Tractability Frontier of Data Complexity in Team Semantics

Arnaud Durand
IMJ-PRG, CNRS UMR 7586
Université Paris-Diderot, France

Juha Kontinen
Department of Mathematics and Statistics
University of Helsinki, Finland

Nicolas de Rugy-Altherre
IMJ-PRG, CNRS UMR 7586
Université Paris-Diderot, France

Jouko Väänänen
Department of Mathematics and Statistics
University of Helsinki, Finland
Institute for Logic, Language and Computation
University of Amsterdam, The Netherlands

We study the data complexity of model-checking for logics with team semantics. For dependence and independence logic, we completely characterize the tractability/intractability frontier of data complexity of both quantifier-free and quantified formulas. For inclusion logic formulas, we reduce the model-checking problem to the satisfiability problem of so-called *Dual-Horn* propositional formulas. Via this reduction, we give an alternative proof for the recent result showing that the data complexity of inclusion logic is in PTIME.

1 Introduction

In this article we study the data complexity of model-checking of dependence, independence, and inclusion logic formulas. Independence and inclusion logic [10, 4] are variants of dependence logic [17] that extends first-order logic by dependence atoms of the form $=(x_1,\ldots,x_n)$ expressing that the value of x_n is functionally determined by the values of the variables $x_1,\ldots,x_{n-1}$. In independence and inclusion logic dependence atoms are replaced by independence and inclusion atoms $\overline{y} \perp_{\overline{x}} \overline{z}$ and $\overline{x} \subseteq \overline{y}$, respectively. The meaning of the independence atom is that, with respect to any fixed value of $\overline{x}$, the variables $\overline{y}$ are independent of the variables $\overline{z}$, whereas the inclusion atom expresses that all the values of $\overline{x}$ appear also as values for $\overline{y}$.

Dependence logic is a new framework for formalizing and studying various notions of dependence and independence pervasive in many areas of science. The novelty of dependence logic is in its *team semantics* in which formulas are interpreted using sets of assignments (with a common finite domain $\{x_1,\ldots,x_n\}$ of variables) instead of single assignments as in first-order logic. Reflecting this, dependence logic has higher expressive power than classical logics used for these purposes previously. Dependence, inclusion, and independence atoms are intimately connected to the corresponding functional, inclusion, and multivalued dependencies studied in database theory, see, e.g., [11]. Interestingly, independence atoms can also be viewed as a qualitative analogue of the notion of conditional independence in statistics, see [7]. Furthermore, a variant of dependence logic is in the heart of *Inquisitive Semantics* which is a novel approach in linguistics that analyzes information exchange through communication, see [1].

Dependence logic and its variants can be used to formalize and study dependence and independence notions in various areas. For example, in the foundations of quantum mechanics, there are a range of notions of independence playing a central role in celebrated No-Go results such as Bell's theorem. Abramsky and Väänänen have recently showed that, under a relational view on these results, some of these No-Go results can be logically formalized and syntactically derived using the axioms of independence and dependence atoms. For another application of team semantics in quantum information theory,

J. Esparza, E. Tronci (Eds.): Games, Automata,
Logics and Formal Verification 2015 (GandALF'15)
EPTCS 193, 2015, pp. 73–85, doi:10.4204/EPTCS.193.6

see [13]. Similarly, in the foundations of social choice theory, there are results such as Arrow's Theorem which can also be formalized in the team semantics setting.

For the applications it is important to understand the complexity theoretic aspects of dependence logic and its variants. During the past few years, these aspects have been addressed in several studies. We will next briefly discuss some previous work. The data complexity of inclusion logic is sensitive to the choice between the two main variants of team semantics: under the so-called lax semantics it is equivalent to positive greatest fixed point logic (GFP^+) and captures PTIME over finite (ordered) structures [6]. On the other hand, under the strict semantics, inclusion logic is equivalent to ESO and hence captures NP [5]. The question whether there is a natural fragment of dependence logic capturing PTIME was recently considered in [3] and a fragment D^*-Horn satisfying D^*-Horn $=$ SO$\exists$-Horn $=$ PTIME over finite successor structures was identified.

In [2] the fragment of dependence logic allowing only sentences in which dependence atoms of arity at most k may appear (atoms $=(x_1,\ldots,x_n)$ satisfying $n \leq k+1$) was shown to correspond to the k-ary fragment $\mathrm{ESO}_f(k\text{-ary})$ of ESO in which second-order quantification is restricted to at most k-ary functions and relations. Also, the fragment $\mathrm{D}(k\forall)$ in which at most k variables are allowed to be universally quantified was related to a fragment $\mathrm{ESO}_f(k\forall)$ of ESO consisting of Skolem normal form sentences with at most k universal first-order quantifiers. Similar results have been obtained for independence and inclusion logic (for the strict semantics) in [5, 12].

The combined complexity of the model-checking problem of dependence logic, and many of its variants, was recently shown to be NEXPTIME-complete [9]. On the other hand, the satisfiability problem for the two variable fragment of dependence logic was shown to be NEXPTIME-complete in [15]. Recently, this result has been generalized to cover many of the variants of dependence logic [16].

The starting point for the present work are the following results of [14] showing that the non-classical interpretation of disjunction in team semantics makes the model-checking of certain quantifier-fee formulas very complicated. Define ϕ_1 and ϕ_2 as follows:

1. ϕ_1 is the formula $=(x,y) \vee =(u,v)$, and
2. ϕ_2 is the formula $=(x,y) \vee =(u,v) \vee =(u,v)$.

Surprisingly, the data complexity of the model-checking problem of ϕ_1 and ϕ_2 is already NL-complete and NP-complete, respectively. In [14] it was also shown that model-checking for $\varphi \vee \psi$ where φ and ψ are 2-*coherent* quantifier-free formulas of D is always in NL. A formula φ is called k-coherent if, for all $\mathfrak{A}$ and X, $\mathfrak{A} \models_X \phi$, if and only if, $\mathfrak{A} \models_Y \phi$ for all $Y \subseteq X$ such that $|Y| = k$. Note that the left-to-right implication is always true due to the downwards closure property of dependence logic formulas. The downwards closure property also implies that, for dependence logic formulas, the strict and the lax semantics are equivalent. For independence and inclusion logic formulas this is not the case.

In this article our goal is to give as complete picture as possible of the tractability frontier of data complexity of model-checking of formulas of dependence, independence, and inclusion logic under the lax team semantics. In order to state our results, we define a new syntactic measure called the disjunction-width $\mathrm{d}_\vee(\varphi)$ of a formula φ. Our results show that, for quantifier-free formulas φ of dependence logic, the data complexity of model-checking is in NL if $\mathrm{d}_\vee(\varphi) \leq 2$. Surprisingly, for independence logic the case of quantifier-free formulas turns out to be more fine grained. We give a complete characterization also in this case and, in particular, exhibit a quantifier-free formula with $\mathrm{d}_\vee(\varphi) \leq 2$ whose data-complexity is NP-complete and a seemingly maximal fragment in NL. For quantified formulas, the complexity is shown to be NP-complete already with simple formulas constructed in terms of existential quantification and conjunction in the empty non-logical vocabulary.

For inclusion logic, we show that model-checking can be reduced to the satisfiability problem of dual-Horn propositional formulas. While interesting in its own right, this also provides an alternative proof of the recent result of [6] showing that the data complexity of inclusion logic is in PTIME, and is also analogous to the classical result of Grädel on certain Horn fragments of second-order logic [8].

2 Preliminaries

In this section we briefly discuss the basic definitions and results needed in this article.

Definition 1. Let $\mathfrak{A}$ be a structure with domain A, and $V = \{x_1, \ldots, x_k\}$ be a finite (possibly empty) set of variables.

- A *team* X of $\mathfrak{A}$ with domain $\mathrm{Dom}(X) = V$ is a finite set of assignments $s\colon V \to A$.
- For a tuple $\overline{x} = (x_1, \ldots, x_n)$, where $x_i \in V$, $X(\overline{x}) := \{s(\overline{x}) : s \in X\}$ is the n-ary relation of A, where $s(\overline{x}) := (s(x_1), \ldots, s(x_n))$.
- For $W \subseteq V$, $X \upharpoonright W$ denotes the team obtained by restricting all assignments of X to W.
- The set of free variables of a formula φ is defined as in first-order logic and is denoted by $\mathrm{Fr}(\phi)$.

We are now ready to define team semantics. As now customary, we will restrict attention to formulas in negation normal form, and use the Lax semantics introduced in [4] that differs slightly from the semantics defined in [17]. Below $\mathfrak{A} \models_s \alpha$ refers to the satisfaction in first-order logic, and $s(m/x)$ is the assignment such that $s(m/x)(x) = m$, and $s(m/x)(y) = s(y)$ for $y \neq x$. The power set of a set A is denoted by $\mathscr{P}(A)$.

Definition 2. Let $\mathfrak{A}$ be a structure, X be a team of A, and φ be a first-order formula such that $\mathrm{Fr}(\varphi) \subseteq \mathrm{Dom}(X)$.

lit: For a first-order literal α, $\mathfrak{A} \models_X \alpha$ if and only if, for all $s \in X$, $\mathfrak{A} \models_s \alpha$.

$\vee$: $\mathfrak{A} \models_X \psi \vee \theta$ if and only if, there are Y and Z such that $Y \cup Z = X$, $\mathfrak{A} \models_Y \psi$ and $\mathfrak{A} \models_Z \theta$.

$\wedge$: $\mathfrak{A} \models_X \psi \wedge \theta$ if and only if, $\mathfrak{A} \models_X \psi$ and $\mathfrak{A} \models_X \theta$.

$\exists$: $\mathfrak{A} \models_X \exists x\psi$ if and only if, there exists a function $F : X \to \mathscr{P}(A) \setminus \{\emptyset\}$ such that $\mathfrak{A} \models_{X(F/x)} \psi$, where $X(F/x) = \{s(m/x) : s \in X, m \in F(s)\}$.

$\forall$: $\mathfrak{A} \models_X \forall x\psi$ if and only if, $\mathfrak{A} \models_{X(A/x)} \psi$, where $X(A/x) = \{s(m/x) : s \in X, m \in A\}$.

A sentence ϕ is *true* in $\mathfrak{A}$ (abbreviated $\mathfrak{A} \models \phi$) if $\mathfrak{A} \models_{\{\emptyset\}} \phi$. Sentences ϕ and ϕ' are *equivalent*, $\phi \equiv \phi'$, if for all models $\mathfrak{A}$, $\mathfrak{A} \models \phi \Leftrightarrow \mathfrak{A} \models \phi'$.

First-order formulas satisfy what is known as the *Flatness* property: $\mathfrak{A} \models_X \phi$, if and only if, $\mathfrak{A} \models_s \phi$ for all $s \in X$. Next we will give the semantic clauses for the new dependency atoms:

Definition 3.

- Let $\overline{x}$ be a tuple of variables and let y be another variable. Then $=(\overline{x}, y)$ is a *dependence atom*, with the semantic rule

  $\mathfrak{A} \models_X =(\overline{x}, y)$ if and only if for all $s, s' \in X$, if $s(\overline{x}) = s'(\overline{x})$, then $s(y) = s'(y)$;

- Let $\overline{x}$, $\overline{y}$, and $\overline{z}$ be tuples of variables (not necessarily of the same length). Then $\overline{x} \perp_{\overline{y}} \overline{z}$ is a *conditional independence atom*, with the semantic rule

  $\mathfrak{A} \models_X \overline{x} \perp_{\overline{y}} \overline{z}$ if and only if for all $s, s' \in X$ such that $s(\overline{y}) = s'(\overline{y})$, there exists a $s'' \in X$ such that $s''(\overline{xyz}) = s(\overline{xy})s'(\overline{z})$.

Furthermore, when $\overline{z}$ is empty, we write $\overline{x}\perp\overline{y}$ as a shorthand for $\overline{x}\perp_{\overline{z}}\overline{y}$, and call it a *pure independence atom*;

- Let $\overline{x}$ and $\overline{y}$ be two tuples of variables of the same length. Then $\overline{x}\subseteq\overline{y}$ is an *inclusion atom*, with the semantic rule

  $\mathfrak{A}\models_X \overline{x}\subseteq\overline{y}$ if and only if for all $s\in X$ there exists a $s'\in X$ such that $s'(\overline{y}) = s(\overline{x})$.

The formulas of dependence logic, D, are obtained by extending the syntax of FO by dependence atoms. The semantics of D-formulas is obtained by extending Definition 2 by the semantic rule defined above for dependence atoms. Independence logic, FO($\perp_c$), and inclusion logic, FO($\subseteq$), are defined analogously using independence and inclusion atoms, respectively.

It is easy to see that the flatness property is lost immediately when FO is extended by any of the above dependency atoms. On the other hand, it is straightforward to check that all D-formulas satisfy the following strong *Downwards Closure* property: if $\mathfrak{A}\models_X\phi$ and $Y\subseteq X$, then $\mathfrak{A}\models_Y\phi$. Another basic property shared by all of the above logics is called *Locality*: $\mathfrak{A}\models_X\phi$, if and only if, $\mathfrak{A}\models_{X\upharpoonright \mathrm{Fr}(\phi)}\phi$.

In this article we study the data complexity of model-checking of dependence, independence, and inclusion logic formulas. In other words, for a fixed formula φ of one of the aforementioned logics, we study the complexity of the following model-checking problem: given a model $\mathfrak{A}$ and a team X, decide whether $\mathfrak{A}\models_X\phi$.

We assume that the reader is familiar with the basics of complexity theory.

3 Dependence and independence logics

In this section we consider the complexity of model-checking for quantifier-free and quantified formulas of dependence and independence logic.

3.1 The case of quantifier-free formulas

In this section we consider the complexity of model-checking for quantifier-free formulas of dependence and independence logic. For dependence logic the problem has already been essentially settled in [14]. The following theorems delineate a clear barrier between tractability and intractability for quantifier-free dependence logic formulas.

Theorem 4 ([14])**.** *The model checking problem for formula* $=(x,y)\vee=(z,v)$ *is* NL*-complete. More generally, the model-checking for* $\varphi\vee\psi$ *where* φ *and* ψ *are* 2*-coherent quantifier-free formulas of* D *is always in* NL.

When two disjunctions can be used, the model checking problem becomes intractable as shown by the following results.

Theorem 5 ([14])**.** *The model checking problem for formula* $=(x,y)\vee=(z,v)\vee=(z,v)$ *is* NP*-complete.*

In order to give a syntactic analogue of Theorem 4, we define next the disjunction-width of a formula.

Definition 6. Let σ be a relational signature. The disjunction-width of a σ-formula φ, denoted $\mathsf{d}_\vee(\varphi)$, is defined as follows:

$$\mathsf{d}_\vee(\varphi)=\begin{cases} 1 \text{ if } \varphi \text{ is } \overline{y}\perp_{\overline{x}}\overline{z} \text{ or } =(\overline{x},y) \text{ or } \overline{x}\subseteq\overline{y} \\ 0 \text{ if } \varphi \text{ is } R(\overline{x}) \text{ or } \neg R(\overline{x}), \text{ for } R\in\sigma\cup\{=\} \\ \max(\mathsf{d}_\vee(\varphi_1),\mathsf{d}_\vee(\varphi_2)) \text{ if } \varphi \text{ is } \varphi_1\wedge\varphi_2 \\ \mathsf{d}_\vee(\varphi_1)+\mathsf{d}_\vee(\varphi_2) \text{ if } \varphi \text{ is } \varphi_1\vee\varphi_2 \\ \mathsf{d}_\vee(\varphi_1) \text{ if } \varphi \text{ is } \exists x\varphi_1 \text{ or } \forall\varphi_1. \end{cases}$$

The next theorem is a syntactically defined analogue of Theorem 4.

Proposition 7. *The data complexity of model-checking of quantifier-free* D*-formulas ϕ with $d_\vee(\phi) \leq 2$ is in* NL.

Proof. We will first show that a formula ϕ with $\mathsf{d}_\vee(\phi) = 1$ is 2-coherent. This follows by induction using the following facts [14]:

- dependence atoms are 2-coherent, and first-order formulas are 1-coherent,
- if ψ is k-coherent, then $\psi \vee \phi$ is also k-coherent assuming ϕ is first-order,
- if ψ is k-coherent and ϕ is $k \leq j$-coherent, then $\psi \wedge \phi$ is j-coherent.

It is also straightforward to check that the data complexity of a formula ϕ with $\mathsf{d}_\vee(\phi) = 1$ is in Logspace (the formula ϕ can be expressed in FO assuming the team X with domain $\overline{x} = \mathrm{Fr}(\phi)$ is represented by the n-ary relation $X(\overline{x})$). We will complete the proof using induction on φ with $\mathsf{d}_\vee(\phi) = 2$. Suppose that $\varphi = \psi \vee \phi$, where $\mathsf{d}_\vee(\psi) = \mathsf{d}_\vee(\phi) = 1$. Then the claim follows by Theorem 4. The case $\varphi = \psi \wedge \phi$ is also clear. Suppose finally that $\varphi = \psi \vee \phi$, where ϕ is first-order. Note that by downward closure

$$\mathfrak{A} \models_X \varphi \Leftrightarrow \mathfrak{A} \models_{X'} \psi,$$

where $X' = \{s \in X \mid \mathfrak{A} \not\models_s \phi\}$. Now since X' can be computed in Logspace, the model-checking problem of ϕ can be decided in NL by the induction assumption for ψ. □

In this section we examine potential analogues of Theorems 4 and 5 for independence logic. It is well-known that the dependence atom $=(\overline{x},y)$ is logically equivalent to the independence atom $y \perp_{\overline{x}} y$. Hence, the following is immediate from Theorem 5 [14].

Corollary 8. *The model checking problem for formula*

$$y \perp_x y \vee v \perp_z v \vee v \perp_z v$$

is NP*-complete.*

For independence logic, the situation is not as clear, in particular concerning tractability. In the following we will exhibit a fragment of independence logic whose data complexity is in NL and which is in some sense the maximal such fragment.

Definition 9. The Boolean closure of an independence atom by first-order formulas, denoted $\mathsf{BC}(\perp, \mathrm{FO})$, is defined as follows:

- Any independence atom $\overline{x} \perp_{\overline{y}} \overline{z}$ is in $\mathsf{BC}(\perp, \mathrm{FO})$.
- If $\varphi \in \mathsf{BC}(\perp, \mathrm{FO})$, then for any formula $\phi \in \mathrm{FO}$, $\varphi \wedge \phi$ and $\varphi \vee \phi$ are in $\mathsf{BC}(\perp, \mathrm{FO})$.

Let $\varphi \in \mathsf{BC}(\perp, \mathrm{FO})$. Up to permutation of disjunction and conjunction, φ can be put into the following normalized form:

$$\varphi \equiv ((\ldots((\overline{x} \perp_{\overline{z}} \overline{y} \wedge \phi_1) \vee \psi_1) \wedge \ldots) \wedge \phi_k) \vee \psi_k$$

Let $\mathfrak{A}$ be any structure, $\mathfrak{C}^+ = \bigcap_{i=1}^k \phi_i(\mathfrak{A})$ and $\mathfrak{C}^- = \bigcup_{i=1}^k \psi_i(\mathfrak{A})$, where, $\phi_i(\mathfrak{A})$ is the set of assignments $s\colon \mathrm{Fr}(\varphi) \to A$ such that $\mathfrak{A} \models_s \phi_i$. We can restate the fundamental property for satisfiability of an independence atom in a team (and a structure) to tackle the case of $\mathsf{BC}(\perp, \mathrm{FO})$ formulas. It holds that, for any $\varphi \in \mathsf{BC}(\perp, \mathrm{FO})$, any team X and structure $\mathfrak{A}$, $\mathfrak{A} \models_X \varphi$ if and only if:

- for all $s \in X$: either $s \in [((..(\psi_1 \wedge \phi_2) \vee \dots) \wedge \phi_k) \vee \psi_k](\mathfrak{A})$ or $s \in \mathfrak{C}^+$, $s \notin \mathfrak{C}^-$ and
- for all $s_1, s_2 \in X$ such that $s_1, s_2 \in \mathfrak{C}^+$, $s_1, s_2 \notin \mathfrak{C}^-$, and $s_1(\overline{z}) = s_2(\overline{z})$, there exists $s_3 \in X$ such that: $s_3(\overline{z}) = s_1(\overline{z}), s_3(\overline{x}) = s_1(\overline{x})$ and $s_3(\overline{y}) = s_2(\overline{y})$.

The first item is true by exhaustive case distinction. The second one comes from the fact that if a tuple s satisfies $s \in \mathfrak{C}^+$ and $s \notin \mathfrak{C}^-$ then it is forced to be in the sub-team satisfying $\overline{x} \perp_{\overline{z}} \overline{y}$.

In the rest of the paper, assignments s_1, s_2 as in the second item will be said *compatible* for formula φ and team X and s_3 is called a *witness* of s_1, s_2 (for formula φ).

Since checking whether a tuple s belongs to the query result $\phi(\mathfrak{A})$ of a first-order formula can be done in logarithmic space, deciding whether $\mathfrak{A} \models_X \varphi$ is in Logspace. The following tractability result can be obtained.

Theorem 10. *The data complexity of the model checking problem for formulas of the form $\varphi_1 \vee \varphi_2$ with $\varphi_1, \varphi_2 \in \mathsf{BC}(\perp, \mathrm{FO})$ is in* NL.

Proof. The proof is given by a Logspace reduction to the satisfiability problem of 2-CNF formulas which is well-known to be in NL. Given a structure $\mathfrak{A}$ and a team X we construct a 2-CNF propositional formula Φ such that:

$$\mathfrak{A} \models_X \varphi_1 \vee \varphi_2 \iff \Phi \text{ is satisfiable.} \tag{1}$$

Recall that if a team X is such that $\mathfrak{A} \models_X \varphi_1 \vee \varphi_2$ then, there exists $Y, Z \subseteq X$ such that $Y \cup Z = X$ and $\mathfrak{A} \models_Y \varphi_1$ and $\mathfrak{A} \models_Z \varphi_2$. For each assignment $s \in X$, we introduce two Boolean variables $Y[s]$ and $Z[s]$. Our Boolean formula Φ will be defined below with these $2|X|$ variables the set of which is denoted by $\mathrm{Var}(\Phi)$. It will express that the set of assignments must split into Y and Z but also make sure that incompatible assignments do not appear in the same subteam.

For each pair s_i, s_j that are incompatible for φ_1 on team X, one adds the 2-clause: $\neg Y[s_i] \vee \neg Y[s_j]$. The conjunction of these clauses is denoted by C_Y. Similarly, for each pair s_i, s_j that are incompatible for φ_2 on team X, one adds the clause: $\neg Z[s_i] \vee \neg Z[s_j]$ and call C_Z the conjunction of these clauses.

Finally, the construction of φ is completed by adding the following conjunction:

$$C \equiv \bigwedge_{s \in X} Y[s] \vee Z[s].$$

It is not hard to see, due to the remark on compatible pairs, that the formula $\Phi \equiv C \wedge C_Y \wedge C_Z$ can be built in Logspace. It remains to show that the equivalence (1) holds.

Assume that the left-hand side of the equivalence holds. Then, there exists $Y, Z \subseteq X$ such that $Y \cup Z = X$, $\mathfrak{A} \models_Y \varphi_1$ and $\mathfrak{A} \models_Z \varphi_2$. We construct a propositional assignment $I : \mathrm{Var}(\Phi) \to \{0,1\}$ as follows. For all $s \in Y$, we set $I(Y[s]) = 1$ and for all $s \in Z$, we set similarly $I(Z[s]) = 1$.

Let us consider a clause $\neg Y[s_i] \vee \neg Y[s_j]$ for incompatible s_i, s_j. Then, $I(Y[s_i]) = 0$ or $I(Y[s_j]) = 0$ must hold. For a contradiction, suppose that $I(Y[s_i]) = I(Y[s_j]) = 1$. Then since $\mathfrak{A} \models_Y \varphi_1$ holds, by construction s_i and s_j must be compatible for φ_1. Hence we get a contradiction and may conclude that I satisfies $\neg Y[s_i] \vee \neg Y[s_j]$. The situation is similar for each clause $\neg Z[s_i] \vee \neg Z[s_j]$. Finally since $X = Y \cup Z$, I also satisfies C.

Let us then assume that Φ is satisfiable, and let $I : \mathrm{Var}(\Phi) \to \{0,1\}$ be a satisfying assignment for Φ. Since $I \models C$, we get that $I(Y[s]) = 1$ or $I(Z[s]) = 1$ for all $s \in X$. Let

$$X_Y = \{s : I(Y[s]) = 1\} \text{ and } X_Z = \{s : I(Z[s]) = 1\}.$$

Note that $X_Y \cup X_Z = X$. We will next show how the sets X_Y and X_Z can be extended to sets Y and Z such that $\mathfrak{A} \models_Y \varphi_1$ and $\mathfrak{A} \models_Z \varphi_2$. Note first that, since I satisfies Φ, for all $s_1, s_2 \in X_Y$, Φ cannot have

a clause of the form $\neg Y[s_1] \vee \neg Y[s_2]$, and hence s_1, s_2 are compatible for φ_1. Analogously we see that all $s_1, s_2 \in X_Z$ are compatible for φ_2. We will define the sets Y and Z incrementally by first initializing them to X_Y and X_Z, respectively. Note that even if $X_Y \cup X_Z = X$, no decision has been made regarding the membership of assignments s in Y (resp. Z) such that $I(Y[s]) = 0$ (resp. $I(Z[s]) = 0$). Let us first consider Y. Until no changes occur, we consider all pairs $s_1, s_2 \in Y$ such that $s_1(\overline{z}) = s_2(\overline{z})$ and add into Y (if they are not already in) all tuples $s_3 \in X$ such that s_3 is a witness for the pair (s_1, s_2) regarding property φ_1. Since by construction s_1, s_2 are compatible then at least one such s_3 exists (but may be out of Y till now). We prove below that this strategy is safe. First of all, it is easily seen that any pair among $\{s_1, s_2, s_3\}$ is compatible for φ_1. Therefore, it remains to show that the new assignments s_3 are compatible with every other element s added to Y so far. Suppose this is not the case and that there exists $s \in Y \setminus \{s_1, s_2\}$ such that s_3 and s are incompatible for φ_1. In passing one must have $s_3(\overline{z}) = s(\overline{z})$. Since s_1, s_2, and s are in Y they are all pairwise compatible. Hence, there exists t_1 such that t_1 is a witness for the pair (s_1, s).

Then, $t_1(\overline{x}) = s_1(x) = s_3(\overline{x})$, and $t_1(\overline{y}) = s(\overline{y})$. Consequently, t_1 is also a witness for s_3, s hence, s_3 and s are compatible which is a contradiction. Therefore, the assignment s_3 can be safely added to Y. The set Z is defined analogously. By the construction, it holds that $\mathfrak{A} \models_Y \varphi_1$ and $\mathfrak{A} \models_Z \varphi_2$.

□

We will next show that a slight relaxation on the form of the input formula immediately yields intractability of model-checking.

Theorem 11. *The exists a formula $\varphi_1 \vee \varphi_2$ such that $\varphi_1 \in \mathsf{BC}(\perp, \mathrm{FO})$ and φ_2 is the conjunction of two independence atoms whose model-checking problem is* NP*-complete.*

Proof. Define $\varphi_1 \equiv w \neq 1 \wedge x \perp_t y$, $\varphi_2 \equiv c_1 \perp_c c_2 \wedge x \perp_z y$. We will reduce 3-SAT to the model-checking problem of $\varphi_1 \vee \varphi_2$. Let $\Phi = \bigwedge_{i=1}^{n} C_i$ be a 3-SAT instance. Each $C_i = p_{i_1} \vee p_{i_2} \vee p_{i_3}$ with $p_{i_1}, p_{i_2}, p_{i_3} \in \{v_1, \ldots, v_m, \neg v_1, \ldots, \neg v_m\}$. To this instance we associate a universe $\mathfrak{A}$ and a team X on the variables $w, c, c_1, c_2, z, x, y, t$. The structure $\mathfrak{A}$ is composed of m new elements $a_1, \ldots, a_m$ and of $\{v_1, \ldots, v_m, \neg v_1, \ldots, \neg v_m\} \cup \{0, 1\}$. For each clause C_i we add in X the 6 assignments displayed on the left below, and for each variable v_i, we add to X the 2 assignments on the right:

w	c	c_1	c_2	z	x	y	t
0	i	1	1	i_1	p_{i_1}	p_{i_1}	a_{6i+1}
0	i	1	1	i_2	p_{i_2}	p_{i_2}	a_{6i+2}
0	i	1	1	i_3	p_{i_3}	p_{i_3}	a_{6i+3}
1	i	0	0	0	0	0	a_{6i+4}
1	i	1	0	0	0	0	a_{6i+4}
1	i	0	1	0	0	0	a_{6i+4}

w	c	c_1	c_2	z	x	y	t
0	0	0	0	i	v_i	v_i	$a_{6(n+1)+i}$
0	0	0	0	i	$\neg v_i$	$\neg v_i$	$a_{6(n+1)+i}$

We will next show that Φ is satisfiable if and only if $\mathfrak{A} \models_X \phi$.

⇒ Suppose there is an assignment $I : \{v_1, \ldots, v_m\} \to \{0, 1\}$ that evaluates Φ to true, i.e., at least one literal in each clause is evaluated to 1. We have to split X into two sub-teams $X = Y \cup Z$ such that $\mathfrak{A} \models_Y (w \neq 1 \wedge x \perp_t y)$ and $\mathfrak{A} \models_Z (c_1 \perp_c c_2 \wedge x \perp_z y)$. We must put every assignment $s \in X$ such that $s(w) = 1$ in Z. There are exactly three such assignments per clause. We put in Z every assignment s such that $s(x) = v_i$ if $I(v_i) = 1$, and $s(x) = \neg v_i$ if $I(v_i) = 0$. The other assignments are put into Y.

For each clause C_i, one literal $p_{i_1}, p_{i_2}, p_{i_3}$ is assigned to 1 by I. Then there is at least one assignment $s(c, c_1, c_2) = (i, 1, 1)$ in Z. In Z, the assignments mapping c to i map (c, c_1, c_2) to $(i, 1, 1), (i, 1, 0), (i, 0, 1)$ or $(i, 0, 0)$. Thus $\mathfrak{A} \models_Z c_1 \perp_c c_2$.

If $s_1, s_2 \in Z$ are such that $s_1(z) = s_2(z) = i$, then $s_1(x)$ (analogously $s_2(x)$) is v_i if $I(v_i) = 1$, $\neg v_i$ otherwise. Therefore, $s_1(x) = s_2(x) = s_1(y) = s_2(y)$, and hence $\mathfrak{A} \models_Z x \perp_z y$ holds.

As for Y, it is immediate that $\mathfrak{A} \models_Y w \neq 1$. The only pair of assignments s_1, s_2 in X such that $s_1(t) = s_2(t)$ are $(0,0,0,0,i,v_i,v_i,a_k)$ and $(0,0,0,0,i,\neg v_i,\neg v_i,a_k)$. Only one of them is in Y (s_1 if $I(v_i) = 1$, s_2 otherwise). Thus $\mathfrak{A} \models_Y x \perp_t y$.

$\Leftarrow$ Suppose then that $X = Y \cup Z$ such that $\mathfrak{A} \models_Y (w \neq 1 \wedge x \perp_t y)$ and $\mathfrak{A} \models_Z (c_1 \perp_c c_2 \wedge x \perp_z y)$. Define an assignment I of the variables of Φ by: $I(v_i) = 1$ if $s_i^t := (0,0,0,0,i,v_i,v_i,a_k)$ is in Z, $I(v_i) = 0$ if $s_i^f := (0,0,0,0,i,\neg v_i,\neg v_i,a_k)$ is in Z. Since $\mathfrak{A} \models_Z x \perp_z y$, $s_i^f(z) = s_i^t(z)$ and because there is no $s' \in X$ such that $s'(x) = s_i^f(x), s'(y) = s_i^t(y)$ and $s'(z) = s_i^f(z)$, for each i at most one of s_i^f, s_i^t can be in Z. Similarly, because $\mathfrak{A} \models_Y x \perp_t y$, only one of them can be in Y. Thus I is indeed a function.

Since $\mathfrak{A} \models_Z x \perp_z y$ and there is no assignment in X such that $(x,y) \mapsto (v_i, \neg v_i)$, every pair $s_1, s_2 \in Z$ such that $s_1(z) = s_2(z) = i$ must have the same value of x and y. Every assignment representing a clause in Z respects the choice of I. Furthermore, since $\mathfrak{A} \models_Z c_1 \perp_c c_2$ and $(w,c,c_1,c_2) \mapsto (1,i,0,0),(1,i,1,0),(1,i,0,1)$ are in Z, $(1,i,1,1)$ must be in Z, i.e., at least one assignment per clause is in Z. By the above we may conclude that I satisfies Φ: at least one literal per clause is evaluate to 1 by I.

□

Hardness result of Proposition 8 concerns conditional independence atoms. We prove an analog for the case of pure independence below.

Theorem 12. *The model checking problem is* NP*-complete for formula ϕ of the form*

$$\phi \equiv (x \perp y) \vee (x \perp y) \vee (x \perp y) \vee x \neq y$$

Proof. Let $G = (V_G, E_G)$ be a graph, $\mathfrak{A} = V_G$ be a first order structure of the empty signature, and X be the team $X = \{(v,v) \mid v \in V_G\} \cup \{(v_1,v_2),(v_2,v_1) \mid (v_1,v_2) \in E_G\}$ (we write an assignment s with $s(x) = v_1$ and $s(y) = v_2$ succinctly as (v_1,v_2)). We are going to show that G has a 3-clique cover if and only if $\mathfrak{A} \models_X \phi$.

$\Rightarrow$ Suppose that G has a 3-clique cover, i.e., there exists C_1, C_2, C_3 three cliques such that $V_G = V_{C_1} \cup V_{C_2} \cup V_{C_3}$. We have to prove $\mathfrak{A} \models_X \phi$. For $i \in \{1,2,3\}$, let

$$X_i = \{(v,v) \mid v \in C_i\} \cup \{(v_1,v_2),(v_2,v_1) \mid v_1, v_2 \in C_i\}$$

and $X_4 = X \setminus (X_1 \cup X_2 \cup X_3)$.

Because it is a vertex cover, every assignment of the form (v,v) is contained in $X_1 \cup X_2 \cup X_3$ and not in X_4, i.e. $\mathfrak{A} \models_{X_4} x \neq y$.

Let $i \in \{1,2,3\}$ and $s, s' \in X_i$ be two assignments. If $s(x,y) = (v,v)$ and $s'(x,y) = (v',v')$, then $v, v' \in C_i$ and there exists two assignments s_1, s_2 in X_i such that $s_1(x,y) = (v,v')$ and $s_2(x,y) = (v',v)$ by construction. Similarly if $s(x,y) = (v,v)$ and $s'(x,y) = (v_1,v_2)$, there exists in X_i the assignments (v,v_2) and (v_1,v) (even if $v_1 = v$ or $v_2 = v$). Finally, if $s(x,y) = (v_1,v_2)$ and $s'(x,y) = (v_1',v_2')$, the assignments $(v_1,v_1),(v_2,v_2),(v_1',v_1'),(v_2',v_2')$ are in X_i and so are $(v_1,v_2'),(v_1',v_2)$. The above implies that $\mathfrak{A} \models_{X_i} x \perp y$.

$\Leftarrow$ Suppose that $\mathfrak{A} \models_X (x \perp y) \vee (x \perp y) \vee (x \perp y) \vee x \neq y$, then $X = X_1 \cup X_2 \cup X_3 \cup X_4$ such that $\mathfrak{A} \models_{X_i} x \perp y$ for $i \in \{1,2,3\}$ and $\mathfrak{A} \models_{X_4} x \neq y$. For $i \in \{1,2,3\}$ let C_i be the graph whose vertices are $\{v \mid (v,v) \in X_i\}$ and edges are

$$\{(v_1,v_2) \mid (v_1,v_2) \in X_i \text{ and } (v_2,v_1) \in X_i\}.$$

Note that some C_i can be empty but they form a vertex cover of G as no assignment (v,v) is in X_4. If $v,v' \in C_i$ then (v,v) and (v',v') are in X_i. By independence, (v,v') and (v',v) are also in X_i. Therefore the edge (v,v') is in C_i: C_i is a clique. Therefore, G is covered by the three disjoint cliques $C_1' = C_1$, $C_2' = C_2/C_1$ and $C_3' = C_3/(C_1 \cup C_2)$.

□

3.2 The case of quantified formulas

In this section we show that existential quantification even without disjunction makes the model checking problem hard for both dependence and independence logic.

Theorem 13. *There is a formula φ of dependence logic of empty non-logical vocabulary build with $\exists$ and $\wedge$ whose model-checking problem is* NP*-complete.*

Proof. Define the formula ϕ as follows:

$$\phi \equiv \exists x{=}(x,r_1,r_2,e,m) \wedge {=}(v_1,v_2,x).$$

We will reduce the problem of determining whether a graph G with n^2 vertices is n-colorable to the model-checking problem of ϕ. This graph problem is easily seen to be NP-complete.

Let G be a graph with n^2 vertices $V_G = \{\alpha_0,\ldots,\alpha_{n^2-1}\}$, $\mathfrak{A} = \{0,\ldots,n-1\}$ a first order structure of the empty signature and $X = \{s_i^j \mid i \in \{0,\ldots,n^2-1\},\ 0 \leq j \leq i\}$ be a team such that :

- $s_i^j(v_1) = \lfloor i/n \rfloor$ and $s_i^j(v_2) = i \mod n$. In other words, $s_i^j(v_1,v_2)$ is the decomposition of i in base n.
- $s_i^j(r_1) = \lfloor j/n \rfloor$ and $s_i^j(r_2) = j \mod n$. In other words, $s_i^j(r_1,r_2)$ is the decomposition of j in base n.
- $s_i^j(m) = 0$ if $i \neq j$ and $s_i^i(m) = 1$.
- $s_{i'}^i(e) = 1$, if $i' = i$, or if there is an edge between α_i and $\alpha_{i'}$ with $i' > i$. Otherwise $s_{i'}^i(e) = 0$.

For example, for $n = 2$ and $E_G = \{(0,1);(1,2);(0,2);(2,3)\}$, we obtain the following team on the universe $A = \{0,1\}$:

x	v_1	v_2	r_1	r_2	m	e
	0	0	0	0	1	1
	0	1	0	0	0	1
	0	1	0	1	1	1
	1	0	0	0	0	1
	1	0	0	1	0	1
	1	0	1	0	1	1
	1	1	0	0	0	0
	1	1	0	1	0	0
	1	1	1	0	0	1
	1	1	1	1	1	1

$$G \text{ is } n\text{-colourable iff } \mathfrak{A} \models_X \phi$$

We are going to demonstrate that $\mathfrak{A} \models_X \phi$ if and only if G is n-colourable.

First the left to right implication. Since $\mathfrak{A} \models_X \phi$ there exists a mapping $F : X \to \mathscr{P}(A) \setminus \{\emptyset\}$ such that $\mathfrak{A} \models_{X(F/x)} =(x, r_1, r_2, e, m) \wedge =(v_1, v_2, x)$. By downwards closure, we may assume without loss of generality that $F(s)$ is a singleton for all $s \in X$. Since $=(v_1, v_2, x)$ holds, F induces a mapping $F' : V_G \to A$, by $F'(\alpha_i) = s_i^0(x)$. If there is an edge between α_i and $\alpha_{i'}$, $i' > i$, then $s_{i'}^i(e) = 1 = s_i^i(e)$. Furthermore, $s_{i'}^i(r_1, r_2) = s_i^i(r_1, r_2) = i$ but $s_{i'}^i(m) = 0$ and $s_i^i(m) = 1$ Therefore, because the atom $=(x, r_1, r_2, e, m)$ holds, we must have $s_{i'}^i(x) \neq s_{i'}^i(x)$ (and $F'(\alpha_i) \neq F'(\alpha_{i'})$) if there is an edge between α_i and $\alpha_{i'}$. Thus F' is a colouring of G with $|A| = n$ colours.

Let us then consider the right to left implication. Let $c : V_G \to \{0, \ldots, n-1\}$ be an n colouring. We extend X to variable x with a new team X' such that $s_i^j(x) = c(\alpha_i)$. The value of x depends only on i, which is encoded in (v_1, v_2), i.e., $\mathfrak{A} \models_{X'} =(v_1, v_2, x)$.

Let $s_i^j, s_{i'}^{j'}$ be two assignments of X'. Suppose that $s_i^j(r_1, r_2, e) = s_{i'}^{j'}(r_1, r_2, e)$ but $s_i^j(m) \neq s_{i'}^{j'}(m)$. In this case we must check that $s_i^j(x)$ is different from $s_{i'}^{j'}(x)$ (because $\mathfrak{A} \models_{X'} =(x, r_1, r_2, e, m)$). Now it holds that $j = j'$ because $s_i^j(r_1, r_2) = s_{i'}^{j'}(r_1, r_2)$. Furthermore, since $s_i^j(m) \neq s_{i'}^{j'}(m)$, either $i = j$ or $i' = j'$. Let us suppose $i = j$. Because $1 = s_i^i(e) = s_i^j(e) = s_{i'}^{j'}(e) = s_{i'}^i(e)$, there is an edge between α_i and $\alpha_{i'}$ in G. Therefore $c(i) \neq c(i')$ and $s_i^j(x) \neq s_{i'}^{j'}(x)$. □

By encoding dependence atoms in terms of conditional independence atoms we get the analogous results for free for independence logic.

Corollary 14. *There is a formula φ of independence logic of empty non-logical vocabulary build with $\exists$ and $\wedge$ whose model-checking problem is* NP*-complete.*

We end this section by noting that existential quantifiers cannot be replaced by universal quantifiers in the above theorems.

Proposition 15. *The model-checking problem for formulas of dependence or independence logic using only universal quantification and conjunction is in* Logspace.

Proof. We first transform φ into prenex normal-form exactly as in first-order logic [17]. We may hence assume that φ has the form

$$\forall x_1 \ldots \forall x_n \bigwedge \theta_i(x_1, \ldots, x_n, y_1, \ldots, y_m),$$

where θ_i is either a first-order, dependence, or independence atom. Let $\mathfrak{A}$ be a model, and X be a team of A with domain $\{x_1, \ldots, x_n, y_1, \ldots, y_m\}$. As in [17], the formula $\bigwedge \theta_i(x_1, \ldots, x_n, y_1, \ldots, y_m)$ can be expressed by a first-order sentence ψ when the team X is represented by the $n+m$-ary relation $X(\overline{x}, \overline{y})$, that is,

$$\mathfrak{A} \models_X \bigwedge \theta_i(x_1, \ldots, x_n, y_1, \ldots, y_m) \Leftrightarrow (\mathfrak{A}, X(\overline{x}, \overline{y})) \models \psi.$$

Since $X(\overline{x}, \overline{y})$ is a first-order definable extension of $X(\overline{y})$ it is clear that we can construct a FO-sentence ψ' such that

$$\mathfrak{A} \models_X \forall \overline{x} \bigwedge \theta_i(\overline{x}, \overline{y}) \Leftrightarrow (\mathfrak{A}, X(\overline{y})) \models \psi',$$

holds for all structures $\mathfrak{A}$ and teams X with domain $\{y_1, \ldots, y_m\}$. The claim follows from the fact that the data complexity of FO is in Logspace. □

4 Inclusion Logic

In this section we show that the model-checking problem of inclusion logic formulas can be reduced to the satisfiability problem of *dual-Horn* propositional formulas. A propositional formula Φ in conjunctive normal form is called dual-Horn if each of its clauses contain at most one negative literal.

For a team X, $\overline{x} = \{x_{i_1}, ..., x_{i_n}\} \subseteq \mathrm{dom}(X)$, and $s \in X$, we denote by $s(\overline{x})$ the restriction of s to the variables $x_{i_1}, ..., x_{i_n}$. In this section, σ denotes a relational signature.

Proposition 16. *There exists an algorithm which, given $\varphi \in \mathrm{FO}(\subseteq)$, a structure $\mathfrak{A}$ over σ, and a team X such that $\varphi \subseteq \mathrm{dom}(X)$, outputs a propositional formula Ψ in dual-Horn form such that: $\mathfrak{A} \models_X \varphi \iff \Psi$ is satisfiable. Furthermore, when φ is fixed, the algorithm runs in logarithmic space in the size of $\mathfrak{A}$ and X.*

Proof. Let $\varphi, \mathfrak{A}, X$ be as above and $r_X = |\mathrm{dom}(X)|$. For any team X, we will consider the set $\mathfrak{X}$ of propositional variables $X[s]$ for $s \in A^{r_X}$. Starting from φ, $\mathfrak{A}$, and X we decompose step by step formula φ into subformulas (until reaching its atomic subformulas) and different teams Y, Z, ... and control the relationships between the different teams by propositional dual-Horn formulas built over the propositional variables issued from $X, Y, Z, ...$ Let $\mathscr{S} = \{(\varphi, X, r_X)\}$ and $\mathscr{C} = \{X[s] : s \in X\} \cup \{\neg X[s] : s \notin X\}$. The propositional formula Ψ is now constructed inductively as follows.

As long as $\mathscr{S} \neq \emptyset$, we apply the following rule: Pick (φ, X, r) in $\mathscr{S}$ and apply the following rules.

- If φ is $R(\overline{x})$ with $R \in \sigma$ then: $\mathscr{S} := \mathscr{S} \setminus \{(\varphi, X, r)\}$ and $\mathscr{C} := \mathscr{C} \cup \{X[s] \to R(s(\overline{x})) : \text{ for all } s \in A^r\}$. Clearly, it holds that $\mathfrak{A} \models_X R(\overline{x})$ iff $\bigwedge_{s(\overline{x}) \notin R} \neg X[s]$ is satisfiable.
- If φ is $\overline{x} \subseteq \overline{y}$ then: $\mathscr{S} := \mathscr{S} \setminus \{(\varphi, X, r)\}$ and

  $$\mathscr{C} := \mathscr{C} \cup \{X[s] \to \bigvee_{s' \in A^r, s'(\overline{y}) = s(\overline{x})} X[s'] : s \in A^r\}.$$

  It holds that $\mathfrak{A} \models_X \overline{x} \subseteq \overline{y}$ iff $\bigwedge_{s \in A^r} (X[s] \to \bigvee_{s' \in A^r, s'(\overline{y}) = s(\overline{x})} X[s'])$ is satisfiable.
- If φ is $\exists x \psi$, then: $\mathscr{S} := (\mathscr{S} \setminus \{(\varphi, X, r)\}) \cup \{(\psi, Y, r+1)\}$and

  $$\mathscr{C} := \mathscr{C} \cup \{X[s] \to \bigvee_{s' = (s,a),\ a \in A} Y[s'] : s \in A^r\},$$

  where the $Y[s]$, $s \in A^{r+1}$ are new propositional variables (not used in $\mathscr{C}$). If $\mathfrak{A} \models_X \exists x \psi$ then, there exists a function $F : X \to \mathscr{P}(A) \setminus \{\emptyset\}$, such that $\mathfrak{A} \models_{X(F/x)} \psi$. In other words, $\mathfrak{A} \models_Y \psi$ for some team Y defined by the solutions of the constraint $\bigwedge_{s \in A^r} X[s] \to \bigvee_{s' = (s,a),\ a \in A} Y[s']$ (which define a suitable function F). Conversely, if $\mathfrak{A} \models_Y \psi$ for a team Y as above defined from X, then clearly $\mathfrak{A} \models_X \exists x \psi$.
- If φ is $\forall x \psi$, then: $\mathscr{S} := (\mathscr{S} \setminus \{(\varphi, X, r)\}) \cup \{(\psi, Y, r+1)\}$ and

  $$\mathscr{C} := \mathscr{C} \cup \{X[s] \to Y[s'] : s \in A^r, s' \in A^{r+1} \text{ s.t. } s'(\overline{x}) = s(\overline{x})\},$$

  where the $Y[s]$, $s \in A^{r+1}$ are new propositional variables (not used in $\mathscr{C}$). The conclusion is similar as for the preceding case.
- If φ is $\psi_1 \wedge \psi_2$ then: $\mathscr{S} := (\mathscr{S} \setminus \{(\varphi, X, r)\}) \cup \{(\psi_1, X, r), (\psi_2, X, r)\}$ and $\mathscr{C}$ is unchanged. By definition, $\mathfrak{A} \models_X \varphi$ iff $\mathfrak{A} \models_X \psi_1 \wedge \psi_2$.

- If φ is $\psi_1 \vee \psi_2$ then: $\mathscr{S} := (\mathscr{S} \setminus \{(\varphi, X, r)\}) \cup \{(\psi_1, Y, r), (\psi_2, Z, r)\}$ and

$$\mathscr{C} := \mathscr{C} \cup \{X[s] \to Y[s] \vee Z[s] : s \in A^r\} \cup \{Y[s] \to X[s], Z[s] \to X[s] : s \in A^r\}$$

  where again the $Y[s]$ and $Z[s]$, $s \in A^r$ are new propositional variables (not used in $\mathscr{C}$). Here again, $\mathfrak{A} \models_X \varphi$ if and only if $\mathfrak{A} \models_Y \psi_1$ and $\mathfrak{A} \models_Z \psi_2$ for some suitable Y and Z such that $Y \cup Z = X$ which is exactly what is stated by the Boolean constraints.

Observe that each new clause added to $\mathscr{C}$ during the process is of dual-Horn form, i.e., contains at most one negative literal. Observe also, that applied to some (φ, X, r), the algorithm above only adds triples $\mathscr{S}$ whose first component is a proper subformula of φ and eliminates (φ, X, r). When the formula φ is atomic, no new triple is added afterwards. Hence the algorithm will eventually terminate with $\mathscr{S} = \emptyset$. Setting $\Psi := \bigwedge_{C \in \mathscr{C}} C$, it can easily be proved by induction that: $\mathfrak{A} \models_X \varphi$ iff Ψ is satisfiable.

Observe also that each clause in $\mathscr{C}$ can be constructed from X and $\mathfrak{A}$ by simply running through their elements (using their index) hence in logarithmic space. □

Remark 1. The construction of Proposition 16 can be done in principle for any kind of atom: dependence, independence, exclusion, constancy etc. To illustrate this remark, one could translate in the above proof a dependence atom of the form $=(\overline{x}, y)$ by (using the notations of the proof):

$$\bigwedge_{\substack{s,s' \in A^r \\ s(\overline{x}) = s'(\overline{x}) \wedge s(y) \neq s'(y)}} (\neg X[s] \vee \neg X[s']).$$

The additional clauses are of length two. A similar treatment can be done for independence atoms $\overline{x} \perp_{\overline{y}} \overline{z}$. In the two cases however, the resulting formula is not in Dual-Horn form anymore and there is no way to do so (unless PTIME = NP).

Since deciding the satisfiability of a propositional formula in dual-Horn form can be done in polynomial time we obtain the following already known corollary.

Corollary 17. *The data complexity of* FO($\subseteq$) *is in* PTIME.

5 Conclusion

We have studied the tractability/intractability frontier of data complexity of both quantifier-free and quantified dependence and independence logic formulas. Furthermore, we defined a novel translation of inclusion logic formulas into dual-Horn propositional formulas, and used it to show that the data-complexity of inclusion logic is in PTIME. It is an interesting open question whether the translation of Proposition 16 can be generalized to hold for some interesting extensions of FO($\subseteq$) by further dependency atoms.

Acknowledgements

The authors would like to thank Arne Meier for a number of corrections and useful suggestions. The second author was supported by the Academy of Finland grant 264917.

References

[1] Ivano Ciardelli & Floris Roelofsen (2011): *Inquisitive Logic*. *J. Philosophical Logic* 40(1), pp. 55–94, doi:10.1007/s10992-010-9142-6.

[2] Arnaud Durand & Juha Kontinen (2012): *Hierarchies in dependence logic*. *ACM Transactions on Computational Logic (TOCL)* 13(4), p. 31, doi:10.1145/2362355.2362359.

[3] Johannes Ebbing, Juha Kontinen, Julian-Steffen Müller & Heribert Vollmer (2014): *A Fragment of Dependence Logic Capturing Polynomial Time*. *Logical Methods in Computer Science* 10(3), doi:10.2168/LMCS-10(3:3)2014.

[4] Pietro Galliani (2012): *Inclusion and exclusion dependencies in team semantics: On some logics of imperfect information*. *Annals of Pure and Applied Logic* 163(1), pp. 68 – 84, doi:10.1016/j.apal.2011.08.005.

[5] Pietro Galliani, Miika Hannula & Juha Kontinen (2013): *Hierarchies in independence logic*. In Simona Ronchi Della Rocca, editor: *Computer Science Logic 2013 (CSL 2013), Leibniz International Proceedings in Informatics (LIPIcs)* 23, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, pp. 263–280, doi:10.4230/LIPIcs.CSL.2013.263. Available at http://drops.dagstuhl.de/opus/volltexte/2013/4202.

[6] Pietro Galliani & Lauri Hella (2013): *Inclusion Logic and Fixed Point Logic*. In Simona Ronchi Della Rocca, editor: *Computer Science Logic 2013 (CSL 2013), Leibniz International Proceedings in Informatics (LIPIcs)* 23, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, pp. 281–295, doi:10.4230/LIPIcs.CSL.2013.281. Available at http://drops.dagstuhl.de/opus/volltexte/2013/4203.

[7] Dan Geiger, Thomas Verma & Judea Pearl (1990): *Identifying independence in bayesian networks*. *Networks* 20(5), pp. 507–534, doi:10.1002/net.3230200504.

[8] Erich Grädel (1992): *Capturing Complexity Classes by Fragments of Second-Order Logic*. *Theor. Comput. Sci.* 101(1), pp. 35–57, doi:10.1016/0304-3975(92)90149-A.

[9] Erich Grädel (2013): *Model-checking games for logics of imperfect information*. *Theor. Comput. Sci.* 493, pp. 2–14. Available at http://dx.doi.org/10.1016/j.tcs.2012.10.033.

[10] Erich Grädel & Jouko Väänänen (2013): *Dependence and Independence*. *Studia Logica* 101(2), pp. 399–410, doi:10.1007/s11225-013-9479-2.

[11] Miika Hannula & Juha Kontinen (2014): *A Finite Axiomatization of Conditional Independence and Inclusion Dependencies*. In Christoph Beierle & Carlo Meghini, editors: *Foundations of Information and Knowledge Systems - 8th International Symposium, FoIKS 2014, Bordeaux, France, March 3-7, 2014. Proceedings*, *Lecture Notes in Computer Science* 8367, Springer, pp. 211–229, doi:10.1007/978-3-319-04939-7_10.

[12] Miika Hannula & Juha Kontinen (2015): *Hierarchies in independence and inclusion logic with strict semantics*. *J. Log. Comput.* 25(3), pp. 879–897, doi:10.1093/logcom/exu057.

[13] T. Hyttinen, G. Paolini & J. Väänänen (2014): *Quantum team logic and Bell's Inequalities*. *ArXiv e-prints*, doi:10.1017/S1755020315000192.

[14] Jarmo Kontinen (2013): *Coherence and Computational Complexity of Quantifier-free Dependence Logic Formulas*. *Studia Logica* 101(2), pp. 267–291, doi:10.1007/s11225-013-9481-8.

[15] Juha Kontinen, Antti Kuusisto, Peter Lohmann & Jonni Virtema (2014): *Complexity of two-variable dependence logic and IF-logic*. *Inf. Comput.* 239, pp. 237–253, doi:10.1016/j.ic.2014.08.004.

[16] Juha Kontinen, Antti Kuusisto & Jonni Virtema (2014): *Decidable Fragments of Logics Based on Team Semantics*. *CoRR* abs/1410.5037. Available at http://arxiv.org/abs/1410.5037.

[17] Jouko Väänänen (2007): *Dependence Logic*. Cambridge University Press, doi:10.1017/CBO9780511611193.