Phi_j$ and, for $j \in \mathcal{K}$, $\Phi_j \subseteq \Phi(\Sigma)$ and $\text{type}^i_{\Phi_j}(y) \in \text{tp}_2(\tau_j)$.

**Definition 11**. Let $ST$ be a set of rigid star types over $\tau(K_e, \Sigma, \Sigma_e, f)$ and let $i \in 2$. Let $\{ \sigma_j \}_{j=1}^i$ be the set
consisting of all star types of tree roots of the forest $i$ in ST. We say that the set ST is well typed in forest $i$ if all star types in ST are well typed in forest $i$ and

$$\Phi^i(\text{ST}) \subseteq \bigcup_{j=1}^{l} \Phi_j$$

where \( \text{type}^i_{\phi_j}(x) \in \pi(\sigma_j) \) for all $j \in \mathbb{L}$ Finally, the set ST is called well typed if it is well typed in both forests 1 and 2.

The definition above simply says that all tree 2-types occurring in the structure occur in the extended type of some root. We now introduce finite and small structures called frames.

Condition 1 says that each king has exactly one star type.

A \( \Sigma \) number of distinguished predicates in \( A \) structure satisfies the formula \( \pi \) rigid and chromatic star types over \( \Sigma \) their restrictions to \( \text{conditions are satisfied} \). The following definition says how to check if the local \( \text{silent} 2\text{-types over} \), if \( \phi \) consisting of all star types of tree roots of the forest \( i \)

\begin{align*}
\text{1)} & \text{ for each } \kappa \in K \text{ there exists exactly one } \sigma \in ST \text{ such that } \\
\text{2)} & \text{ for each two 1-types } \kappa, \kappa' \in (\pi(ST) \setminus K) \text{ there exists a } \\
\text{3)} & \text{ for each star type } \sigma \in ST \text{ and each } i \in M, \text{ if } \sigma[i] > 0 \\
\text{4)} & \text{ for each star type } \sigma \in ST, \text{ we have } \sigma \models \text{forest.} \\
\end{align*}

Frames are intended to describe local configurations in 3mc-decorated structures \( A \) from \( F_d(\Sigma_e, s_1, s_2) \) (recall that \( m \) is the number of distinguished predicates in \( \Sigma \)). The set \( K \) contains all king 1-types of \( A_{\Sigma_e} \), \( ST \) all star types of \( A \) and the set \( \Xi \) all silent 2-types realized in \( A_{\Sigma_e} \) between non-kings. Condition 1 says that each king has exactly one star type. Condition 2 expresses that for each two non-king 1-types realized in a model there exist a silent 2-type that connects elements of these types and is also realized in this model. Condition 3 ensures that if a neighbor of a node in a structure has some 1-type \( \pi \), then there exists a star type \( \sigma \in ST \) such that \( \pi \in \pi(ST) \). Finally, Condition 4 states that the underlying structure satisfies the formula forest.

The following definition says how to check if the local configurations described in a frame are consistent with the input formula.

**Definition 13** (\( F \models \varphi \)). Let \( F = (K, ST, \Xi, \Sigma_e, f, c) \) be a frame and let \( \varphi \) be a \( C^2 \) formula in normal form (1) over vocabulary \( (\Sigma, f) \). We say that \( F \) satisfies \( \varphi \), in symbols \( F \models \varphi \), if

- for each 2-type \( \mu \in \Xi \) the formula \( \varphi \) is a consequence of \( \mu \) and of \( \mu^{-1} \), that is \( \models \mu \rightarrow \alpha \) and \( \models \mu^{-1} \rightarrow \alpha \), where \( \mu \) is seen as conjunction of literals, and
- for each \( \sigma \in ST \), \( \sigma \models \varphi \).

Intuitively, we want to check satisfiability of a \( C^2 \) formula \( \varphi \) by guessing a frame that satisfies \( \varphi \). This, however, checks only local consistency. It still remains to be checked whether there exists a structure that fits to the frame. A silent type \( \tau \) such that \( tp_1(\tau) \) and \( tp_2(\tau) \) are not king types is a non-king silent type.

**Definition 14.** Let \( \langle K, ST, \Xi, \Sigma_e, \Sigma, f, c \rangle \) be a frame, and \( A \) structure over \( (\Sigma_e, f) \). We say that \( A \) fits to the frame \( F \) if

- the set of king 1-types realized in structure \( A_{|\Sigma_e} \) is \( K \), and
- the set of all non-king silent types realized in \( A_{\Sigma} \) is a subset of \( \Xi \), and
- the set of star types of \( A_{\Xi} \) is a subset of \( ST_{|\Sigma} \), in symbols \( st(A_{|\Sigma_e}) \subseteq ST_{|\Sigma} \).

We say that \( A \) precisely fits to \( F \) if the last condition is replaced by \( st(A) = ST \).

Note the two involved signatures: a structure \( A \) that precisely fits to a frame must be defined over the signature \( \Sigma_e \) while to simply fit to a frame it enough if \( A \) is defined over \( \Sigma \). Here we use the observation that since \( \Sigma \subseteq \Sigma_e \), each structure over \( \Sigma \) can be trivially promoted to a structure over \( \Sigma_e \).

The following proposition reduces the satisfiability problem in the class \( F_d(\Sigma, s_1, s_2) \) to the problem of existence of a structure in \( F_d(\Sigma, s_1, s_2) \) that fits to a given frame.

**Proposition 1.** Let \( \varphi \) be a \( C^2 \) formula in normal form over a vocabulary \( (\Sigma, f) \).

1) Let \( A \) be a 3mc-decorated structure in \( F_d(\Sigma_e, s_1, s_2) \). If \( A \models \varphi \) then there exists a frame \( F \), such that \( A \) precisely fits to \( F \) and \( F \models \varphi \).

2) Let \( A \) be a structure in \( F_d(\Sigma, s_1, s_2) \). If there exists a frame \( F \) such that \( A \) fits to \( F \) and \( F \models \varphi \) then \( A \models \varphi \).

In the following we will want to decide for a given frame \( F \) if there exists a structure that fits to this frame. This will be done by a reduction to a satisfiability problem for systems of linear inequalities in natural numbers. First (Lemma 2) we show that if a structure precisely fits to a frame then the frame induces a satisfiable system of inequalities. Second (Lemma 4) we show that if there exists a frame with satisfiable induced system of inequalities then we can construct a structure that fits to the frame.

**Definition 15** (\( \lambda_F \)). Let \( F = \langle K, ST, \Xi, \Sigma_e, \Sigma, f, c \rangle \) be a frame. Let \( \sigma_1, \ldots, \sigma_l \) be an arbitrary enumeration of star types from ST and let \( m = |f| \). Define \( \lambda_F \) as the system of (in)equalations over variables \( w_1, \ldots, w_l \) such that

1) for each 1-type \( \kappa \in \pi(ST)_{|\Sigma} \)

- if \( \kappa \in K \) then there is an equation

\[
w_1 = 1
\]
where $i$ is the index of the (unique, by point 1 of Definition 12) star type $\sigma_i \in \text{ST}$ satisfying $\pi(\sigma_i)_{|\Sigma} = \kappa$.

- otherwise, if $\kappa \in \pi(\text{ST}_{|\Sigma}) \setminus K$ then there is an inequality
\[
\sum_{\{k|\pi(\sigma_k)_{|\Sigma} = \kappa\}} w_k \geq 3mc,
\]

2) for each 1-type $\pi \in \Pi(\text{ST})$ and $i \in 2$ if $r_i(x) \in \pi$ then there is an inequality
\[
\sum_{\{k|\pi(\sigma_k) = \pi\}} w_k \geq 1
\]

3) for each king 1-type $\kappa \in K$ let $\sigma \in \text{ST}$ be (the only) star type such that $\pi(\sigma)_{|\Sigma} = \kappa$. Then for each $i \in M$ such that $\mu_i$ is a message type and $tp_1(\mu_i) = \pi(\sigma)$ we have
\[
\sum_{\{j|\sigma_j(i) = 1\}} w_j = \sigma[i]
\]

where $i'$ is such that $\mu_i' = (\mu_i)^{-1}$, and

4) for each invertible 2-type $\mu_i$, $i \in M^*$, there is an equation
\[
\sum_{\{k|\sigma_k(i) = 1\}} w_k = \sum_{\{k|\sigma_k(j) = 1\}} w_k
\]

where $\mu_j = \mu_j^{-1}$.

The equations and inequalities in point 1 of Definition 15 express that the underlying structure is $3mc$-differentiated. The equation in point 2 says that each root 1-type is realized. Point 3 says that the number of edges labeled by a message type $\mu_i$ originating in a king is the number of edges labeled by $\mu^{-1}$ finishing in this king. Finally, the equation in point 4 expresses that the number of edges labeled by an invertible message type $\mu_i$ is the same as the number of edges labeled by $\mu^{-1}$. Note that by chromaticity assumption there are no two edges labeled by the same invertible message type originating in the same element, so the number of such edges is the same as the number of their origins.

It is not surprising that the system of inequalities constructed from a model of the input formula is satisfiable. Formally we have the following lemma.

**Lemma 2.** Let $\mathcal{F} = \langle K, \text{ST}, \Xi, \Sigma_c, \Sigma, \bar{f}, c \rangle$ be a frame. If there exists a $3mc$-decorated structure in $\mathbb{F}_d(\Sigma_c, s_1, s_2)$ that precisely fits to $\mathcal{F}$ then $\lambda_\mathcal{F}$ is satisfiable in $\mathbb{N}$.

A more difficult problem is the construction of a model from a solution of the system and a frame. Following [21], this is done in several steps. We first construct the elements of the structure — this is easy because the solution of the system of inequalities tells us exactly how many elements of each type we should use; in particular we know which elements are supposed to be kings.

Then we have to construct 2-types connecting each pair of elements. We start by constructing invertible message types. Since we know star types of all elements, the equation in point 4 of the system of inequalities gives a simple naive way of constructing invertible message types.

This naive construction (done still with the use of the extended signature) may introduce cycles of tree 2-types, but it ensures that all nodes labeled as roots are indeed roots of trees, and that the union of the extended types of all roots contains all tree 2-types occurring in the structure. At this point we erase from the constructed structure the predicates from the extended signature, and proceed with the surgery described in Section III-A. Formally, this is done with Lemma 3 below.

By a partial structure we mean a (usually not complete) graph whose nodes are labeled with 1-types and edges are labeled with invertible 2-types such that whenever an edge $\langle e_1, e_2 \rangle$ is labeled with $\tau$ then $e_1$ is labeled with $tp_1(\tau)$ and $e_2$ is labeled with $tp_2(\tau)$. Intuitively, partial structures are graphs that can be extended to full relational structures by adding edges labeled with noninvertible message types and silent types.

**Lemma 3.** Let $\mathcal{M}$ be a finite partial chromatric structure over $\langle \Sigma, ST \rangle$ such that for both $i \in 2$

- $\mathcal{M}_{s_i}$ is a disjoint union of trees and tree-like cycles, and
- for all tree, 2-types $\tau$ realized in $\mathcal{M}_{s_i}$, the 2-type $\tau$ is also realized in some tree in $\mathcal{M}_{s_i}$.

Then $\mathcal{M}$ can be transformed to a partial structure $\mathcal{M}'$ over the same universe, such that for both $i \in 2$ the substructure $\mathcal{M}'_{s_i}$ is a forest. Moreover, for each node the transformation preserves the set of labels of edges originating in this node.

A sketch of proof of this lemma is given in Section III-A in the paragraph on the sixth observation. Unfortunately, we cannot extend Lemma 3 to deal with three forests $\mathcal{M}_{s_1}, \mathcal{M}_{s_2}$ and $\mathcal{M}_{s_3}$; in that case some cycle $C$ made of edges from $\mathcal{M}_{s_1}$ may also be composed of two edge-disjoint paths, one from $\mathcal{M}_{s_2}$ and the other from $\mathcal{M}_{s_3}$. Now, implanting an edge from $C$ into a tree from $\mathcal{M}_{s_1}$ could increase the number of cycles in $\mathcal{M}_{s_2}$ or in $\mathcal{M}_{s_3}$. Even worse, the created cycle may detach important edges from trees in $\mathcal{M}$ therefore wrecking the property that every tree 2-type realized in the structure is also realized in some tree.

After constructing invertible message types we construct the remaining noninvertible message types and silent types. The construction of noninvertible message types involves the circular witnessing scheme from [7]. The whole construction is summarized by the following lemma.

**Lemma 4.** Let $\mathcal{F} = \langle K, \text{ST}, \Xi, \Sigma_c, \Sigma, \bar{f}, c \rangle$ be a frame. If $\lambda_\mathcal{F}$ is satisfiable in $\mathbb{N}$ then there exists a structure $\mathcal{A} \in \mathbb{F}_d(\Sigma_c, s_1, s_2)$ that fits to $\mathcal{F}$.

We are now ready for the main theorem of this section.

**Theorem 1.** A formula $\varphi \in C^2$ in normal form is satisfiable in $\mathbb{F}_d(\Sigma_c, s_1, s_2)$ if and only if there exists a frame $\mathcal{F}$ such that $\lambda_\mathcal{F}$ is satisfiable in $\mathbb{N}$ and $\mathcal{F} \models \varphi$.

The theorem above suggests the following decision procedure for the satisfiability problem of $C^2$ in the class $\mathbb{F}_d(\Sigma, s_1, s_2)$. Without loss of generality we can assume that the input formula is in normal form (otherwise we can bring
the input formula to a normal form and check if before normalization it had a small model of size bounded by \(c\).

For a given formula \(\varphi\) in normal form simply guess a frame \(F\) such that \(F \models \varphi\) and then guess a solution of \(\lambda_F\). The size of a frame is potentially triply exponential in \(|\varphi| + \lceil \log d \rceil\).

However, a careful choice of the signature \(\Sigma_e\) decreases it to doubly exponential. Then Lemma 2 in [21] implies that whenever \(\lambda_F\) is solvable, it has a solution with only exponentially many nonzero values, which shows that we may restrict our attention to frames and solutions of exponential size. In consequence we have the following theorem.

**Theorem 2.** The satisfiability problem of \(C^2\) over \(F_d(\Sigma, s_1, s_2)\) is \(\text{NEXPTIME}\)-complete.

### IV. Conclusion and Future Work

We have shown that the satisfiability problem for the two-variable logic with counting quantifiers (\(C^2\)) interpreted over \(F_d(\Sigma, s_1, s_2)\) is \(\text{NEXPTIME}\)-complete. To our knowledge this is the first significant extension of the logic \(C^2\) that goes beyond the first-order logic and remains decidable.

Our result implies for example that the logic \(C^2\) with two linear orders where the order is accessed only by the successor relation is decidable in \(\text{NEXPTIME}\). We conjecture that \(C^2\) with one linear order is also decidable, but with much higher complexity.

The ability to express trees or (successor relations of) linear orders can significantly increase the complexity of a logic. While the satisfiability problem for \(FO^2\) is \(\text{EXPSPACE}\)-complete, it becomes \(\text{EXPSPACE}\)-complete in presence of two linear orders [10] and undecidable with three [11].

The satisfiability problem for the universal fragment of the two-or three-variable logic (\(3^*\exists^*\forall^2\) and \(\exists^*\forall^3\)) is decidable in \(\text{NP}\).

However, by a modification of the hardness result from [5] one can show that even the \(\exists^*\forall^2\) fragment with a successor relation of a finite linear order is \(\text{NEXPTIME}\)-hard.

It is also not difficult to show that \(\exists^*\forall^3\) with that extension is undecidable. In contrast, as we have shown, expressing trees does not increase the complexity of \(C^2\).

We expect some other extensions of \(C^2\) to be decidable. One of them is \(C^2\) over the class of structures containing unranked forests \(F(\Sigma, s_1, s_2)\) (that is \(F_d(\Sigma, s_1, s_2)\) without the restriction on the degree of trees). It would be interesting to investigate the connections of this problem with XML reasoning. Another interesting possibility is adding linear inequalities to logic — it seems that it costs nothing because we use inequalities as the backend for the logic, but it should allow some quantitative reasoning about models.

Another possible extension of \(C^2\) would be to relax the requirement that distinguished predicates are interpreted as successor relations in trees and to replace it with a condition that the induced substructure is acyclic. In a separate paper (a forthcoming PhD thesis) the second author shows that this problem is much harder: already with one acyclic relation it is at least as hard as reachability in vector addition systems.

### ACKNOWLEDGMENT

The authors would like to thank Ian Pratt-Hartmann and Emanuel Kieronski for fruitful discussions and anonymous referees for their comments.

### References


APPENDIX A

CHROMATIC, Z-DIFFERENTIATED STRUCTURES WITH RIGID FORESTS

We will need an assumption that models of $C^2$ formulas we consider are structures with rigid forests. Fortunately, we can stick to $C^2$ formulas in normal form that have only such models.

Lemma 5. Let $\varphi$ be a $C^2$ formula in normal form. There exists a $C^2$ formula $\varphi'$ such that $\varphi$ and $\varphi'$ are equisatisfiable in $E_2(s_1, s_2)$, and every model of $\varphi'$ is a structure with rigid forests. Moreover, $\varphi'$ is in normal form, and its size is linear in $|\varphi| + \log d$.

Proof: Let $\varphi$ be a formula over $\langle \Sigma, \{f_1, \ldots, f_m\} \rangle$ in normal form (1). For both $i \in 2$ we introduce fresh binary predicates $\text{parent}_i$ and $\text{child}_i$, intended to point from a child to its parent in a forest defined by $s_i$, and from a parent to each of its children respectively. We then add to $\varphi$ the following conjuncts that describe bounds for the introduced predicates: $\bigwedge_{i=1}^2 \forall x \exists y. (\text{parent}_i(x, y) \land x \neq y)$ and $\bigwedge_{i=1}^2 \forall x \exists y. (\text{child}_i(x, y) \land x \neq y)$. Moreover we update the formula $\alpha$ by adding a conjunct $\bigwedge_{i=1}^2 (s_i(x, y) \leftrightarrow \text{child}_i(x, y) \land \text{parent}_i(y, x))$. Call the obtained formula $\varphi'$. Clearly, $\varphi'$ is a formula in normal form over the classified signature $\Sigma_i \cup \{\text{parent}_i, \text{child}_i\}_{i=1}^2$, and the size of $\varphi'$ is linear in $|\varphi| + \log d$.

To argue that every model $\mathcal{M}'$ of $\varphi'$ is a structure with rigid forests take an arbitrary tree type $\tau$. There are two cases. The first is when $\models \tau \rightarrow s_i(x, y)$, for some $i \in 2$. Then $\models \tau \rightarrow \text{child}_i(x, y) \land \text{parent}_i(y, x)$. The second is when $\models \tau \rightarrow s_i(y, x)$. Then $\models \tau \rightarrow \text{child}_i(y, x) \land \text{parent}_i(x, y)$. In both cases the arrangement of $\text{parent}_i$ and $\text{child}_i$ predicates confirms that $\tau$ is an invertible message type.

To show that $\varphi$ and $\varphi'$ are equisatisfiable, observe first that every model $\mathcal{M}'$ of $\varphi'$ is also a model of $\varphi$. Second, notice that any model $\mathcal{M} \in E_2(s_1, s_2)$ of $\varphi$ can be expanded to a model $\mathcal{M}' \in E_2(s_1, s_2)$ of $\varphi'$ by interpreting $\text{parent}_i$ and $\text{child}_i$ as parent and child relations in forest $s_i$, for both $i \in 2$.

The following two lemmas show that every satisfiable $C^2$ formula has a model which is chromatic and $Z$-differentiated, for any natural number $Z$. Recall that every model of a formula $\varphi$ in normal form is a $c$-bounded structure, where $c$ is maximal value of the bounds in counting quantifiers in $\varphi$. Both lemmas come from and are proved in [20].

Lemma 6 ([20], Lemma 2). Let $\mathcal{A}$ be a $c$-bounded structure over $\langle \Sigma, f \rangle$ and let $\mathcal{A}'$ be a signature formed by adding $[\log((|f|e)^2)]$ new unary predicates to $\Sigma$. Structure $\mathcal{A}$ can be expanded to a chromatic structure $\mathcal{A}'$ over $\langle \Sigma', f \rangle$.

Lemma 7 ([20], Lemma 3). Any structure $\mathcal{A}$ can be made $Z$-differentiated by interpreting $[\log Z]$ new unary predicates.

We will use the above lemma for $Z = 3mc$, where $m$ and $c$ are numbers defined by the formula $\varphi$. Notice that properties of being a rigid and a chromatic structure are preserved when new unary predicates are interpreted in the structure.

Therefore we may restrict our attention to $C^2$ formulas $\varphi$ in normal form with chromatic and $3mc$-differentiated models with rigid forest. Moreover these structures are over signatures of sizes polynomial in the size of $\varphi$.

APPENDIX B

TYPING LEMMAS

Before proving Lemma 1 we need one more lemma. Let $\mathcal{A} \in E_2(\Sigma_1, s_2, s_3)$ be a bichromatic and $3mc$-differentiated structure over $\langle \Sigma_1, f \rangle$, let $i \in 2$ let $e$ be any node in $A_{s_i}$ and let $T = T^i_3(e)$ be the tree rooted at $e$ in $A_{s_i}$. Define $\Phi(T)$ to be the set of all 2-types realized in $T$.

We say that a node $e$ in $B$ over the signature $\langle \Sigma_e, f \rangle$ is well-typed in forest $i$ with respect to $\Phi$ if $e$ is well-typed in forest $i$ and $B \models \text{type}_{\Phi}(e)$. The following lemma says that it is possible to type a single tree in a structure from $E_2(\Sigma_1, s_2, s_3)$ using a small number of type predicates.

Lemma 8. Let $\mathcal{A} \in E_2(\Sigma_1, s_2, s_3)$ be a bichromatic and $3mc$-differentiated structure over $\langle \Sigma_1, f \rangle$, let $i \in 2$ let $e$ be any node in $A_{s_i}$ and let $\Phi$ be any subset of $\Phi(T^i_3(e))$. Then there exists an extension $\mathcal{B}$ of the structure $\mathcal{A}$ to a signature $\langle \Sigma_e, f \rangle$ where $\Sigma_e \subseteq \Sigma_{\text{ext}}$ such that $e$ is well typed in forest $i$ with respect $\Phi$. Moreover $|\Sigma_e \setminus \Sigma| \leq \max(2|\Phi|, 1)$.

Proof: The proof proceeds by induction on the structure of $T = T^i_3(e)$.

Base case: $T$ is a single node. Then $\Phi(T)$ is an empty set, and thus $\Phi = \emptyset$. In order to obtain a structure $\mathcal{B}$ it is enough to label $e$ with predicate type$\Phi$($\cdot$).

Inductive case: If $\Phi = \emptyset$ then we obtain $\mathcal{B}$ by labeling each node in $T$ by predicate type$\Phi$($\cdot$). If $\Phi \neq \emptyset$ then let $\{e_j\}_{j=1}^k$ be immediate successors of $e$ in $A_{s_i}$, let $\tau_j = \text{tp}^3(e, e_j)$ and $T_j = T^i_3(e_j)$ for $j \in \textbf{k}$. We have

$$\Phi(T) = \bigcup \{\Phi(T_j)\}_{j=1}^k \cup \{\tau_1, \ldots, \tau_k\}$$

Since $\Phi \subseteq \Phi(T)$ there exists a sequence $\{\tau_j\}_{j=1}^k$ of pairwise disjoint sets such that $\Phi_j \subseteq \Phi(T_j)$, for $j \in \textbf{k}$ and

$$\Phi \setminus \{\tau_1 \ldots \tau_k\} = \bigcup_{j=1}^k \Phi_j.$$

We define the structure $\mathcal{B}$ by labeling $e$ with the predicate type$\Phi$($\cdot$) and using induction to well type $e_j$ in forest $i$ with respect to $\Phi_j$, for $j \in \textbf{k}$. It is easy to check that $e$ becomes well typed in forest $i$ with respect to $\Phi$.

Now we need justify that the number of distinct predicates type$\Phi$($\cdot$) that were used in the above construction is at most
2|Φ|. There are two cases. If Φj = ∅ for every j ∈ k
then the number of predicates used is two, that is at most
2|Φ|. Otherwise, let d1, . . . , dk be the maximal subsequence
of 1, . . . , k such that Φdi ̸= ∅ for j ∈ l. Notice that
predicate type ϕi of predicate oj occurs in each tree Tdi as a label of leaf
nodes. By the inductive assumption the number of distinct
type predicates used when typing successors of e is at most
n_{succ} = (k−1)2|Φi|−l + 1. We count here the predicate
type ϕi only once.

There are two cases. First, if l > 1 then the number of distinct
type predicates used when typing T is n_{succ} + 1, that is
at most 2|Φ| − l + 2, which is at most 2|Φ|. The maximum value
can be achieved when l = 2. Second, if l = 1 then either Φ =
Φi or Φi ⊆ Φ. In the former case no new type predicate is
introduced when typing the node e, and the number of distinct
type predicates used when typing T is just n_{succ}, which in this
case equals 2|Φ|. In the latter case n_{succ} equals 2|Φi|, which
is strictly less than 2|Φ| and the number of predicates used
when typing T is n_{succ} + 1, what is at most 2|Φ|.

As a consequence we can type all trees in a structure from
Ex(Σ, s1, s2) using a small number of type predicates and we
can guarantee that all 2-types of edges in the forest occur in
labels of the roots of the forest.

A. Proof of Lemma 1

Proof: Let A ∈ Ex(Σ, s1, s2) be a chromatic and 3mc-
differentiated structure over (Σ, f). For a fixed i ∈ L all trees
in A, are disjoint and can be typed separately as follows. Let
{erj}j=1 be the sequence of all tree roots in A. Let for every
j ∈ L let Tj = Tj,A (erj) and let Φ(Tj) be the set of all 2-types realized
in tree Tj. Take an arbitrary sequence {Φj}j=1 of pairwise
disjoint sets such that Φj ⊆ Φ(Tj) and
\[
\bigcup_{j=1}^{k} \Phi(T_j) = \bigcup_{j=1}^{k} \Phi_j.
\]

Using Lemma 8 we can make each root node erj well typed
in forest i with respect to Φj. The number of type predicates
used is at most max(2|Φj|, 1), for every j ∈ L. Recall that
\[
\bigcup_{j=1}^{k} \Phi(T_j)
\]
is the set of all tree 2-types realized in A, that is Φ(A). By reasoning similar to that in Lemma 8 it follows,
that total number of type predicates used when typing all
root nodes is at most max(2|Φ(A)|, 1). Moreover we can type
both forest separately and therefore obtain a well typed
structure B over (Σe, f), where Σe is the sum of extended
signatures used when well typing both forests. Therefore
\[
|Σ_e \setminus Σ| \leq 2(|Φ(A)| + |d^2(A)| + 1).
\]

APPENDIX C
PROOF OF PROPOSITION 1

The two statements of the proposition are proved with
Lemmas 10 and 11 below. The following lemma is used in
proof of Lemma 10.

Lemma 9. Let A be a c-bounded structure over (Σ, T). Then
for any two 1-types π1, π2 both realized at least 2|f|c + 2
times in A there exist a silent 2-type τ and distinct e, e′ ∈ A
such that tp^A(e) = π1, tp^A(e′) = π2 and tp^A(e, e′) = τ.

Proof: Let m = |f|. There are two cases.

• (π1 = π2). Let A^i be the substructure of A induced by
elements of type π1 and let k be the cardinality of A^i. The
number of edges that realize message types in A^i is bounded
by m(k−1). By assumption k > 2mc + 1 it follows that some
edges are labeled by silent types.

• (π1 ̸= π2). Let k = 2mc + 1 and let S_π be a subset of
A that contains k elements of type π1 and let S_π be the subset
containing k elements of type π2. Consider the substructure
A_π,|π_2| induced by the set S_π ∩ S_π. The number of edges in A_π,|π_2|
that realize noisy types starting in S_π and ending in S_π is bounded
by 2km. The number of all edges starting in S_π and ending
in S_π is k^2. Since k^2 > 2km, some of these edges realize
silent types.

Lemma 10. Let φ be a C^2 formula in normal form over a
vocabulary (Σ, T). Let A be a 3mc-decorated structure in
Ex(Σ, s1, s2). If A |= φ then there exists a frame F, such
that A precisely fits to F and F |= φ.

Proof: Let A ∈ Ex(Σe, s1, s2) be a 3mc-decorated structure
over (Σe, f) such that A |= φ and m = |f|. Define a frame F to be
⟨K, ST, Ξ, Σe, Σ, f, c⟩, where Ξ is the set of all 1-types of kings in A_{Σe},
ST is the set of all star types realized in A and Ξ is the set of all non-king silent 2-types realized in A_{Σe}. Note that since A is decorated, the set ST is a well
typed set of rigid and chromatic star types over (Σe, f), such
that their restrictions to Ξ are also chromatic.

We will first check that F is indeed a frame. Structure A is
3mc-decorated, so it is chromatic. Thus ST is a chromatic set
of star types. Tuple F satisfies conditions 1–4 in Definition 12
which can be checked by inspection. Condition 1 is obvious
by construction. Condition 2 is also quite obvious: Structure
A_{Σe} is a model of φ and thus it is c-bounded, moreover it is
3mc-decorated. Therefore A_{Σe} is a 3mc-differentiated and
2 types of edges in the forest occur in
labels of the roots of the forest.

A. Proof of Lemma 1

Proof: Let A ∈ Ex(Σ, s1, s2) be a chromatic and 3mc-
differentiated structure over (Σ, f). For a fixed i ∈ L all trees
in A, are disjoint and can be typed separately as follows. Let
{erj}j=1 be the sequence of all tree roots in A. Let for every
j ∈ L let Tj = Tj,A (erj) and let Φ(Tj) be the set of all 2-types realized
in tree Tj. Take an arbitrary sequence {Φj}j=1 of pairwise
disjoint sets such that Φj ⊆ Φ(Tj) and
\[
\bigcup_{j=1}^{k} \Phi(T_j) = \bigcup_{j=1}^{k} \Phi_j.
\]

Using Lemma 8 we can make each root node erj well typed
in forest i with respect to Φj. The number of type predicates
used is at most max(2|Φj|, 1), for every j ∈ L. Recall that
\[
\bigcup_{j=1}^{k} \Phi(T_j)
\]
is the set of all tree 2-types realized in A, that is Φ(A). By reasoning similar to that in Lemma 8 it follows,
that total number of type predicates used when typing all
root nodes is at most max(2|Φ(A)|, 1). Moreover we can type
both forest separately and therefore obtain a well typed
structure B over (Σe, f), where Σe is the sum of extended
signatures used when well typing both forests. Therefore
\[
|Σ_e \setminus Σ| \leq 2(|Φ(A)| + |d^2(A)| + 1).
\]

APPENDIX C
PROOF OF PROPOSITION 1

The two statements of the proposition are proved with
Lemmas 10 and 11 below. The following lemma is used in
proof of Lemma 10.
Proof: Let $\varphi$ be in normal form (1). Assume that there exists a frame $F$ and a 3mc-decorated structure $A$ such that $A \models \varphi$. Let $\mu$ be any 2-type realized in $A_{\Sigma}$. Then by Definition 14 we have the following alternative: $\mu \in \pi(\Sigma)^{ST}_{\Sigma} \lor \mu^{-1} \in \pi(\Sigma)^{ST}_{\Sigma}$ or $\mu$ is a non-king silent type and $\mu \in \Xi$. By Definition 13 it follows that $\models \mu \rightarrow \alpha$. So $A \models \forall x \forall y.(\alpha \lor x = y)$.

Since $A$ fits to $F$ and $F \models \varphi$, it also follows that for each star type $\sigma$ realized in $A$ and each $h$ such that $1 \leq h \leq m$ we have

$$( \sum_{i \in h}(\sum_{x,y} \sigma[i]) ) \leq h \cdot C_h,$$

and thus $A \models \forall 1 \leq h \leq m \forall x \exists y.\pi^{-1}(x,y)(f_h(x,y) \land x \neq y)$. Hence $A \models \varphi$ as required.

**Appendix D**

**Description Lemmas**

**A. Proof of Lemma 2**

Proof: Let $A$ be a 3mc-decorated structure in $\mathbb{F}_d(\Sigma_{\alpha}, s_1, s_2)$ that precisely fits to $F$. Let $\sigma_1, \ldots, \sigma_I$ be a standard enumeration of ST and let $n_i, 1 \leq i \leq I$ be the number of star types $\sigma_i$ realized in $A$. We show that $n_1, \ldots, n_I$ is a solution of $\lambda_F$. The (in)equations defined in point 1 of Definition 15 are satisfied, because the structure $A_{\Sigma'}$ is 3mc-differentiated. The equation from point 2 of Definition 15 is satisfied because $A$ precisely fits to $F$. Every $\sigma \in ST$ is realized in $A$ and thus, if $\pi_i(x) \in \pi(\sigma)$ for any $i \in \{1, 2\}$ then we know that $\pi(\sigma)$ is realized at least once in $A$. Point 3 is obvious by construction: for any king type $\kappa$ and star type $\sigma$ realized in $A$ such that $\pi(\sigma) = \kappa$ and any message type $\mu_i$ there exists precisely $\sigma[i]$ elements of $A$ that want to accept an edge of type $\mu_i$. To see that the equations in point 4 of this definition are also satisfied, recall that $A$ is a chromatic structure. In particular, for each invertible 2-type $\mu_i$ realized in $A$, $\mu_i$ and $\mu_i^{-1}$ are distinct and also no star type $\sigma \in ST$ contains more than one occurrence of type $\mu_i$ (that is if $\sigma[i] > 0$ then $\sigma[i] = 1$). Then the number of types $\mu_i$ realized in structure $A$ is the number of star types $\sigma$ realized in $A$ such that $\sigma[i] = 1$, which is $\sum_{k}[\sigma[i]=1] nk$. Let $\mu_j$ be $\mu_i^{-1}$. Since $A$ is a relational structure, the above sum is equal to the number of types $\mu_j^{-1}$ realized in $A$, which is $\sum_{k}[\sigma[j]=1] nk$. So $n_1, \ldots, n_I$ satisfy $\lambda_F$.

Recall that we consider signatures $(\Sigma, \pi)$ containing symbols of arity at most two and that every relational structure over such signatures can be seen as a complete directed graph, where nodes are labeled by 1-types and edges are labeled by 2-types. Moreover, 1-types that label nodes of $M$ math 1-types on ends of 2-types emanating from nodes.

By a partial structure we mean a (usually not complete) graph whose nodes are labeled with 1-types and edges are labeled with invertible 2-types such that whenever an edge $(e_1, e_2)$ is labeled with $\tau$ then $e_1$ is labeled with $tp_1(\tau)$ and $e_2$ is labeled with $tp_2(\tau)$. Intuitively, partial structures are graphs that can be extended to full relational structures by adding edges labeled with noninvertible message types and silent types.

**B. Proof of Lemma 3**

Proof: We will show that the number of cycles in $M_{s_1}$ and $M_{s_2}$ can be decreased while preserving the assumptions of the Lemma. To break a cycle in $M_{s_1}$ it is enough to remove one of its edges, but (in order to preserve the sets of edge labels) we cannot just erase it and some model surgery is required here.

Without loss of generality we can assume that $i = 1$ (the case of forest no. 2 is symmetric). We will remove cycles by implanting them into trees in $M_{s_1}$. Let $C = \{e_1, \ldots, e_2\}$ be a sequence of pairwise distinct nodes in $M_{s_1}$ forming a simple cycle in some tree-like cycle in $M_{s_1}$. That is, for every $j \in I-1$ we have $M \models s_1(e_j, e_{j+1})$ and additionally $M \models s_1(e_1, e_1)$. There are two cases.

1) For some $k \in I-1$ there exists an edge $\langle e_k, e_k' \rangle$ of the cycle $C$ that does not belong to $M_{s_2}$. To simplify notation and to remain in accordance with Figure 2 let us assume that $k=1$. That is $M \models \neg s_2(e_1, e_2) \land \neg s_2(e_2, e_1)$. Denote $tp^M(e_1, e_2)$ by $\tau$. Because $\tau$ is a tree, 2-type realized in $M$, there exists a tree $T$ in $M_{s_1}$ and an edge $(h_1, h_2)$ in $T$, such that $tp^M(h_1, h_2) = \tau$. We will implant cycle $C$ into the tree $T$. Because partial structure $M$ is chromatic and $tp^M(e_2) = tp^M(h_2)$, the 2-type $tp^M(e_1, h_2)$ is not yet defined; the same holds for $tp^M(h_1, e_2)$. We remove edges $(e_1, e_2)$ and $(h_1, h_2)$ from $M$, add to it edges $\langle e_1, h_2 \rangle$ and $\langle h_1, e_2 \rangle$. Then we label the edges $\langle e_1, h_2 \rangle$ and $\langle h_1, e_2 \rangle$ by type $\tau$ as shown on Figure 2. Notice that we have broken one cycle in $M_{s_1}$, we have not created another one in $M_{s_2}$, and that the structure $M_{s_2}$ was not touched during the surgery. Tree $T$ has grown and it still contains at least these edges that were present in it before the surgery. Moreover, we have preserved the sets of labels of edges originating in every node. It follows that the number of cycles is decreased and assumptions of the lemma are preserved.

2) Otherwise, every edge of the cycle $C$ in $M_{s_1}$ is also an edge in $M_{s_2}$. The surgery is the same as in the previous case, but the argument that no new cycles are introduced is a bit more involved.

Because $C$ forms an undirected cycle in $M_{s_2}$ it also forms a simple (directed) cycle in some tree-like cycle in $M_{s_2}$. It is because the assumption that every cycle in $M_{s_2}$ is a tree-like cycle implies that every cycle present there is directed. Take the edge $(e_1, e_2)$ and denote $tp^M(e_1, e_2)$ by $\tau$. There are two cases, either $s_2((y, x) \in \tau$, and the cycle in $M_{s_2}$ has the same direction as the cycle in $M_{s_1}$, or $s_2(y, x) \in \tau$ and both cycles have opposite directions. In both cases the surgery proceeds as in previous case and is shown on Figure 3. Recall that $\tau$ is a 2-type of an edge $(h_1, h_2)$, selected from tree $T$ during surgery. The edge belongs to both $M_{s_1}$ and $M_{s_2}$. It is then replaced by a simple path made of nodes...
from $C$. Moreover, directions of binary predicates $s_i$ are the same in $\tau$ and on the path, for both $i \in 2$. This way we have implant the cycle $C$ into the tree $T$ in $M_{s_1}$ thus decreasing the number of cycles in $M_{s_1}$. We also break the corresponding cycle in $M_{s_2}$, and either implant it into a tree or into another, existing cycle in $M_{s_2}$ thus decreasing the number of cycles in $M_{s_2}$, or we create a new cycle in $M_{s_2}$ that contains all nodes of $C$ but is not a cycle in $M_{s_1}$. We thus decrease the number of cycles in $M_{s_1}$ and do not increase the number of cycles in $M_{s_2}$. As previously, tree $T$ has grown and the sets of labels of edges are preserved. Therefore the conditions of the lemma are preserved.

By repeating the above operation for all cycles in $M_{s_1}$ and $M_{s_2}$ we obtain a substructure $M'$ where $M'_{s_1}$ and $M'_{s_2}$ are forests. Moreover the universe of the structure $M$ and the sets of labels of edges are preserved.

Unfortunately, we cannot extend Lemma 3 to deal with three forests $M_{s_1}, M_{s_2}$ and $M_{s_3}$; in that case some cycle $C$ made of edges from $M_{s_1}$ may also be composed of two edge-disjoint paths, one from $M_{s_2}$ and the other from $M_{s_3}$. Now, implanting an edge from $C$ into a tree from $M_{s_1}$ could increase the number of cycles in $M_{s_2}$ or in $M_{s_3}$. Even worse, the created cycle may detach important edges from trees in $M$ therefore wrecking the property that every tree 2-type realized in the structure is also realized in some tree.

C. Proof of Lemma 4

Proof: We construct a structure $A$ over signature $\langle \Sigma, \bar{f} \rangle$ such that $K$ is the set of all king 1-types of $A$, the set of star types realized in $A$ is a subset of $ST|_{\Sigma}$ and the set of non-king silent 2-types in $A$ is a subset of $\Xi$. In first two steps we build a partial structure $B$ over the signature $\langle \Sigma, \bar{f} \rangle$. Then we restrict $B$ to vocabulary $\Sigma$ and continue construction of a structure $A$ over $\langle \Sigma, \bar{f} \rangle$. Let $\sigma_1, \ldots, \sigma_l$ be a standard enumeration of $ST$ used in the definition of $\lambda^e$ and let $n_1, \ldots, n_l$ be a natural solution of $\lambda^e$. Let $U_1, \ldots, U_l$ be disjoint sets of fresh elements such that $|U_i| = n_i$ for $1 \leq i \leq l$. Both $A$ and $B$ are structures over the same universe $U = \bigcup_{i \leq l} U_i$ for any $a \in U$, if $a \in U_i$ then we say that the postulated star type of $a$ is $\sigma_i$, that is the element $a$ “wants” to have the type $\sigma_i$ in $B$ and the type $\sigma_i|_{\Sigma}$ in $A$. We introduce a function $\text{pst} : U \rightarrow ST$ such that $\text{pst}(a)$ is the postulated star type of $a$. We construct the partial structure $B$ in such a way that the star type $st^B(a)$ of $a$ in $B$ will be equal to $\text{pst}(a)$ for all $a \in U$.

Recall that to construct a structure, after defining the universe we have to define 1-types of its elements and 2-types of all pairs of elements. This is done in the following steps.

1) (establishing 1-types) For each $a \in U$ put $\text{tp}^B(a) = \pi(\text{pst}(a))$, that is each $a$ obtains the 1-type defined by its postulated star type. Note that this definition implies in particular that the set of king 1-types of $B|_{\Sigma}$ is $K$.

2) (establishing invertible message types)
For any set of invertible message types $\Upsilon$, let $\Upsilon^{-1}$ be the set containing all inversions of types from $\Upsilon$, that is $\Upsilon^{-1} = \{\mu^{-1} \mid \mu \in \Upsilon\}$. Now fix any set $\Upsilon$ such that $\Upsilon \cap \Upsilon^{-1} = \emptyset$ and $\Upsilon \cup \Upsilon^{-1}$ is the set of invertible types in $\tau(ST)$. Such a set can be found because of chromaticity: there exists no invertible 2-type $\mu \in \tau(ST)$ such that $\mu = \mu^{-1}$.

We iterate the following construction for each pair of invertible message types $\mu_i, \mu_j$ such that $\mu_i \in \Upsilon$ and $\mu_j = \mu_i^{-1}$.

Let $Z_i$ be the set of these elements of $U$ that intend to send message type $\mu_i$,

$$Z_i = \{a \in U \mid \exists \sigma. \text{pst}(a) = \sigma \land \sigma[i] = 1\}$$

and let $Z_j$ be the set of these elements of $U$ that intend to accept message type $\mu_i$, or equivalently — that intend to send message type $\mu_j$.

$$Z_j = \{a \in U \mid \exists \sigma. \text{pst}(a) = \sigma \land \sigma[j] = 1\}.$$ 

By point 4 of Definition 15 it follows that $|Z_i| = |Z_j|$; by chromaticity of $ST$ we have $Z_i \cap Z_j = \emptyset$. Then for each pair of elements $e_1 \in Z_i$, $e_2 \in Z_j$, the 2-type between $e_1$ and $e_2$ was not established in another iteration of the present procedure: otherwise, if $\text{tp}^B(e_1, e_2)$ was established in an iteration $i$, then $e_1$ sends two invertible message types $\mu_i$ and $\mu_j$ with $\text{tp}_2(\mu_i) = \text{tp}_2(\mu_j) = \text{tp}^B(e_2)$, which contradicts the assumption that the star type $st^B(e_1)$ is chromatic. Now we choose an arbitrary bijection $b : Z_i \rightarrow Z_j$ and define $\text{tp}^B(e, b(e)) = \mu_i$ and $\text{tp}^B(b(e), e) = \mu_j$ for every $e \in Z_i$.

3) (clearing type predicates and breaking cycles)
So far we have constructed a partial structure $B$, where all invertible message types are established. For every $e$ in $B$, $\text{pst}(e) = \text{forest}$ and all edges that should form forests are already present in $B$. Therefore all cycles in $B_{s_1}$ and $B_{s_2}$ are tree-like cycles.

Moreover, every root is realized in $B$ at least once, as enforced by condition 2 of $\lambda^e$. Because $ST$ is well typed it follows that all tree, 2-types realized in $B_{s_1}$ are also realized in some tree in $B_{s_1}$, for $i \in 2$. We now get rid of type predicates by defining $A = B|_{\Sigma}$. Structure $A$ satisfies the assumptions of Lemma 3 and it can be transformed in such a way that invertible message types are still properly arranged and there are no cycles in $A_{s_1}$ and in $A_{s_2}$. In other words both forests are fully defined in the present step. In consequence $A$, when fully constructed, will belong to $E_6(\Sigma, s_1, s_2)$.

4) (establishing non-invertible message types)

- We first establish non-invertible message types between pairs $e_k, e$, where $e_k$ is a king in $A$. Let $\sigma = \text{pst}(e_k)$. For any $i$ such that $M^* + 1 \leq i \leq M$ and $\mu_i$ is a non-invertible message type we need to find $\sigma[i]$ elements $e$ of $U$ that want to accept the type $\mu_i$, that is such that $\sigma'[i] = 1$, where $\sigma' = \text{pst}(e)$ and $\mu_i = (\mu_i)^{-1}$. By definition of $\lambda^e$ (point 3 of Definition 15) such elements indeed can be found. Define then $\text{tp}^A(e_k, e) = \mu_i$ and $\text{tp}^A(e, e_k) = \mu_j$ for all such elements $e$. Note that since $\mu_i$ is non-invertible, $\mu_j$ is not a message type. Therefore, even
if $e$ is a king, the procedure described here does not apply to the pair $e, e_k$.

- Now we establish non-invertible message types between pairs $e, e_k$, where $e$ is a non-king but $e_k$ is a king. Let $\sigma = \text{pst}(e)$. For any $i$ such that $M^* + 1 \leq i \leq M$ and $\mu_i$ is a message type and $tp_2(\mu_i) = tp^A(e_k)$ we have $\sigma[i] \in \{0, 1\}$, by definition of star type (Def. 8). Moreover, $\sigma[i] = 1$ for at most one such $i$. Let $e_k$ be the king that realizes type $tp_2(\mu_i)$. If $\sigma[i] = 1$ then we define $tp^A(e, e_k) = \mu_i$ and $tp^A(e_k, e) = (\mu_i)^{-1}$. This again does not collide with the previous point because $(\mu_i)^{-1}$ is not a message type.

- We now establish non-invertible message types between pairs $e, e'$, where neither $e$ nor $e'$ are kings. Let $A_0$ be the structure defined so far, that is the structure with universe $U$ where 1-types, invertible types, non-invertible types between kings and their witnesses and between non-kings and kings have already been assigned. Let $K$ be a substructure of $A$ generated by its kings. Let $D, E$ and $F$ be disjoint substructures of $A_0$ such that

- $D, E, F$ are disjoint from $K$ and $A_0 = D \cup E \cup F \cup K$,

- each of the sets $D, E$ contains exactly $mc$ elements of each non-king 1-type and $F$ contains at least $mc$ elements of each non-king 1-type.

The existence of such structures follows from the second inequality in point 1 of Definition 15, which guarantees at least $3mc$ elements of each non-king 1-type.

Now we define non-invertible types between elements in $D \cup E \cup F$. Since each element of $A$ requires at most $mc$ witnesses, each element of $D \cup E \cup F$ requires at most $mc$ witnesses of any 1-type. In each of the sets $D, E$ and $F$ we have at least $mc$ elements of every non-king 1-type. Therefore we can find witnesses for elements from $D$ in $E$ witnesses for elements from $E$ in $F$ and for elements from $F$ in $D$. By using three different sets we are sure that we never try to assign a 2-type to a pair $e, e'$ of elements such that the type of $e'$, $e$ was already assigned.

5) (establishing silent types)

To finish the construction it is the enough to assign silent types to the remaining pairs of elements of $A$. Let $e, e'$ be any such a pair. There are two cases

- (neither $e$ nor $e'$ is a king). The existence of some silent type $\mu$, such that $tp_1(\mu) = tp^A(e)$ and $tp_2(\mu) = tp^A(e')$ is guaranteed by point 2 of Definition 12. We then define $tp^A(e, e') = \mu$ and $tp^A(e', e) = (\mu)^{-1}$.

- ($e'$ is a king) Let $\text{pst}(e) = \sigma$. By the definition of a star type (condition 3) there exists exactly one $i$ such that $M^* + 1 \leq i \leq M$ and $\sigma[i] = 1$ and $tp_2(\mu_i) = tp^A(e')$. Since all noisy types have already been assigned, it follows that $\mu_i$ is a silent type. We then define $tp^A(e, e') = \mu_i$ and $tp^A(e', e) = (\mu_i)^{-1}$.

Note that the steps 4 and 5 create no cycles in $A_{s_1, s_2}$ because there are no tree 2-types that are silent or non-invertible message types. Thus we have constructed a relational structure $A \in \mathcal{F}_d(\Sigma, s_1, s_2)$ over $\Sigma$ such that the substructure of $A$ generated by its kings realizes all types from $K$, the set of realized star types is a subset of $\mathcal{ST}_{\Sigma}$ (some $n_i$’s may be equal to 0), and the set of realized non-king silent 2-types is a subset of $\Xi$. That is, $A$ fits to $\mathcal{F}$ as required.

**Appendix E**

**Proofs of Theorems 1 and 2**

The following lemma is proved in [21].

**Lemma 12** ([21], Lemma 2). Let $E$ be a set of $m$ linear inequalities of the form $a_0 + a_1 x_1 + \ldots + a_n x_n \leq b_0 + b_1 x_1 + \ldots + b_n x_n$ in variables $x_1, \ldots, x_n$, where $a_0, b_0 \in \mathbb{N}$ and $a_i, b_i \in \{0, 1\}$ for all $i \in \mathbb{N}$. If $E$ has a solution over $\mathbb{N}$, then it has a solution over $\mathbb{N}$ in which at most $3m(\log m + 1)$ variables take non-zero values.

**A. Theorem 1**

$(\Rightarrow)$ Suppose that $\varphi$ is satisfiable. Then there exists a $3mc$-decorated structure $A \in \mathcal{F}_d(\Sigma, s_1, s_2)$ such that $A \models \varphi$. By Lemma 10 there exists a frame $\mathcal{F}$, such that $A$ precisely fits to $\mathcal{F}$ and $\mathcal{F} \models \varphi$. By Lemma 2, $\lambda_\mathcal{F}$ is satisfiable in $\mathbb{N}$.

$(\Leftarrow)$ Suppose that $\mathcal{F}$ is a frame such that $\lambda_\mathcal{F}$ is satisfiable in $\mathbb{N}$ and $\mathcal{F} \models \varphi$. By Lemma 4 there exists a structure $A \in \mathcal{F}_d(\Sigma, s_1, s_2)$ that fits to $\mathcal{F}$. Then $A \models \varphi$ by Lemma 11.

**B. Theorem 2**

The lower bound trivially follows from the NEXPTIME lower bound for $C^2$. A naïve implementation of the decision procedure suggested by Theorem 1 gives a 3NEXPTIME procedure since a potential size of a frame is triply exponential in $|\varphi| + \lceil \log d \rceil$. We decrease this complexity to NEXPTIME in two steps.

First we show that it is enough to search for doubly exponential frames. The sizes of the sets $\Sigma, f$ and the size of $c$ are polynomial in $|\varphi| + \lceil \log d \rceil$ (note that $c$ itself can be exponential in $|\varphi| + \lceil \log d \rceil$, but its size is polynomial). Therefore there are at most exponentially many 1-types and 2-types over $\Sigma$. By Lemma 1 it is enough to guess an exponential signature $\Sigma_c$. In the construction of a frame in Lemma 10 each used 2-type uses at most two predicates of the form type$_f^c$ (one for $x$ and one for $y$) and at most two predicates of the form type$_p^c$. Therefore there are at most exponentially many (in $|\varphi| + \lceil \log d \rceil$) 2-types over $\Sigma_c$ to be guessed. Finally, the size of $\mathcal{ST}$ is exponential in the number of these 2-types, which is doubly exponential in $|\varphi| + \lceil \log d \rceil$.

Checking that $\mathcal{F}$ defines a frame and that its restriction satisfies $\varphi$ is polynomial in the size of $\mathcal{F}$. The size of $\lambda_\mathcal{F}$ is also polynomial in $|\mathcal{F}|$ and thus doubly exponential in $|\varphi| + \lceil \log d \rceil$. Since the satisfiability problem for a set of linear
(in)equations is in NP, it follows that the upper bound for finite satisfiability is $2\text{NEXPTIME}$.

In the second step, using methods from [21], we give an exponential speedup in our procedure. We start by observing that $\lambda \mathcal{F}$ contains only exponentially many (in)equations (the number of equations of types 1–3 is bounded by three times the number of 1-types and the number of equations of type 4 is bounded by the number of 2-types) and that the doubly exponential size of $\lambda \mathcal{F}$ is caused only by the doubly exponential number of variables. Moreover, Lemma 12 implies that if $\lambda \mathcal{F}$ is satisfiable, then it is enough to guess only exponentially many variables among $w_1, \ldots, w_I$ that take nonzero values. This means that we have to guess only exponentially many star types, so the size of the guessed frame is exponential. This reduces the complexity of our procedure to NEXPTIME.