Decidability of predicate logics with team semantics ⋆

Juha Kontinen[1], Antti Kuusisto[2], and Jonni Virtema[1,3]

[1] University of Helsinki, Finland
[2] University of Bremen, Germany
[3] Leibniz Universität Hannover, Germany
{juha.kontinen, jonni.virtema}@helsinki.fi
kuusisto@uni-bremen.de

Abstract. We study the complexity of predicate logics based on team semantics. We show that the satisfiability problems of two-variable independence logic and inclusion logic are both *NEXPTIME*-complete. Furthermore, we show that the validity problem of two-variable dependence logic is undecidable, thereby solving an open problem from the team semantics literature. We also briefly analyse the complexity of the Bernays-Schönfinkel-Ramsey prefix classes of dependence logic.

1 Introduction

The satisfiability problem of *two-variable logic* FO^2 was shown to be *NEXPTIME*-complete in [9]. The extension of two-variable logic with counting quantifiers, FOC^2, was proved decidable in [10,21], and it was subsequently shown to be *NEXPTIME*-complete in [22]. Research on extensions and variants of two-variable logic is currently very active. Recent research efforts have mainly concerned decidability and complexity issues in restriction to particular classes of structures and also questions related to different built-in features and operators that increase the expressivity of the base language. Recent articles in the field include for example [1], [4], [13], [16], [23], and several others.

In this article we study two-variable fragments of logics based on *team semantics*. Team semantics was originally conceived in [15] in the context of *independence friendly* (IF) *logic* [14]. In [24], Väänänen introduced *dependence logic*, which is a novel approach to IF logic based on new atomic formulas $=(x_1, ...x_k, y)$ stating that the interpretation of the variable y is functionally determined by the interpretations of the variables $x_1, ..., x_k$.

After the introduction of dependence logic, research on logics based on team semantics has been active. Several different logics with different applications have been suggested. In particular, team semantics has proved to be a powerful framework for studying different kinds of *dependency notions*. *Independence logic* [11] is a variant of dependence logic that extends first-order logic by new atomic formulas $x_1, ..., x_k \perp y_1, ..., y_l$ with the intuitive meaning that the interpretations of the variables $x_1, ..., x_k$ are informationally independent of the interpretations of the variables $y_1, ..., y_l$. *Inclusion logic* [6] extends first-order logic by atomic formulas $x_1, ..., x_k \subseteq y_1, ..., y_k$, whose intuitive meaning is that tuples interpreting the variables $x_1, ..., x_k$ are also tuples interpreting $y_1, ..., y_k$. Currently dependence, independence and inclusion logics are the three most important and most widely studied systems based on team semantics.

Both dependence logic and independence logic are equiexpressive with existential second-order logic (see [24], [11]), and thereby capture NP. Curiously, inclusion logic is equiexpressive with *greatest fixed point logic* (see [7]), and thereby characterizes P on finite ordered models. While the descriptive complexity of most known logics based on team semantics is understood reasonably well, the complexity of related satisfiability problems has received somewhat less attention. The satisfiability problem of the two-variable fragment of dependence logic and IF-logic have been studied in [18]. It is shown that while the two-variable IF-logic is undecidable, the corresponding fragment of dependence logic is *NEXPTIME*-complete.

⋆ Juha Kontinen was supported by grant 292767 of the Academy of Finland. Antti Kuusisto was supported by the ERC grant 647289 "CODA" and the Jenny and Antti Wihuri Foundation. Jonni Virtema was supported by grant 266260 of the Academy of Finland and a grant by the Finnish Academy of Science and Letters.

In this article we establish that the satisfiablity problems of the two-variable fragments of independence and inclusion logics are likewise *NEXPTIME*-complete. This result is established via proving a more general theorem that implies also a range of other decidability results for a variety of team-semantics-based logics with generalized dependency notions. Furthermore, we prove that the *validity* problem of two-variable dependence logic is undecidable; this result is the main result of the paper. The problem has been open for some time in the team semantics literature and has been explicitly posed in, e.g., [5], [18], [25], and elsewhere.

In addition to studying two-variable logics, we study the Bernays-Schönfinkel-Ramsey prefix class, i.e., sentences with the quantifier prefix $\exists^*\forall^*$. We show that—as in the case of ordinary first-order logic—the prefix class $\exists^*\forall^*$ of $\mathsf{FO}(\mathcal{A})$ is decidable for any uniformly polynomial time computable class $\mathcal{A}$ of generalized dependencies closed under substructures. We prove inclusion in 2*NEXPTIME*, and furthermore, for vocabularies of fixed arity, we show *NEXPTIME*-completeness. We also prove a partial converse of the result concerning logics $\mathsf{FO}(\mathcal{A})$ with a decidable prefix class $\exists^*\forall^*$, see Theorem 22.

2 Preliminaries

The domain of a structure $\mathfrak{A}$ is denoted by A. We assume that the reader is familiar with first-order logic FO. The extension of FO with counting quantifiers $\exists^{\geq i}$ is denoted by FOC. The two-variable fragments FO^2 and FOC^2 are the fragments of FO and FOC with formulas in which only the variables x and y appear. We let Σ_1^1 denote the fragment of formulas of second-order logic of the form $\exists X_1 ... \exists X_k\, \varphi$, where $X_1, ..., X_k$ are relation symbols and φ a first-order formula. $\Sigma_1^1(\mathsf{FOC}^2)$ is the extension of FOC^2 consisting of formulas of the form $\exists X_1 ... \exists X_k\, \chi$, where $X_1, ..., X_k$ are relation symbols and χ a formula of FOC^2.

2.1 Logics based on team semantics

Let $\mathbb{Z}_+$ denote the set of positive integers, and let $\mathrm{VAR} = \{\, v_i \mid i \in \mathbb{Z}_+ \,\}$ be the set of exactly all first-order *variable symbols*. We mainly use metavariables x, y, z, x_1, x_2, etc., in order to refer to variable symbols in VAR. We let $\overline{x}, \overline{y}, \overline{z}, \overline{x}_1, \overline{x}_2$, etc., denote finite nonempty tuples of variable symbols, i.e., tuples in VAR^n for some $n \in \mathbb{Z}_+$. When we study two-variable logics, we use the metavariables x and y, and assume they denote distinct variables in VAR.

Let $D \subseteq \mathrm{VAR}$ be a *finite*, possibly empty set. Let $\mathfrak{A}$ be a model. We do not allow for models to have an empty domain, so $A \neq \emptyset$. A function $s : D \to A$ is called an *assignment* with codomain A. If $\overline{x} = (x_1, \ldots, x_n)$, we denote $(s(x_1), \ldots, s(x_n))$ by $s(\overline{x})$. We let $s[a/x]$ denote the variable assignment with the domain $D \cup \{\, x \,\}$ and codomain A defined such that $s[a/x](y) = a$ if $y = x$, and $s[a/x](y) = s(y)$ if $y \neq x$. Let $T \in \mathcal{P}(A)$, where $\mathcal{P}$ denotes the power set operator. We define $s[\,T/x\,] = \{\, s[a/x] \mid a \in T \,\}$.

Let $D \subseteq \mathrm{VAR}$ be a finite, possibly empty set of first-order variable symbols. Let X be a set of assignments $s : D \to A$. Such a set X is a *team* with the *domain* D and *codomain* A. Note that the empty set is a team, as is the set $\{\emptyset\}$ containing only the empty assignment. The team $\emptyset$ does not have a unique domain; any finite subset of VAR is a domain of $\emptyset$. The domain of the team $\{\emptyset\}$ is $\emptyset$.

Let X be a team with the domain D and codomain A. Let $T \subseteq A$. We define $X[T/x] = \{\, s[a/x] \mid a \in T,\ s \in X \,\}$. Let $F : X \to \mathcal{P}(A)$ be a function. We define $X[\,F/x\,] = \bigcup\limits_{s \in X} s[\,F(s)/x\,]$. Let $C \subseteq A$. We define $X \upharpoonright C = \{\, s \in X \mid s(x) \in C \text{ for all } x \in D \,\}$.

Let X be a team with domain D. Let $k \in \mathbb{Z}_+$, and let $y_1, ..., y_k$ be variable symbols. Assume that $\{y_1, ..., y_k\} \subseteq D$. We define $\mathrm{rel}\big(X, (y_1, ..., y_k)\big) = \{\, \big(s(y_1), ..., s(y_k)\big) \mid s \in X \,\}$.

Let τ be a relational vocabulary, i.e., a vocabulary containing relation symbols only. (In this article we consider only relational vocabularies.) The syntax of a logic based on team semantics is usually given in negation normal form. We shall also follow this convention in the current article. For this reason, we define the syntax of first-order logic as follows.

$$\varphi \ ::= \ R(\overline{x}) \mid \neg R(\overline{x}) \mid x_1 = x_2 \mid \neg x_1 = x_2 \mid (\varphi_1 \vee \varphi_2) \mid (\varphi_1 \wedge \varphi_2) \mid \ \exists x\varphi \mid \forall x\varphi \ ,$$

where $R \in \tau$. The first four formula formation rules above introduce *first-order literals* to the language. Below we shall consider logics $\mathsf{FO}(\mathcal{A})$, where the above syntax is extended by clauses of the type $A_Q\,(\overline{y}_1, ..., \overline{y}_k)$. Here A_Q is (a symbol corresponding to) a *generalized atom* in $\mathcal{A}$ and each $\overline{y}_i$ is a tuple of variables. Before considering such novel atoms, let us define *lax team semantics* for first-order logic.

Definition 1 ([15,24]). *Let $\mathfrak{A}$ be a model and X a team with codomain A. The satisfaction relation $\mathfrak{A} \models_X \varphi$ is defined as follows.*

1. *If φ is a first-order literal, then $\mathfrak{A} \models_X \varphi$ iff for all $s \in X$: $\mathfrak{A}, s \models_{\mathsf{FO}} \varphi$. Here $\models_{\mathsf{FO}}$ refers to the ordinary Tarskian satisfaction relation of first-order logic.*
2. *$\mathfrak{A} \models_X \psi \wedge \varphi$ iff $\mathfrak{A} \models_X \psi$ and $\mathfrak{A} \models_X \varphi$.*
3. *$\mathfrak{A} \models_X \psi \vee \varphi$ iff there exist teams Y and Z such that $X = Y \cup Z$, $\mathfrak{A} \models_Y \psi$, and $\mathfrak{A} \models_Z \varphi$.*
4. *$\mathfrak{A} \models_X \exists x\, \psi$ iff $\mathfrak{A} \models_{X[F/x]} \psi$ for some $F\colon X \to (\mathcal{P}(A) \setminus \{\emptyset\})$.*
5. *$\mathfrak{A} \models_X \forall x\, \psi$ iff $\mathfrak{A} \models_{X[A/x]} \psi$.*

Finally, a sentence φ is true in a model $\mathfrak{A}$ ($\mathfrak{A} \models \varphi$) if $\mathfrak{A} \models_{\{\emptyset\}} \varphi$.

Proposition 2 ([15,24]). *Let ψ be a formula of first-order logic. We have $\mathfrak{A} \models_X \psi$ iff $\mathfrak{A}, s \models_{\mathsf{FO}} \psi$ for all $s \in X$.*

In this paper we consider first-order logic extended with generalized dependency atoms. Before formally introducing the notion of a generalized dependency atom, we recall some particular atoms familiar from the literature related to team semantics.

Dependence atoms $=\!(x_1, \dots, x_n, y)$, inspired by the slashed quantifiers of Hintikka and Sandu [14], were introduced by Väänänen [24]. The intuitive meaning of the atom $=\!(x_1, \dots, x_n, y)$ is that the value of the variable y depends solely on the values of the variables $x_1, \dots, x_n$. The semantics for dependence atoms is defined as follows:

$$\mathfrak{A} \models_X =\!(x_1, \dots, x_n, y) \text{ iff } \forall s, s' \in X: \text{ if } s((x_1, \dots, x_n)) = s'((x_1, \dots, x_n)) \text{ then } s(y) = s'(y).$$

Dependence logic (D) is the extension of first-order logic with dependence atoms.

While dependence atoms of dependence logic declare dependences between variables, *independence atoms*, introduced by Grädel and Väänänen [11], do just the opposite; independence atoms are used to declare independencies between variables. Independence atom is an atomic formula of the form $(x_1, ..., x_k) \perp_{(z_1,...,z_t)} (y_1, ..., y_l)$ with the intuitive meaning that for any fixed interpretation of the variables $z_1, \dots, z_t$, the interpretations of the variables $x_1, ..., x_k$ are independent of the interpretations of the variables $y_1, ..., y_l$. The semantics for independence atoms is defined as follows:

$$\mathfrak{A} \models_X (x_1, ..., x_k) \perp_{(z_1,...,z_t)} (y_1, ..., y_l) \text{ iff } \forall s, s' \in X\, \exists s'' \in X : \bigwedge_{i \le t} s(z_i) = s'(z_i)$$
$$\text{implies that } \bigwedge_{i \le k} s''(x_i) = s(x_i) \wedge \bigwedge_{i \le t} s''(z_i) = s(z_i) \wedge \bigwedge_{i \le l} s''(y_i) = s'(y_i).$$

Independence logic (Ind) is the extension of first-order logic with independence atoms.

Galliani [6] introduced *inclusion* and *exclusion atoms*. The intuitive meaning of the inclusion atom $(x_1, \dots, x_n) \subseteq (y_1, \dots, y_n)$ is that tuples interpreting the variables $x_1 \dots, x_n$ are also tuples interpreting $y_1, \dots, y_n$. The intuitive meaning of the exclusion atom $(x_1, \dots, x_n) \mid (y_1, \dots, y_n)$ on the other hand is that tuples interpreting the variables $x_1 \dots, x_n$ and the tuples interpreting $y_1, \dots, y_n$ are distinct. The semantics for inclusion atoms and exclusion atoms is defined as follows:

$$\mathfrak{A} \models_X (x_1, \dots, x_n) \subseteq (y_1, \dots, y_n) \text{ iff } \forall s \in X\, \exists s' \in X : s((x_1, \dots, x_n)) = s'((y_1, \dots, y_n)),$$
$$\mathfrak{A} \models_X (x_1, \dots, x_n) \mid (y_1, \dots, y_n) \text{ iff } \forall s, s' \in X : s((x_1, \dots, x_n)) \neq s'((y_1, \dots, y_n)).$$

The extension of first-order logic with inclusion atoms (exclusion atoms) is called *inclusion logic* (*exclusion logic*) and denoted by Inc (Exc). The extension of first-order logic with both inclusion atoms and exclusion atoms is called *inclusion/exclusion logic* and denoted by $\mathsf{Inc}/\mathsf{Exc}$.

2.2 Generalized atoms

In this section we first give the well known definition of generalized quantifiers (Lindström quantifiers [20]). We then show how each generalized quantifier naturally gives rise to a generalized atom. Finally, we discuss on some fundamental properties of first-order logic extended with generalized atoms. Generalized atoms were first defined in [19].

Let $(i_1, ..., i_n)$ be a nonempty sequence of positive integers. A generalized quantifier of the type $(i_1, ..., i_n)$ is a class $\mathcal{C}$ of structures $(A, B_1, ..., B_n)$ such that the following conditions hold.

1. $A \neq \emptyset$, and for each $j \in \{1, ..., n\}$, we have $B_j \subseteq A^{i_j}$.
2. If $(A', B'_1, ..., B'_n) \in \mathcal{C}$ and if there is an isomorphism $f : A' \to A''$ from $(A', B'_1, ..., B'_n)$ to another structure $(A'', B''_1, ..., B''_n)$, then $(A'', B''_1, ..., B''_n) \in \mathcal{C}$.

Let Q be a generalized quantifier of the type $(i_1, ..., i_n)$. Let $\mathfrak{A}$ be a model with the domain A. We define $Q^{\mathfrak{A}}$ to be the set $\{ (B_1, ..., B_n) \mid (A, B_1, ..., B_n) \in Q \}$.

Let n be a positive integer. Let Q be a generalized quantifier of the type $(i_1, ..., i_n)$. Extend the syntax of first-order logic with atomic expressions of the type $A_Q(\overline{y}_1, ..., \overline{y}_n)$, where each $\overline{y}_j$ is a tuple of variables of length i_j. Let X be a team whose domain contains all variables occurring in the tuples $\overline{y}_1, ..., \overline{y}_n$. Extend team semantics such that $\mathfrak{A} \models_X A_Q(\overline{y}_1, ..., \overline{y}_n)$ if and only if $\big(\mathrm{rel}(X, \overline{y}_1), ..., \mathrm{rel}(X, \overline{y}_n)\big) \in Q^{\mathfrak{A}}$. The generalized quantifier Q defines a *generalized atom* A_Q of the type $(i_1, ..., i_n)$.

A generalized atom A_Q is *downwards closed* if for all $\mathfrak{A}$, X and $\overline{y}_1, ..., \overline{y}_k$, it holds that if $\mathfrak{A} \models_X A_Q(\overline{y}_1, ..., \overline{y}_k)$ and $Y \subseteq X$, then $\mathfrak{A} \models_Y A_Q(\overline{y}_1, ..., \overline{y}_k)$. Similarly, a generalized atom A_Q is *closed under substructures* if for all $\mathfrak{A}$, X and $\overline{y}_1, ..., \overline{y}_k$, it holds that if $\mathfrak{A} \models_X A_Q(\overline{y}_1, ..., \overline{y}_k)$, $\mathfrak{A}' := \mathfrak{A} \upharpoonright B$ and $X' := X \upharpoonright B$ for some $B \subseteq A$, then we have $\mathfrak{A}' \models_{X'} A_Q(\overline{y}_1, ..., \overline{y}_k)$. Finally, a generalized atom A_Q is *universe independent* if for all $\mathfrak{A}$, $\mathfrak{B}$, X and $\overline{y}_1, ..., \overline{y}_k$, where both A and B are codomains for X, it holds that $\mathfrak{A} \models_X A_Q(\overline{y}_1, ..., \overline{y}_k)$ if and only if $\mathfrak{B} \models_X A_Q(\overline{y}_1, ..., \overline{y}_k)$.

Let φ be a formula of first-order logic, possibly extended with generalized atoms. The set $\mathrm{Fr}(\varphi)$ of *free variables* of φ is defined in the same way as in first-order logic. The set $\mathrm{Fr}(A_Q(\overline{y}_1, ..., \overline{y}_k))$ of course contains exactly all variable that occur in the tuples $\overline{y}_i$. The satisfiability problem of a (possibly team-semantics-based) logic L takes as an input a sentence of L and asks whether $\mathfrak{A} \models \varphi$ for some model $\mathfrak{A}$. The validity problem asks, given a sentence φ, whether $\mathfrak{A} \models \varphi$ for all models $\mathfrak{A}$.

Let $k \in \mathbb{Z}_+$ and let A_Q be a generalized atom of the type $(i_1, ..., i_n)$, where $i_j \leq k$ for each j. Let $\varphi(R_1, ..., R_n)$ be a sentence of $\Sigma^1_1(\mathsf{FOC}^k)$ with unquantified relation symbols $R_1, ..., R_n$ of arities $i_1, ..., i_n$, respectively. Assume that for all models $\mathfrak{A}$ and teams X with codomain A and domain containing the variables in $A_Q(\overline{x}_1, ..., \overline{x}_n)$, we have $\mathfrak{A} \models_X A_Q(\overline{x}_1, ..., \overline{x}_n)$ iff

$$\big(\mathfrak{A}, R_1 := \mathrm{rel}(X, \overline{x}_1), ..., R_n := \mathrm{rel}(X, \overline{x}_n)\big) \models_{\mathsf{FO}} \varphi(R_1, ..., R_n).$$

Then we say that the atom A_Q is definable in $\Sigma^1_1(\mathsf{FOC}^k)$.

We now show that, *for any generalized atom* A_Q, the logic $\mathsf{FO}(A_Q)$ has the so-called locality property. We also show that, for a downwards closed atom A_Q, all formulas of $\mathsf{FO}(A_Q)$ satisfy the downwards closure property. These two properties have previously turned out to be very useful in the study of dependence logic.

Let X be a team with domain $\{x_1, \ldots, x_k\}$, and let $V \subseteq \{x_1, \ldots, x_k\}$. We denote by $X(V)$ the team $\{s \upharpoonright V \mid s \in X\}$ with the domain V. The following proposition shows that the truth of an $\mathsf{FO}(A_Q)$-formula depends only on the interpretations of the variables occurring free in the formula. The proof uses the fact that generalized atoms satisfy the claim by definition. Otherwise the proof is identical to the corresponding proof given in [6].

Proposition 3 (Locality). *Let A_Q be a generalized atom and $\varphi \in \mathsf{FO}(A_Q)$ a formula. If $V \supseteq \mathrm{Fr}(\varphi)$, then $\mathfrak{A} \models_X \varphi$ if and only if $\mathfrak{A} \models_{X(V)} \varphi$.*

The next proposition is also very useful. The proof is almost identical to the corresponding proof for dependence logic, see [24]. The additional case for generalized atoms follows by the assumption of downwards closure.

Proposition 4 (Downward closure). *Let A_Q be a downwards closed generalized atom. Suppose φ is an $\mathsf{FO}(A_Q)$-formula, $\mathfrak{A}$ a model, and $Y \subseteq X$ teams. Then $\mathfrak{A} \models_X \varphi$ implies $\mathfrak{A} \models_Y \varphi$.*

3 Satisfiability problems of logics $\mathsf{FO}^2(\mathcal{A})$

In this section we show that for any finite collection $\mathcal{A}$ of $\Sigma_1^1(\mathsf{FOC}^2)$-definable atoms A_Q, both $\mathrm{SAT}(\mathsf{FO}^2(\mathcal{A}))$ and $\mathrm{FINSAT}(\mathsf{FO}^2(\mathcal{A}))$ are *NEXPTIME*-complete. Our proof relies on a translation from $\mathsf{FO}^2(\mathcal{A})$ into $\Sigma_1^1(\mathsf{FOC}^2)$ and the fact that $\mathrm{SAT}(\mathsf{FOC}^2)$ and $\mathrm{FINSAT}(\mathsf{FOC}^2)$ are *NEXPTIME*-complete [22].

We start by establishing a more general translation. We show that for every $k \geq 1$ and every $\Sigma_1^1(\mathsf{FOC}^k)$ definable atom A_Q, we have $\mathsf{FO}^k(A_Q) \leq \Sigma_1^1(\mathsf{FOC}^k)$. Note that strictly speaking $\mathsf{FO}^k(A_Q)$ uses only one atom A_Q instead of a finite collection $\mathcal{A}$ of atoms, but our proof below generalizes *directly* to the case with a finite collection of atoms. The reason for considering a single atom is simply to keep the notation light.

When considering k-variable logic, we let $\{x_1, ..., x_k\}$ denote the k distinct variables used in the syntax of the logic, and we let $rel(X)$ denote $rel\big(X, (x_1, ..., x_k)\big)$. The following lemma is possibly the technically most involved part of our argument in this section for establishing decidability of two-variable inclusion and independence logics. The proof significantly modifies and extends the argument establishing Lemma 3.3.14 of [25]. See also [18] and Theorem 6.2 in [24].

Lemma 5. *Assume that $k, t \geq 1$. Let τ be a relational vocabulary, let $R \notin \tau$ be a k-ary relation symbol and let A_Q be a $\Sigma_1^1(\mathsf{FOC}^k)$-definable atom of type $(i_1, \ldots, i_t)$, where $i_j \leq k$ for each j. For every formula $\varphi \in \mathsf{FO}^k(A_Q)$ there exists a sentence $\varphi^* \in \Sigma_1^1(\mathsf{FOC}^k)(\tau \cup \{R\})$ such that for every model $\mathfrak{A}$ and team X with codomain A and $\mathrm{dom}(X) = \{x_1, \ldots, x_k\}$, we have*

$$\mathfrak{A} \models_X \varphi \quad \textit{iff} \quad \big(\mathfrak{A}, \mathrm{rel}(X)\big) \models \varphi^*, \tag{1}$$

where $(\mathfrak{A}, \mathrm{rel}(X))$ is the expansion $\mathfrak{A}'$ of $\mathfrak{A}$ into the vocabulary $\tau \cup \{R\}$ such that $R^{\mathfrak{A}'} := \mathrm{rel}(X)$. Moreover φ^* is computable from φ in polynomial time.

Proof. Fix $k \geq 1$ and the $\Sigma_1^1(\mathsf{FOC}^k)$-definable atom A_Q. Let $(i_1, \ldots, i_t)$, where $i_j \leq k$ for each j, be the type of A_Q. Let $\varphi_{A_Q}(R_1, \ldots, R_t)$ be the $\Sigma_1^1(\mathsf{FOC}^k)$-sentence that defines A_Q. We will define a translation

$$tr_k : \mathsf{FO}^k(A_Q)(\tau) \;\to\; \Sigma_1^1(\mathsf{FOC}^k)(\tau \cup \{R\})$$

inductively. Below we always assume that the quantified relations S and T are fresh, i.e., they are assumed not to appear in $tr_k(\psi)$ or $tr_k(\vartheta)$. Notice that for every $\mathsf{FO}^k(A_Q)$-formula φ, we have $tr_k(\varphi) = \exists S_1 \ldots \exists S_n \varphi'$ for some k-ary relation variables $S_1 \ldots S_n$ ($n \in \mathbb{N}$) and some FOC^k-formula φ'. The translation tr_k is defined as follows.

1. If φ is a first-order literal (and thus not a generalized atom), then

$$tr_k(\varphi) := \forall x_1 \ldots \forall x_k \big(R(x_1, \ldots, x_k) \to \varphi\big).$$

2. Assume that φ is a generalized atom $A_Q(\overline{y}_1, \ldots, \overline{y}_t)$, where $\overline{y}_j \in \{x_1, \ldots, x_k\}^{i_j}$ for each $j \leq t$. Let $\overline{Y}$ and $\psi \in \mathsf{FOC}^k(R_1, ..., R_t)$ be such that $\varphi_{A_Q} = \overline{\exists Y}\psi$. For technical reasons, we will simulate i_j-ary relations by k-ary relations. Define that, for each $j \leq t$,

$$\mathrm{Id}_j := \{(l, m) \in \mathbb{N}^2 \mid y_{j_l} \text{ and } y_{j_m} \text{ denote the same variable symbol}\},$$

where y_{j_l} (y_{j_m}) denotes the l-th (m-th) element of $\overline{y}_j$. Now $tr_k(\varphi)$ is defined to be the formula

$$\overline{\exists Y}\, \exists T_1 \ldots \exists T_t \Big(\bigwedge_{j \leq t} \Big(\varphi_{j\text{-}padding} \,\wedge\, \varphi_{j\text{-}identities}\Big) \,\wedge\, \psi' \Big),$$

where the relation variables T_j and formulas $\varphi_{j\text{-}padding}$, $\varphi_{j\text{-}identities}$ and ψ' are defined as follows. Each variable T_j is a fresh k*-ary* relation variable. The formula ψ' is the conjunction

$\psi'' \wedge \bigwedge_{j\leq t} \chi_j$, where ψ'' and χ_j are as follows. The conjunct ψ'' is obtained from ψ by replacing each atomic formula $R_j(z_1,\ldots,z_{i_j})$ by $T_j(z_1,\ldots,z_{i_j},z_1,\ldots,z_1)$. For each $j \leq t$, χ_j is the formula

$$\forall x_1 \ldots \forall x_k \Big(\exists x_{i_{j+1}}\, T_j(x_1,...,x_{i_j},x_{i_{j+1}},...,x_{i_{j+1}}) \\ \rightarrow \forall x_{i_{j+1}}\, T_j(x_1,...,x_{i_j},x_{i_{j+1}},...,x_{i_{j+1}})\Big),$$

where in the case $i_j = k$ the formulas $\exists x_{i_{j+1}}\, T_j(x_1,...,x_{i_j},x_{i_{j+1}},...,x_{i_{j+1}})$ and

$$\forall x_{i_{j+1}}\, T_j(x_1,...,x_{i_j},x_{i_{j+1}},...,x_{i_{j+1}})$$

are replaced by $T_j(x_1,...,x_k)$. The formula $\varphi_{j\text{-}identities}$ is

$$\forall x_1 \ldots \forall x_k \Big(T_j(x_1,\ldots,x_k) \rightarrow (\bigwedge_{(l,m)\in \mathrm{Id}_j} (x_l = x_m) \wedge \bigwedge_{l,m>i_j} x_l = x_m)\Big).$$

The formula $\varphi_{j\text{-}padding}$ is the formula

$$\forall x_1 \ldots \forall x_k \Big(\big(R(x_1,\ldots,x_k) \rightarrow \forall x_{m_j} T_j(\overline{y}_j, x_{m_j},\ldots,x_{m_j})\big) \\ \wedge \big(\exists x_{m_j} T_j(\overline{y}_j, x_{m_j},\ldots,x_{m_j}) \rightarrow \exists \overline{z}_j R(x_1,\ldots,x_k)\big)\Big),$$

where $\overline{z}_j$ is the tuple of variables in $(x_1,\ldots,x_k)$ but not in $\overline{y}_j$, and $m_j \leq k$ is the smallest integer such that the variable x_{m_j} does not occur in the tuple $\overline{y}_j$; in the case that such variable does not exist the formulas $\forall x_{m_j} T_j(\overline{y}_j, x_{m_j},\ldots,x_{m_j})$ and $\exists x_{m_j} T_j(\overline{y}_j, x_{m_j},\ldots,x_{m_j})$ are replaced by $T_j(\overline{y}_j)$.

3. Assume that $tr_k(\psi) = \exists S_1 \ldots \exists S_n \psi'$ and $tr_k(\vartheta) = \exists T_1 \ldots \exists T_m \vartheta'$, where ψ' and ϑ' are FOC^k-formulas. Furthermore, assume that the relation variables $S_1,\ldots S_n, T_1,\ldots,T_m$ are all distinct.
   (a) If φ is of the form $(\psi \vee \vartheta)$, then $tr_k(\varphi)$ is defined to be the formula

$$\exists S \exists T \exists S_1 \ldots \exists S_n \exists T_1 \ldots \exists T_m \Big(\forall x_1 \ldots \forall x_k \Big(R(x_1,\ldots,x_k) \\ \leftrightarrow \big(S(x_1,\ldots,x_k) \vee T(x_1,\ldots,x_k)\big)\Big) \wedge \psi'(S/R) \wedge \vartheta'(T/R)\Big),$$

   where $\psi'(S/R)$ denotes the formula obtained from ψ' by replacing occurrences of R by S, and analogously for $\vartheta'(T/R)$.
   (b) If $\varphi = (\psi \wedge \vartheta)$, then $tr_k(\varphi)$ is the formula $\exists S_1 \ldots \exists S_n \exists T_1 \ldots \exists T_m \big(\psi' \wedge \vartheta'\big)$.
4. If φ is of the form $\exists x_i \psi$ and $tr_k(\psi) = \exists S_1 \ldots \exists S_n \psi'$, where ψ' is an FOC^k-formula, then $tr_k(\varphi)$ is the formula

$$\exists S \exists S_1 \ldots \exists S_n \Big(\forall x_1 \ldots \forall x_k \big(\exists x_i R(x_1,\ldots,x_k) \leftrightarrow \exists x_i S(x_1,\ldots,x_k)\big) \wedge \psi'(S/R)\Big).$$

5. If φ is of the form $\forall x_i \psi$ and $tr_k(\psi) = \exists S_1 \ldots \exists S_n \psi'$, where ψ' is an FOC^k-formula, then $tr_k(\varphi)$ is defined to be the formula

$$\exists S \exists S_1 \ldots \exists S_n \Big(\forall x_1 \ldots \forall x_k \Big(\big(R(x_1,\ldots,x_k) \rightarrow \forall x_i S(x_1,\ldots,x_k)\big) \\ \wedge \big(S(x_1,\ldots,x_k) \rightarrow \exists x_i R(x_1,\ldots,x_k)\big)\Big) \wedge \psi'(S/R)\Big).$$

A straightforward induction on φ shows that for every model $\mathfrak{A}$ and every team with codomain A such that $\mathrm{dom}(X) = \{x_1,\ldots,x_k\}$, $\mathfrak{A} \models_X \varphi$ iff $\big(\mathfrak{A}, \mathrm{rel}(X)\big) \models tr_k(\varphi)$.

Theorem 6. *For every $k \geq 1$ and for every $\Sigma^1_1(\mathsf{FOC}^k)$-definable atom A_Q it holds that $\mathsf{FO}^k(A_Q) \leq \Sigma^1_1(\mathsf{FOC}^k)$, i.e., for every sentence of $\mathsf{FO}^k(A_Q)$, there exists an equivalent sentence of $\Sigma^1_1(\mathsf{FOC}^k)$.*

Proof. Let τ be a relational vocabulary, $k \geq 1$, and A_Q a $\Sigma_1^1(\mathsf{FOC}^k)$-definable atom. Let φ be an $\mathsf{FO}^k(A_Q)(\tau)$-sentence and $\varphi^* = \exists R_1 \ldots \exists R_n \psi$ the related $\Sigma_1^1(\mathsf{FOC}^k)(\tau \cup \{R\})$-sentence given by Lemma 5. The following conditions are equivalent.

1. $\mathfrak{A} \models \varphi$.
2. $\mathfrak{A} \models_X \varphi$ for some nonempty team X such that $\mathrm{dom}(X) = \{x_1, \ldots, x_k\}$.
3. $(\mathfrak{A}, \mathrm{rel}(X)) \models \varphi^*$ for some nonempty team X such that $\mathrm{dom}(X) = \{x_1, \ldots, x_k\}$.
4. $(\mathfrak{A}, R) \models \exists R_1 \ldots \exists R_n (\exists x_1 \ldots \exists x_k R(x_1, \ldots, x_k) \wedge \psi)$ for some $R \subseteq A^k$.
5. $\mathfrak{A} \models \exists R \exists R_1 \ldots \exists R_n (\exists x_1 \ldots \exists x_k R(x_1, \ldots, x_k) \wedge \psi)$.

The equivalence of 1 and 2 follows from Proposition 3 and the fact that $\mathrm{Fr}(\varphi) = \emptyset$. By Lemma 5, conditions 2 and 3 are equivalent. The equivalence of 3 and 4 follows from the fact that $\varphi^* = \exists R_1 \ldots \exists R_n \psi$. The conditions 4 are 5 clearly equivalent.

Theorem 7. *Let A_Q be a $\Sigma_1^1(\mathsf{FOC}^2)$-definable generalized atom. Then the problems* $\textsc{Sat}(\mathsf{FO}^2(A_Q))$ *and* $\textsc{FinSat}(\mathsf{FO}^2(A_Q))$ *are NEXPTIME-complete.*

Proof. Since the translation $\varphi \mapsto \varphi^*$ is computable in polynomial time and (finite) satisfiability of $\Sigma_1^1(\mathsf{FOC}^2)$ can be checked in *NEXPTIME* [22], we conclude that both $\textsc{Sat}(\mathsf{FO}^2(A_Q))$ and $\textsc{FinSat}(\mathsf{FO}^2(A_Q))$ are in *NEXPTIME*. On the other hand, since $\mathsf{FO}^2 \leq \mathsf{FO}^2(A_Q)$ by Proposition 2, and since both $\textsc{Sat}(\mathsf{FO}^2)$ and $\textsc{FinSat}(\mathsf{FO}^2)$ are *NEXPTIME*-hard [9], it follows that both $\textsc{Sat}(\mathsf{FO}^2(A_Q))$ and also $\textsc{FinSat}(\mathsf{FO}^2(A_Q))$ are as well.

The result of Theorem 7 can be directly generalized to concern finite collections $\mathcal{A}$ of generalized atoms. The proof of the following theorem is practically the same as that of Theorem 7.

Theorem 8. *Let $\mathcal{A}$ be a finite collection of $\Sigma_1^1(\mathsf{FOC}^2)$-definable generalized atoms. The satisfiability and the finite satisfiability problems of $\mathsf{FO}^2(\mathcal{A})$ are NEXPTIME-complete.*

We shall next make use of Theorem 8 in order to show that the satisfiability and the finite satisfiability problems of two-variable fragments of dependence logic, inclusion logic, exclusion logic and independence logic are *NEXPTIME*-complete. The result for two-variable dependence logic was already established in [18]. Note that when regarded as generalized atoms, each of the dependency notions above correspond to a collection of generalized atoms; for example the atomic formulas $=\!(x, y)$ and $=\!(x, y, z)$ refer to two different atoms, one of type (2) and the other of type (3). However, in order to capture the two-variable fragments of of these logics, we only need a finite number of generalized atoms for each logic, as we shall see. We define $\varphi_{const} := \exists^{\leq 1} x R(x)$, $\varphi_{dep} := \forall x \exists^{\leq 1} y R(x, y)$, $\varphi_{inc} := \forall x \forall y (R(x, y) \rightarrow S(x, y))$, $\varphi_{exc} := \forall x \forall y (R(x, y) \rightarrow \neg S(x, y))$, $\varphi_{ind} := \forall x \forall y ((\exists y R(x, y) \wedge \exists x R(x, y)) \rightarrow R(x, y))$.

The formulas φ_{const}, φ_{dep}, φ_{inc}, φ_{exc} and φ_{ind} define the generalized atoms A_{const} of type (1), A_{dep} of type (2), A_{inc} of type $(2, 2)$, A_{exc} of type $(2, 2)$, and A_{ind} of type (2), respectively.

Theorem 9. *The satisfiability and finite satisfiability problems of the two-variable fragments of dependence logic, inclusion logic, exclusion logic, inclusion/exclusion logic, and independence logic are all NEXPTIME-complete.*

Proof. We establish polynomial time translations $\mathsf{D}^2 \rightarrow \mathsf{FO}^2(\{A_{const}, A_{dep}\})$, $\mathsf{Inc}^2 \rightarrow \mathsf{FO}^2(A_{inc})$, $\mathsf{Exc}^2 \rightarrow \mathsf{FO}^2(A_{exc})$, $\mathsf{Inc/Exc}^2 \rightarrow \mathsf{FO}^2(A_{inc}, A_{exc})$, and $\mathsf{Ind}^2 \rightarrow \mathsf{FO}^2(\{A_{const}, A_{dep}, A_{ind}\})$ that preserve equivalence. The result then follows from Theorem 8 and the fact that the generalised atoms A_{const}, A_{dep}, A_{exc}, A_{inc}, A_{ind} are all $\Sigma_1^1(\mathsf{FOC}^2)$-definable.

Notice first that in dependence atoms, repetition of variables can always be avoided. The atom $=\!(\overline{x}, y)$ is equivalent to the atom $=\!(\overline{x}\,', y)$, where $\overline{x}\,'$ is obtained from $\overline{x}$ by simply removing the repetition of variables. Furthermore, if y occurs in the tuple $\overline{x}$, then $=\!(\overline{x}, y)$ is equivalent to $y = y$. Thus we may assume that in formulas of two-variable dependence logic, only dependence atoms $=\!(x)$, $=\!(y)$, $=\!(x, y)$, and $=\!(y, x)$ may occur. Clearly $=\!(x)$ is equivalent to the generalized atom $A_{const}(x)$, while $=\!(x, y)$ is equivalent to the generalized atom $A_{dep}(x, y)$. Since

A_{const} and A_{dep} are $\Sigma^1_1(\mathsf{FOC}^2)$-definable atoms, by Theorem 8, $\textsc{Sat}(\mathsf{FO}^2(\{A_{const}, A_{dep}\}))$ and $\textsc{FinSat}(\mathsf{FO}^2(\{A_{const}, A_{dep}\}))$ are *NEXPTIME*-complete. Thus both $\textsc{Sat}(\mathsf{D}^2)$ and $\textsc{FinSat}(\mathsf{D}^2)$ are as well.

It is straightforward to show that in two-variable inclusion logic, only inclusion atoms of type $(y_1, y_2) \subseteq (z_1, z_2)$, where $y_1, y_2, z_1, z_2 \in \{x, y\}$, are needed. For example, the inclusion atom $x \subseteq y$ can be replaced by the equivalent inclusion atom $(x, x) \subseteq (y, y)$, and the inclusion atoms $(x, y, x) \subseteq (x, y, y)$ and $(x, y, y) \subseteq (y, x, x)$ can be replaced by the equivalent atomic formulas $x = y$ and $(x, y) \subseteq (y, x)$, respectively. Thus we may assume that in formulas of two-variable inclusion logic, only inclusion atoms of type $(y_1, y_2) \subseteq (z_1, z_2)$ may occur; inclusion atoms of other kinds can easily be eliminated in polynomial time. Clearly $(y_1, y_2) \subseteq (z_1, z_2)$ is equivalent to the generalized atom $A_{inc}\big((y_1, y_2), (z_1, z_2)\big)$. Since A_{inc} is a $\Sigma^1_1(\mathsf{FOC}^2)$-definable atom, it follows from Theorem 7 that $\textsc{Sat}(\mathsf{FO}^2(A_{inc}))$ and $\textsc{FinSat}(\mathsf{FO}^2(A_{inc}))$ are *NEXPTIME*-complete. Thus $\textsc{Sat}(Inc^2)$ and $\textsc{FinSat}(Inc^2)$ are as well.

Using analogous argumentation, it is straightforward to show that in two-variable exclusion logic, only exclusion atoms of type $(y_1, y_2) \mid (z_1, z_2)$, where $y_1, y_2, z_1, z_2 \in \{x, y\}$, are needed. Clearly $(y_1, y_2) \mid (z_1, z_2)$ is equivalent to the generalized atom $A_{exc}\big((y_1, y_2), (z_1, z_2)\big)$. Since A_{exc} is a $\Sigma^1_1(\mathsf{FOC}^2)$-definable atom, it follows from Theorem 7 that both $\textsc{Sat}(\mathsf{FO}^2(A_{exc}))$ and $\textsc{FinSat}(\mathsf{FO}^2(A_{exc}))$ are *NEXPTIME*-complete. Thus $\textsc{Sat}(\mathsf{Exc}^2)$ and $\textsc{FinSat}(\mathsf{Exc}^2)$ are as well. Similarly it follows that $\textsc{Sat}(\mathsf{Inc/Exc}^2)$ and $\textsc{FinSat}(\mathsf{Inc/Exc}^2)$ are *NEXPTIME*-complete.

Likewise, it is easy to show that in the formulas of two-variable independence logic, only restricted versions of independence atoms are needed. First notice that we may always assume that in independence atoms $\overline{x}\perp_{\overline{y}}\overline{z}$, repetition of variables does not occur in any of the tuples $\overline{x}$, $\overline{y}$ and $\overline{z}$. By the semantics of independence atoms, it is also easy to check that the atoms $\overline{x}\perp_{\overline{y}}\overline{z}$ and $\overline{z}\perp_{\overline{y}}\overline{x}$ are always equivalent. Furthermore, it is clear that the order of variables in the tuples $\overline{x}$, $\overline{y}$, and $\overline{z}$ makes no difference. Notice then that each of the following atoms in the variables x, y is equivalent to the formula $\exists x\, x = x$:

$$\emptyset\perp_{\overline{x}}\overline{y}, \quad \overline{x}\perp_{(x,y)}\overline{y}, \quad x\perp_x x, \quad x\perp_x y, \quad x\perp_x (x, y), \quad y\perp_y y, \quad x\perp_y y, \quad y\perp_y (x, y).$$

Notice also the following equivalences:

$$(x, y)\perp_x (x, y) \equiv y\perp_x y, \quad y\perp_x (x, y) \equiv y\perp_x y, \quad (x, y)\perp_y (x, y) \equiv x\perp_y x,$$
$$x\perp_y (x, y) \equiv x\perp_y x, \quad x\perp (x, y) \equiv x\perp x, \quad y\perp (x, y) \equiv y\perp y.$$

Thus we may assume that only the independence atoms $x\perp x$, $y\perp y$, $x\perp y$, $(x, y)\perp(x, y)$, $x\perp_y x$, and $y\perp_x y$ occur in the formulas of two-variable independence logic. It is straightforward to check that the following equivalences between independence atoms and generalized atoms hold:

$$x\perp x \equiv A_{const}(x), \quad y\perp y \equiv A_{const}(y), \quad x\perp y \equiv A_{ind}\big((x, y)\big),$$
$$(x, y)\perp(x, y) \equiv A_{const}(x) \wedge A_{const}(y), \quad x\perp_y x \equiv A_{dep}(y, x), \quad y\perp_x y \equiv A_{dep}(x, y).$$

Since A_{const}, A_{dep}, and A_{ind} are all $\Sigma^1_1(\mathsf{FOC}^2)$-definable atoms, it follows from Theorem 8 that $\textsc{Sat}(\mathsf{FO}^2(\{A_{const}, A_{dep}, A_{ind}\}))$ and $\textsc{FinSat}(\mathsf{FO}^2(\{A_{const}, A_{dep}, A_{ind}\}))$ are *NEXPTIME*-complete. Thus $\textsc{Sat}(\mathsf{Ind}^2)$ and $\textsc{FinSat}(\mathsf{Ind}^2)$ are as well.

4 Undecidability via non-tiling

In this section we introduce structures and methods that we will later employ to prove undecidability of the validity problem of two-variable dependence logic. Curiously, all attempts (by us or known to us) to use the standard (Π^0_1-complete) tiling problem for the undecidability proof have failed; we will instead use the (Σ^0_1-complete) non-tiling problem in our arguments below.

The grid is the structure $\mathfrak{G} = (\mathbb{N}^2, V, H)$, where $V = \{\big((i, j), (i, j+1)\big) \in \mathbb{N}^2 \times \mathbb{N}^2 \mid i, j \in \mathbb{N}\}$ and $H = \{\big((i, j), (i+1, j)\big) \in \mathbb{N}^2 \times \mathbb{N}^2 \mid i, j \in \mathbb{N}\}$. A function $t : 4 \longrightarrow \mathbb{N}$ is called a *tile type*. Define the set $\mathrm{TILES} := \{P_t \mid t \text{ is a tile type}\}$ of unary relation symbols. The unary relation symbols in the set TILES are called *tiles*. The number $t(0)$ is the *top colour*, $t(1)$ the *right colour*, $t(2)$ the *bottom colour*, and $t(3)$ the *left colour* of P_t.

Let T be a finite nonempty set of tiles and V and H binary relation symbols. We say that a structure $\mathfrak{A} = (A, V, H)$ is T*-tilable*, if there exists an expansion of $\mathfrak{A}$ to the vocabulary $\{H, V\} \cup \{ P_t \mid P_t \in T \}$ such that the following conditions hold for all $u, v \in A$.

1. The point u belongs to the extension of exactly one symbol P_t in T.
2. If uHv, $P_t(u)$ and $P_s(v)$, then the right colour of P_t is the same as the left colour of P_s.
3. If uVv, $P_t(u)$ and $P_s(v)$, then the top colour of P_t is the same as the bottom colour of P_s.

We will next define the *tiling problem* and the *non-tiling problem*. Let $\mathcal{F}$ denote the set of finite, nonempty subsets of TILES. We define $\mathcal{T} := \{T \in \mathcal{F} \mid \mathfrak{G} \text{ is } T\text{-tilable}\}$ and $\bar{\mathcal{T}}' := \{T \in \mathcal{F} \mid \mathfrak{G} \text{ is not } T\text{-tilable}\}$. The *tiling problem (non-tiling problem,* resp.) is the membership problem of the set $\mathcal{T}$ ($\bar{\mathcal{T}}'$, resp.) with the input set $\mathcal{F}$.

Theorem 10 ([2]). *The tiling problem is Π^0_1-complete.*

The *non-tiling problem* is the complement of the tiling problem. Thus the following corollary follows.

Corollary 11. *The non-tiling problem is Σ^0_1-complete.*

The proof of the following lemma is straightforward.

Lemma 12. *There is a computable function associating each input T to the non-tiling problem with an FO^2-sentence φ_T of the vocabulary $\tau := \{H, V\} \cup T$ such that for every structure $\mathfrak{A}$ of the vocabulary $\{H, V\}$, the structure $\mathfrak{A}$ is not T-tilable iff for every expansion $\mathfrak{A}^*$ of $\mathfrak{A}$ to the vocabulary τ, it holds that $\mathfrak{A}^* \models \varphi_T$.*

Definition 13. *Let $\tau = \{V, H\}$ be a vocabulary where V and H are binary relation symbols. Let $\mathfrak{A} = (A, V, H)$ be a τ-structure. We say that $\mathfrak{A}$ is* gridlike *if the below conditions hold.*

1. *The extension of V in $\mathfrak{A}$ is serial (i.e., $\forall x \in A\ \exists y \in A$ s.t. $V(x, y)$).*
2. *The extension of H in $\mathfrak{A}$ is serial (i.e., $\forall x \in A\ \exists y \in A$ s.t. $H(x, y)$).*
3. *If $a, b, c, b', c' \in A$ are such that $V(a, b)$, $H(b, c)$, $H(a, b')$, and $V(b', c')$, then $c =$ c'.*

Note that it follows from the above definition that in gridlike structures, for every point a, there exist points b, c and d such that $H(a, b)$, $V(a, c)$, $V(b, d)$, and $H(c, d)$.

Let τ be the vocabulary of gridlike structures and U, P, Q, C unary relation symbols. We say that a $\tau \cup \{U, P, Q, C\}$-structure $\mathfrak{A}$ is *striped and gridlike* if the τ-reduct of $\mathfrak{A}$ is gridlike, the extensions of P and Q in $\mathfrak{A}$ are *distinct* singleton sets, the extension of U in $\mathfrak{A}$ is the union of the extensions of P and Q, and $\mathfrak{A}$ has the following property (intuitively C creates stripes in $\mathfrak{A}$):

$$\big(H(a, b) \Rightarrow (C(a) \Leftrightarrow C(b))\big) \text{ and } \big(V(a, b) \Rightarrow (C(a) \Leftrightarrow \neg C(b))\big). \tag{2}$$

The following lemma can be now proven by a simple inductive argument.

Lemma 14. *If $\mathfrak{A}$ is striped and gridlike, then there exists a homomorphism from the grid into $\mathfrak{A}$.*

Lemma 15. *Let T be an input to the non-tiling problem. The grid is non-T-tilable iff (the {H,V}-reduct of) every striped gridlike structure is non-T-tilable.*

Proof. The direction from left to right follows from Lemma 14 in a straightforward way. The converse holds since the grid is an {H,V}-reduct of a striped gridlike structure.

5 The validity problem of D^2 is undecidable

In this section we give a reduction from the non-tiling problem to the validity problem of D^2.

Let $\tau = \{V, H, C, U, P, Q\}$ be the vocabulary of striped gridlike structures. We will first define a formula $\varphi_{non-grid}$ of D^2 such that $\mathfrak{A}$ is not striped and gridlike iff $\mathfrak{A} \models \varphi_{non-grid}$. We first notice that the first two conditions of Definition 13 are easy to deal with. Define $\varphi_{non-serial} := \exists x \forall y \neg V(x,y) \vee \exists x \forall y \neg H(x,y)$. The third condition of Definition 13 is nontrivial. In the below construction, we will use the predicates P, Q, U for counting (only). We will first show how to force the extensions of P and Q to be distinct singletons and the extension of U to be the union of P and Q. The next formulae will be used for dealing with the cases where this *does not* hold.

$$\begin{aligned}
\varphi_{non-singleton}(X) &:= \forall x \neg X(x) \vee \exists x \exists y \big(X(x) \wedge X(y) \wedge \neg x = y \big) \\
\varphi_{non-distinct}(X,Y) &:= \exists x \big(X(x) \wedge Y(x) \big) \\
\varphi_{non-union}(X,Y,Z) &:= \exists x \Big(X(x) \wedge \big(\neg Y(x) \vee \neg Z(x) \big) \Big) \vee \exists x \Big(\neg X(x) \wedge \big(Y(x) \vee Z(x) \big) \Big) \\
\varphi_{|U| \neq 2} &:= \varphi_{non-singleton}(P) \vee \varphi_{non-singleton}(Q) \vee \varphi_{non-distinct}(P,Q) \\
&\quad \vee \varphi_{non-union}(U,P,Q).
\end{aligned}$$

It is easy to check that the τ-models $\mathfrak{A}$ such that $\mathfrak{A} \not\models \varphi_{|U| \neq 2}$ are exactly those models where the extensions of P and Q are distinct singletons and the extension of U is the union of the extensions of P and Q (and thus the cardinality of the extension of U is 2).

We will now show how to enforce Equation (2). The formula $\varphi_{non-stripes}$ below takes care of the cases where (2) does *not* hold. Define

$$\varphi_{non-stripes} := \exists x \exists y \Big(\Big(H(x,y) \wedge \big(C(x) \leftrightarrow \neg C(y) \big) \Big) \vee \Big(V(x,y) \wedge \big(C(x) \leftrightarrow C(y) \big) \Big) \Big).$$

We are now ready to show how to deal with models that violate the last condition of Definition 13. To understand the intended meaning of the following formula, assume that the extension of U is of size two and that the condition given by Equation (2) holds. Note also that from (2) it follows that if such points c and c' exist that violate the last condition of Definition 13, then c and c' agree about C, i.e., we have $C(c)$ iff $C(c')$. We first deal with the case where $C(c)$ and $C(c')$ both hold. We denote by φ_{non-C^+-join} the following formula (whose meaning is fully explained in the proof of Lemma 16):

$$\begin{aligned}
\forall x \Big(\neg U(x) \vee \exists y \Big(C(y) \wedge {=}(y,x) \wedge \exists x \Big({=}(x,y) \wedge \big(\big({=}(x) \wedge H(x,y) \big) \vee \big({=}(x) \wedge V(x,y) \big) \big) \\
\wedge \exists y \big({=}(y) \wedge \big(V(y,x) \vee H(y,x) \big) \wedge \neg C(y) \big) \big) \Big) \Big) \Big).
\end{aligned}$$

To deal with the case where $\neg C(c)$ and $\neg C(c')$, we define the formula φ_{non-C^--join} which is obtained from φ_{non-C^+-join} by simultaneously replacing each $C(x)$ and $C(y)$ by $\neg C(x)$ and $\neg C(y)$, respectively. Finally, we define that $\varphi_{non-join} := \varphi_{non-C^+-join} \vee \varphi_{non-C^--join}$ and $\varphi_{non-grid} := \varphi_{non-serial} \vee \varphi_{|U| \neq 2} \vee \varphi_{non-stripes} \vee \varphi_{non-join}$.

Lemma 16. *Let $\tau = \{V, H, C, U, P, Q\}$ be the vocabulary of striped gridlike structures. Let $\mathfrak{A}$ be a τ-structure such that the extension of U is of cardinality 2. Assume the condition (2) holds. Then $\mathfrak{A} \models \varphi_{non-join}$ iff the last condition of Definition 13 fails in $\mathfrak{A}$.*

Proof. From (2) it follows that if such c and c' exist in $\mathfrak{A}$ that violate the last condition of Definition 13, then c and c' agree on C. We will show that

$$\mathfrak{A} \models \varphi_{non-C^+-join} \text{ iff the last condition of Def. 13 fails in } \mathfrak{A} \text{ for some } c, c' \text{ s.t. } C(c) \;\&\; C(c'). \quad (3)$$

The analogous argument for φ_{non-C^--join} and the case where $\neg C(c)$ and $\neg C(c')$ hold is similar.

Below we denote by $\{(x_1, v_1), ..., (x_k, v_k)\}$ the variable assignment that maps x_i to v_i for each i. Let u, u' be the elements that are in the extension of U in $\mathfrak{A}$. We thus have $\mathfrak{A} \models \varphi_{non-C^+-join}$ iff

$$\mathfrak{A} \models_{X_1} \exists y\Big(C(y) \wedge =\!(y,x) \wedge \exists x\Big(=\!(x,y) \wedge \big(\big(=\!(x) \wedge H(x,y)\big) \\ \vee \big(=\!(x) \wedge V(x,y)\big)\big) \wedge \exists y\big(=\!(y) \wedge \big(V(y,x) \vee H(y,x)\big) \wedge \neg C(y)\big)\big)\Big)\Big),$$

where $X_1 = \{\{(x,u)\}, \{(x,u')\}\}$. Now, recalling that dependence logic has the downwards closure property (cf. proposition 4), we observe that the above holds if and only if there exist *distinct* (distinctness being due to the atom $=\!(y,x)$) points c, c' in the extension of C such that

$$\mathfrak{A} \models_{X_2} \exists x\Big(=\!(x,y) \wedge \big(\big(=\!(x) \wedge H(x,y)\big) \vee \big(=\!(x) \wedge V(x,y)\big)\big) \\ \wedge \exists y\big(=\!(y) \wedge \big(V(y,x) \vee H(y,x)\big) \wedge \neg C(y)\big)\big)\Big),$$

where $X_2 = \{\{(x,u),(y,c)\}, \{(x,u'),(y,c')\}\}$. The above holds if and only if there exist distinct points b, b' of $\mathfrak{A}$ such that $H(b,c)$ and $V(b',c')$ (or $V(b,c)$ and $H(b',c')$ in which case the argument is analogous) and

$$\mathfrak{A} \models_{X_3} \exists y\big(=\!(y) \wedge \big(V(y,x) \vee H(y,x)\big) \wedge \neg C(y)\big)\big),$$

where $X_3 = \{\{(x,b),(y,c)\}, \{(x,b'),(y,c')\}\}$. The above holds if and only if there exists a point a in $\mathfrak{A}$ such that $\neg C(a)$, ($V(a,b)$ or $H(a,b)$) and ($V(a,b')$ or $H(a,b')$). Since $C(c)$ and $C(c')$ hold, it follows from the assumption that (2) holds that $C(b)$ and $\neg C(b')$. Now since also $\neg C(a)$ holds, it follows again from (2) that $V(a,b)$ and $H(a,b')$. When all of the above is combined, we obtain (3). The analogous condition where $\neg C(c)$ and $\neg C(c')$ is proved similarly. Since (2) holds for $\mathfrak{A}$, any points c and c' of $\mathfrak{A}$ that violate the last condition of Definition 13, must agree on C. Thus the lemma holds.

The next lemma follows from Lemma 16 together with the observations made earlier in this section.

Lemma 17. *Let $\tau = \{V, H, C, U, P, Q\}$ be the vocabulary of striped gridlike structures and let $\mathfrak{A}$ be a τ-model. Then $\mathfrak{A}$ is striped and gridlike iff $\mathfrak{A} \not\models \varphi_{non-grid}$.*

Theorem 18. *The validity problem for D^2 is undecidable (more precisely, Σ^0_1-hard).*

Proof. We give a computable reduction from the non-tiling problem to the validity problem of D^2. Since the former is Σ^0_1-complete (Corollary 11), we obtain Σ^0_1-hardness for the latter.

If T is an input to the non-tiling problem, then φ_T denotes the FO^2-sentence given by Lemma 12 and $\varphi_{non-T-tiling} := (\varphi_{non-grid} \vee \varphi_T)$. Let τ be as defined in Lemma 17. Let $C_{\tau,T}$ denote the class of all $\tau \cup T$-structures and let $\mathcal{C}^{\tau,T}_{s-gridlike}$ be the class of exactly all expansions of striped gridlike structures to the vocabulary $\tau \cup T$.

Let T be an input to the non-tiling problem. We will show that the grid is non-T-tilable iff the D^2-sentence $\varphi_{non-T-tiling}$ is valid. By definition, $\varphi_{non-T-tiling}$ is valid iff $\mathfrak{A} \models \varphi_{non-grid} \vee \varphi_T$ holds for every $\mathfrak{A} \in \mathcal{C}_{\tau,T}$. Since $\varphi_{non-grid}$ and φ_T are sentences, the right-hand side of this equivalence is equivalent to the claim that

$$\forall \mathfrak{A} \in \mathcal{C}_{\tau,T} : \mathfrak{A} \models \varphi_{non-grid} \text{ or } \mathfrak{A} \models \varphi_T. \tag{4}$$

By Lemma 17, $\mathfrak{B}^* \models \varphi_{non-grid}$ holds for every τ-reduct $\mathfrak{B}^*$ of $\mathfrak{B} \in \mathcal{C}_{\tau,T}$ that is not striped and gridlike. Hence for every $\mathfrak{B} \in \mathcal{C}_{\tau,T}$ such that the τ-reduct $\mathfrak{B}^*$ of $\mathfrak{B}$ is not striped and gridlike, it holds that $\mathfrak{B} \models \varphi_{non-grid}$. Thus (4) is equivalent to the claim that

$$\forall \mathfrak{A} \in \mathcal{C}^{\tau,T}_{s-gridlike} : \mathfrak{A} \models \varphi_{non-grid} \text{ or } \mathfrak{A} \models \varphi_T. \tag{5}$$

Now let $\mathfrak{B}$ be an arbitrary striped and gridlike τ-structure. By Lemma 17, $\mathfrak{B} \not\models \varphi_{non-grid}$. Thus $\mathfrak{B}^* \not\models \varphi_{non-grid}$ for every expansion $\mathfrak{B}^*$ of $\mathfrak{B}$ to the vocabulary $\tau \cup T$. From this it follows that (5) is equivalent to the claim that

$$\forall \mathfrak{A} \in \mathcal{C}^{\tau,T}_{s-gridlike} : \mathfrak{A} \models \varphi_T. \quad (6)$$

Thus, by Lemma 12, (6) holds if and only if every striped gridlike structure is non-T-tilable. Finally, from Lemma 15 it follows that this is equivalent to the claim that the grid is non-T-tilable.

6 Satisfiability of $\exists^*\forall^*$-formulas

In this section we consider the complexity of satisfiability for sentences of dependence logic and its variants in the prefix class $\exists^*\forall^*$. For first-order logic, the satisfiability and finite satisfiability problems of the prefix class $\exists^*\forall^*$ are known to be *NEXPTIME*-complete. The results hold for both the case with equality and the case without equality, see [3].

Let $\mathcal{A}$ be a collection of generalized atoms. We denote by $\exists^*\forall^*[\mathcal{A}]$ the class of sentences of $\mathsf{FO}(\mathcal{A})$ of the form $\exists x_0 \cdots \exists x_n \forall y_0 \cdots \forall y_m \theta$, where θ is a quantifier-free formula whose generalized atoms are in $\mathcal{A}$. It is worth noting that, depending on the set $\mathcal{A}$, the expressive power and complexity of sentences in $\exists^*\forall^*[\mathcal{A}]$ can vary considerably even when $\mathcal{A}$ is finite and contains only computationally non-complex atoms. For example, there are universal sentences of dependence logic that define NP-complete problems [17]. Furthermore, every sentence of inclusion logic is equivalent to a sentence with a prefix of the form $\exists^*\forall^1$ [12] implying that the satisfiability problem of the $\exists^*\forall^*$-fragment of inclusion logic is undecidable.

Recall that we say that a formula φ is *closed under substructures* if for all $\mathfrak{A}$ and X it holds that if $\mathfrak{A} \models_X \varphi$, $\mathfrak{A}' := \mathfrak{A} \upharpoonright B$ and $X' := X \upharpoonright B$ for some $B \subseteq A$, then we have $\mathfrak{A}' \models_{X'} \varphi$.

Lemma 19. *Let $\mathcal{A}$ be a collection of generalized atoms that are closed under substructures. Then the following conditions hold.*

1. *Suppose $\varphi \in \mathsf{FO}[\mathcal{A}]$ is of the form $\forall y_0 \cdots \forall y_m \theta$, where θ is quantifier-free. Then φ is closed under substructures.*
2. *Let $\varphi \in \exists^*\forall^*[\mathcal{A}]$ be a sentence. Then, if φ is satisfiable, φ has a model with at most $max\{1, k\}$ elements, where k refers to the number of existentially quantified variables in φ.*

Proof. We will first prove claim (1). Suppose that $\varphi := \forall y_0 \cdots \forall y_m \theta$. We will first show the claim for quantifier-free formulas θ, i.e., we will show that for all $\mathfrak{A}$, X, $\mathfrak{A}'$, and X' such that $\mathfrak{A}' := \mathfrak{A} \upharpoonright B$ and $X' := X \upharpoonright B$ for some $B \subseteq A$, the following implication holds.

$$\mathfrak{A} \models_X \theta \Rightarrow \mathfrak{A}' \models_{X'} \theta. \quad (7)$$

The claim obviously holds if θ is a first-order literal. If θ is a generalized atom from $\mathcal{A}$, then the claim holds by assumption. The case $\theta := \psi_1 \wedge \psi_2$ follows immediately from the induction hypothesis. Let us then assume that $\theta := \psi_1 \vee \psi_2$. Since $\mathfrak{A} \models_X \theta$, there are sets Y and Z such that $Y \cup Z = X$, $\mathfrak{A} \models_Y \psi_1$ and $\mathfrak{A} \models_Z \psi_2$. By the induction hypothesis, we have $\mathfrak{A}' \models_{Y'} \psi_1$ and $\mathfrak{A}' \models_{Z'} \psi_2$, where $Y' := Y \upharpoonright B$ and $Z' := Z \upharpoonright B$. Since $Y' \cup Z' = X'$, it follows that $\mathfrak{A}' \models_{X'} \theta$.

We will now show that the claim also holds for φ. Suppose that $\mathfrak{A} \models_X \varphi$. Then, by the truth definition, $\mathfrak{A} \models_{X[A/y_0]\cdots[A/y_m]} \theta$. Using (7), we have $\mathfrak{A}' \models_{(X[A/y_0]\cdots[A/y_m])\upharpoonright B} \theta$. It is easy to check that $(X[A/y_0]\cdots[A/y_m]) \upharpoonright B = (X \upharpoonright B)[B/y_0]\cdots[B/y_m]$. Hence we have $\mathfrak{A}' \models_{X'} \varphi$.

Let us then prove 2. Assume φ is a sentence of the form $\exists x_0 \cdots \exists x_n \forall y_0 \cdots \forall y_m \theta$, where θ is quantifier-free, and that there is a structure $\mathfrak{A}$ such that $\mathfrak{A} \models \varphi$. Hence there exists functions F_i such that $\mathfrak{A} \models_X \forall y_0 \cdots \forall y_m \theta$, where $X = \{\emptyset\}[F_0/x_0]\cdots[F_n/x_n]$. Let s be some assignment in X. Let $\mathrm{range}(s)$ denote the set of elements b such that $s(x) = b$ for some variable x in the domain of s. If $\mathrm{range}(s) \neq \emptyset$ define $B := \mathrm{range}(s)$, and if $\mathrm{range}(s) = \emptyset$ (i.e., $s = \emptyset$), define $B = \{b\}$, where b is an arbitrary element in A. By claim (1), the formula $\forall y_0 \cdots \forall y_m \theta$ is closed under substructures. Thus $\mathfrak{A} \upharpoonright B \models_{X \upharpoonright B} \forall y_0 \cdots \forall y_m \theta$. Thus it follows that $\mathfrak{A} \upharpoonright B \models \varphi$.

A generalized atom A_Q is said to be *polynomial time computable* if the question whether $\mathfrak{A} \models_X A_Q(\overline{y}_1, ..., \overline{y}_n)$ holds can be decided in time polynomial in the size of $\mathfrak{A}$ and X. A class of atoms $\mathcal{A}$ is said to be *uniformly polynomial time computable* if there exists a polynomial function $f : \mathbb{N} \to \mathbb{N}$ such that for every atom $A_Q \in \mathcal{A}$ it holds that the question whether $\mathfrak{A} \models_X A_Q(\overline{y}_1, ..., \overline{y}_n)$ holds can be decided in time $f\big(|\mathfrak{A}| + |X| + |A_Q(\overline{y}_1, ..., \overline{y}_n)|\big)$. Note that every finite class of polynomial time computable atoms is also uniformly polynomial time computable.

The following theorem now follows from Lemma 19. We will make use of the recent result of Grädel showing that for a uniformly polynomial time computable collection $\mathcal{A}$ of atoms, the model checking problem for $\mathsf{FO}(\mathcal{A})$-formulas is in *NEXPTIME* [8].

Theorem 20. *Let A_Q be a generalized atom that is closed under substructures and polynomial time computable. Then* $\text{SAT}(\exists^*\forall^*[A_Q])$ *and* $\text{FINSAT}(\exists^*\forall^*[A_Q])$ *are in* 2*NEXPTIME. If τ is a vocabulary consisting of relation symbols of arity at most k, $k \in \mathbb{Z}_+$, then* $\text{SAT}(\exists^*\forall^*[A_Q](\tau))$ *and* $\text{FINSAT}(\exists^*\forall^*[A_Q](\tau))$ *are NEXPTIME-complete.*

Proof. Note first that the lower bounds follow from the fact that both $\text{SAT}(\exists^*\forall^*)$ and $\text{FINSAT}(\exists^*\forall^*)$ are already *NEXPTIME*-complete. It hence suffices to show containments in 2*NEXPTIME* and *NEXPTIME*, respectively.

Let $\varphi \in \exists^*\forall^*[A_Q]$. By Lemma 19, φ is satisfiable if and only if it has a model of cardinality at most $|\varphi|$. We can decide satisfiability of φ as follows: non-deterministically guess a structure $\mathfrak{A}$ of cardinality at most $|\varphi|$ and accept iff $\mathfrak{A} \models \varphi$. By the result of Grädel in [8], the question whether $\mathfrak{A} \models \varphi$ can be checked non-deterministically in exponential time with input $\mathfrak{A}$ and φ. Assume first that the maximum arity of relation symbols that may occur in φ is not a fixed constant. Relation symbols of arity at most $|\varphi|$ may occur in φ. Thus the size of the *binary encoding of a model* $\mathfrak{A}$ of φ such that $A \leq |\varphi|$ is worst case exponential with respect to $|\varphi|$. If, on the other hand, the maximum arity of relation symbols that can occur in φ is a fixed constant, then the size of the encoding of $\mathfrak{A}$ is just worst case polynomial with respect to $|\varphi|$. Therefore it follows that our algorithm for checking satisfiability of φ is in *NEXPTIME* in the case of fixed arity vocabularies and in 2*NEXPTIME* in the general case. The corresponding results for the finite satisfiability problem follow by the observation that $\exists^*\forall^*[A_Q]$ has the finite model property, Lemma 19.

Corollary 21. *Let $\mathcal{A}$ be a uniformly polynomial time computable class of generalized atoms that are closed under substructures. Then* $\text{SAT}(\exists^*\forall^*[\mathcal{A}])$ *and* $\text{FINSAT}(\exists^*\forall^*[\mathcal{A}])$ *are in* 2*NEXPTIME. If τ is a vocabulary consisting of relation symbols of arity at most k, $k \in \mathbb{Z}_+$, then* $\text{SAT}(\exists^*\forall^*[\mathcal{A}](\tau))$ *and* $\text{FINSAT}(\exists^*\forall^*[\mathcal{A}](\tau))$ *are NEXPTIME-complete.*

In the following sense Theorem 20 is optimal: there exists a polynomial time computable generalized atom A_Q such that $\text{SAT}(\exists^3\forall[A_Q])$ and $\text{FINSAT}(\exists^3\forall[A_Q])$ are undecidable. This already holds for vocabularies with at least one binary relation symbol and a countably infinite set of unary relation symbols. Let $\varphi_{5-inc} := \forall x_1 \ldots \forall x_5 \big(R(x_1, \ldots, x_5) \to S(x_1, \ldots x_5)\big)$, and let A_{5-inc} be the related generalized atom of the type $(5, 5)$, i.e., A_{5-inc} is the 5-ary inclusion atom interpreted as a generalized atom. Clearly A_{5-inc} is computable in polynomial time.

Theorem 22. *Let τ be a vocabulary consisting of one binary relation symbol and a countably infinite set of unary relation symbols. Then both* $\text{SAT}(\exists^3\forall[A_{5-inc}](\tau))$ *and* $\text{FINSAT}(\exists^3\forall[A_{5-inc}](\tau))$ *are undecidable.*

Proof. It well known that for the Kahr class (i.e., the prefix class $\forall\exists\forall$ of FO with vocabulary τ) the satisfiability and the finite satisfiability problems are undecidable (see, e.g., [3]). From the proof of [12, Theorem 5] it follows that there exists a polynomial time translation $\varphi \mapsto \varphi^*$ from the Kahr class into $\exists^3\forall[A_{5-inc}](\tau)$ such that $\mathfrak{A} \models_X \varphi \Leftrightarrow \mathfrak{A} \models_X \varphi^*$ holds for every model $\mathfrak{A}$ and team X with codomain A. Thus $\text{SAT}(\exists^3\forall[A_{5-inc}](\tau))$ and $\text{FINSAT}(\exists^3\forall[A_{5-inc}](\tau))$ are undecidable.

It is easy to see that dependence atoms viewed as generalized atoms are closed under substructures because they are both downwards closed and universe independent. Likewise, it is straightforward to check that the class of dependence atoms is uniformly polynomial time computable. Hence we obtain the following corollary.

Corollary 23. *Both the satisfiability and the finite satisfiability problems for the $\exists^*\forall^*$-sentences of dependence logic are in 2NEXPTIME. If τ is a vocabulary consisting of relation symbols of arity at most k, then the satisfiability and the finite satisfiability problems for the $\exists^*\forall^*$-sentences of dependence logic over the vocabulary τ are NEXPTIME-complete.*

7 Conclusion

We have tied some loose ends concerning the complexity of predicate logics based on team semantics. Using a general approach, we have shown that the satisfiability and the finite satisfiability problems of the two-variable fragments of inclusion logic, exclusion logic, inclusion/exclusion logic, and independence logic are all *NEXPTIME*-complete. Additionally, we have shown that the satisfiability and the finite satisfiability problems of the prefix class $\exists^*\forall^*$ of dependence logic are *NEXPTIME*-complete for any vocabulary of bounded arity, and in 2*NEXPTIME* in the general case. The general approach we have employed of course also implies a range of other results on team-semantics-based logics. Finally, we have proved that the validity problem of two-variable dependence logic is undecidable, thereby answering an open problem from the literature on team semantics.

This article clears path to a more comprehensive classification of the decidability and complexity of different fragments of logics with generalized atoms and team semantics. In the future, we aim to identify further interesting related systems with a decidable satisfiability problem.

References

1. Saguy Benaim, Michael Benedikt, Witold Charatonik, Emanuel Kieroński, Rastislav Lenhardt, Filip Mazowiecki, and James Worrell. Complexity of two-variable logic on finite trees. In *ICALP (2)*, pages 74–88, 2013.
2. R. Berger. The undecidability of the domino problem. *Memoirs of the American Mathematical Society*, (66):369–395, 1966.
3. Egon Börger, Erich Grädel, and Yuri Gurevich. *The Classical Decision Problem*. Perspectives in Mathematical Logic. Springer, 1997.
4. Witold Charatonik and Piotr Witkowski. Two-variable logic with counting and trees. In *LICS*, pages 73–82. IEEE Computer Society, 2013.
5. A. Durand, J. Kontinen, and H. Vollmer. *Expressivity and Complexity of Dependence Logic, in Dependence Logic: Theory and Applications*. Springer, In Press, 2016.
6. Pietro Galliani. Inclusion and exclusion dependencies in team semantics - on some logics of imperfect information. *Ann. Pure Appl. Logic*, 163(1):68–84, 2012.
7. Pietro Galliani and Lauri Hella. Inclusion logic and fixed point logic. In *proceedings of CSL 2013*, pages 281–295, 2013.
8. Erich Grädel. Model-checking games for logics of imperfect information. *Theor. Comput. Sci.*, 493:2–14, 2013.
9. Erich Grädel, Phokion G. Kolaitis, and Moshe Y. Vardi. On the decision problem for two-variable first-order logic. *The Bulletin of Symbolic Logic*, 3(1):53–69, 1997.
10. Erich Grädel, Martin Otto, and Eric Rosen. Undecidability results on two-variable logics. In *Proceedings of STACS '97*, pages 249–260, London, UK, 1997. Springer-Verlag.
11. Erich Grädel and Jouko Väänänen. Dependence and independence. *Studia Logica*, 101(2):399–410, 2013.
12. Miika Hannula. Hierarchies in inclusion logic with lax semantics. In Mohua Banerjee and Shankara Narayanan Krishna, editors, *Logic and Its Applications: 6th Indian Conference, ICLA 2015, Mumbai, India, January 8-10, 2015. Proceedings*, pages 100–118, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.
13. Lauri Hella and Antti Kuusisto. One-dimensional fragment of first-order logic. In *Advances in Modal Logic 10, invited and contributed papers from the tenth conference on "Advances in Modal Logic," held in Groningen, The Netherlands, August 5-8, 2014*, pages 274–293, 2014.
14. Jaakko Hintikka and Gabriel Sandu. Informational independence as a semantical phenomenon. In J. E. Fenstad, I. T. Frolov, and R. Hilpinen, editors, *Logic, Methodology and Philosophy of Science VIII*, volume 126, pages 571–589. Elsevier, Amsterdam, 1989.
15. Wilfrid Hodges. Compositional semantics for a language of imperfect information. *Log. J. IGPL*, 5(4):539–563 (electronic), 1997.

16. Emanuel Kieroński, Jakub Michaliszyn, Ian Pratt-Hartmann, and Lidia Tendera. Two-variable first-order logic with equivalence closure. *SIAM Journal of Computing*, 43(3):1012–1063, 2014.
17. Jarmo Kontinen. Coherence and computational complexity of quantifier-free dependence logic formulas. *Studia Logica*, 101(2):267–291, 2013.
18. Juha Kontinen, Antti Kuusisto, Peter Lohmann, and Jonni Virtema. Complexity of two-variable dependence logic and if-logic. *Inf. Comput.*, 239:237–253, 2014.
19. Antti Kuusisto. A double team semantics for generalized quantifiers. *Journal of Logic, Language and Information*, 24(2):149–191, 2015.
20. Per Lindström. First order predicate logic with generalized quantifiers. *Theoria*, 32:186–195, 1966.
21. Leszek Pacholski, Wieslaw Szwast, and Lidia Tendera. Complexity of two-variable logic with counting. In *Proceedings of LICS '97*, pages 318 –327, 1997.
22. Ian Pratt-Hartmann. Complexity of the two-variable fragment with counting quantifiers. *Journal of Logic, Language and Information*, 14(3):369–395, 2005.
23. Wieslaw Szwast and Lidia Tendera. FO^2 with one transitive relation is decidable. In *STACS*, pages 317–328, 2013.
24. Jouko Väänänen. *Dependence logic: A new approach to independence friendly logic*. Number 70 in London Mathematical Society student texts. Cambridge University Press, 2007.
25. Jonni Virtema. *Approaches to Finite Variable Dependence: Expressiveness and Computational Complexity*. PhD thesis, University of Tampere, 2014.