On the Complexity of Team Logic and Its Two-Variable Fragment

Martin Lück
Institut für Theoretische Informatik, Leibniz Universität Hannover
Appelstraße 4, 30167 Hannover, Germany
lueck@thi.uni-hannover.de

Abstract

We study the logic FO(∼), the extension of first-order logic with team semantics by unrestricted Boolean negation. It was recently shown to be axiomatizable, but otherwise has not yet received much attention in questions of computational complexity. In this paper, we consider its two-variable fragment $\mathrm{FO}^2(\sim)$ and prove that its satisfiability problem is decidable, and in fact complete for the recently introduced non-elementary class TOWER(poly). Moreover, we classify the complexity of model checking of FO(∼) with respect to the number of variables and the quantifier rank, and prove a dichotomy between PSPACE- and ATIME-ALT(exp, poly)-complete fragments. For the lower bounds, we propose a translation from modal team logic MTL to $\mathrm{FO}^2(\sim)$ that extends the well-known standard translation from modal logic ML to FO^2. For the upper bounds, we translate FO(∼) to fragments of second-order logic with PSPACE-complete and ATIME-ALT(exp, poly)-complete model checking, respectively.

2012 ACM Subject Classification Theory of computation → Complexity theory and logic, Theory of computation → Logic

Keywords and phrases team logic, two-variable logic, complexity, satisfiability, model checking

Digital Object Identifier 10.4230/LIPIcs.MFCS.2018.27

Related Version A full version of the paper can be found at [30], `https://arxiv.org/abs/1804.04968`.

Acknowledgements I wish to thank Juha Kontinen, Heribert Vollmer and the anonymous referees for many helpful comments.

1 Introduction

In the past decades, the work of logicians has unearthed a plethora of decidable fragments of first-order logic FO. Many decidability results are rooted in a finite model property: if there exists a (computable) upper bound on the size of minimal models with respect to a class of formulas, and if the logic admits sufficiently feasible model checking, then the question of satisfiability can be settled by exhaustively searching all structures of suitable size. Prominent examples meeting the above criteria are logics with restricted quantifier prefixes, such as the BSR-fragment which contains only $\exists^*\forall^*$-sentences [34]. Others include the monadic class [27], the guarded fragment GF [2], the recently introduced separated fragment SF [36, 37], or the two-variable fragment FO^2 [31, 35, 19], which all are decidable. See also the excellent book by Börger et al. [6] for a comprehensive classification.

The above fragments all have been subject to intensive research with the purpose of further pushing the boundary of decidability. One example is the guarded fixpoint logic, μGF, which extends GF and is 2-EXPTIME-complete [18, 3]. Another is FOC^2, the extension FO^2

43rd International Symposium on Mathematical Foundations of Computer Science (MFCS 2018).
Editors: Igor Potapov, Paul Spirakis, and James Worrell; Article No. 27; pp. 27:1–27:22
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

with counting quantifiers. Due to an exponential model property, satisfiability is NEXPTIME-complete for both FO^2 and FOC^2 [16, 33].

Another novel and very actively studied formalism is *team semantics*, introduced by Hodges [22]. At its core, it refers to the simultaneous evaluation of formulas on whole *sets* of assignments, called *teams*. This extension is conservative in the sense that the evaluation of singleton teams, which consist of a single assignment, coincides with classical Tarski semantics. Logics with team semantics offer many applications in areas such as statistics, database theory, physics, cryptography and social choice theory (see also Abramsky et al. [1]).

As a prototypical logic with team semantics, Väänänen [38] introduced *dependence logic* D. It extends FO by *dependence atoms* $=(x_1, \dots, x_n, y)$, which intuitively state that the value of y in the team functionally depends on the values of $x_1, \dots, x_n$. With respect to expressive power, D coincides with existential second-order logic. Nonetheless, its two-variable fragment D^2 was recently proved by Kontinen et al. [23] to have a NEXPTIME-complete satisfiability problem due to a satisfiability-preserving translation to FOC^2. However, D is not closed under Boolean negation, and the *validity* problem of D^2 is in fact undecidable [24], and non-arithmetical for full D [38]. By adding a negation operator $\sim$ to D, Väänänen [38] introduced *team logic* TL, which is equivalent to full second-order logic SO [25].

As a generalization of TL, we study the logic $\mathsf{FO}(\sim, \mathcal{D})$ introduced by Galliani [13, 12]. It extends FO under team semantics by a Boolean negation $\sim$ and a set $\mathcal{D}$ of so-called generalized dependence atoms (cf. [26]). We focus on FO-definable atoms, which covers the dependence atom and many other important atoms such as the independence $\perp$ [17] or inclusion atom $\subseteq$ [10]. We abbreviate $\mathsf{FO}(\sim, \emptyset)$ as $\mathsf{FO}(\sim)$. While $\mathsf{FO}(\sim)$ and D have incomparable expressive power, in terms of complexity, $\mathsf{FO}(\sim)$ is much weaker than D. In particular, unlike D it is axiomatizable [29] and its validity problem is complete for the class Σ^0_1 of recursively enumerable sets, as with ordinary FO.

As a new result, we prove in Section 4 that its two-variable fragment $\mathsf{FO}^2(\sim)$ is decidable. More precisely, we show that satisfiability and validity of $\mathsf{FO}^2(\sim)$ are complete for the recently introduced non-elementary complexity class TOWER(poly) [28]. This pushes the "decidability frontier" away from FO^2 into a new direction, and creates the curious situation that the satisfiability problem for $\mathsf{FO}^2(\sim)$ is strictly harder than for D^2, while for validity the exact opposite is the case (cf. Table 1).

On the path to decidability, we also investigate the model checking problem of $\mathsf{FO}(\sim, \mathcal{D})$. In the first-order setting, model checking in team semantics has received only little attention so far, unlike the well-understood propositional [21] and modal [9, 32, 39] variants of team logic and dependence logic. In Section 3 and 6, we fill this gap and show that model checking for $\mathsf{FO}(\sim, \mathcal{D})$ (for "well-behaved" $\mathcal{D}$) is complete for the class ATIME-ALT(exp, poly), i.e., for exponential runtime with polynomially many alternations. This complements the result of Grädel [14] that model checking for D is NEXPTIME-complete.

Finally, we also consider fragments $\mathsf{FO}^n_k(\sim, \mathcal{D})$ which have only n variables and quantifier rank k, and relate them to certain "sparse" fragments of SO which we call $\mathsf{SO}[p]$. We prove that model checking of $\mathsf{SO}[p]$ and $\mathsf{FO}^n_k(\sim, \mathcal{D})$ is only PSPACE-complete, as opposed to unrestricted SO and $\mathsf{FO}^\omega_\omega(\sim, \mathcal{D})$.

Due to space constraints, some proofs are moved to the appendix and marked with ($\star$), and can also be found in the full version of this paper [30].

Table 1 Complexity of logics with team semantics. Completeness unless stated otherwise. $\mathcal{D}$ is a set of generalized dependency atoms, the superscript refers to the number of variables, and the subscript to the quantifier rank.

Logic	Satisfiability	Validity	References
FO^2	NEXPTIME	co-NEXPTIME	[19]
D^2	NEXPTIME	Σ^0_1-hard	[24]
$\mathsf{FO}^2(\sim)$	TOWER(poly)	TOWER(poly)	Theorem 6.6
TL^2_2	Π^0_1-hard	Σ^0_1-hard	Theorem 6.7
	Model Checking		
$\mathsf{FO}_k, \mathsf{FO}^n$	$\in$ PTIME		see, e.g., [15]
FO	PSPACE		see, e.g., [15]
$\mathsf{FO}_k(\sim,\mathcal{D}), \mathsf{FO}^n(\sim,\mathcal{D})$	PSPACE		Theorem 6.4
$\mathsf{FO}(\sim,\mathcal{D})$	ATIME-ALT(exp, poly)		Theorem 6.4

2 Preliminaries

The domain of a function f is $\mathsf{dom}\, f$. For $f\colon X \to Y$ and $Z \subseteq X$, $f\restriction Z$ is the restriction of f to the domain Z. The power set of X is $\mathfrak{P}(X)$. The cardinality of the natural numbers is ω. The class of recursively enumerable sets (resp. their complements) is Σ^0_1 (resp. Π^0_1).

Given a logic $\mathcal{L}$, the sets of all satisfiable and valid formulas of $\mathcal{L}$ are written $\mathsf{SAT}(\mathcal{L})$ and $\mathsf{VAL}(\mathcal{L})$, respectively. Likewise, the model checking problem $\mathsf{MC}(\mathcal{L})$ contains the tuples (A, φ) such that φ is an $\mathcal{L}$-formula and A is a model of φ.

We assume the reader to be familiar with basic complexity theory and alternating Turing machines [7]. When stating that a problem is hard or complete for a complexity class $\mathcal{C}$, we refer to logspace-computable reductions. In this paper, we require Turing machines that are restricted in both their runtime and their *alternation depth*, as introduced by Berman [4], where the alternation depth is the maximal number of alternations between existential and universal non-determinism that a given machine performs on any computation path.

In what follows, we use the tetration function $\exp_k$, defined by $\exp_0(n) := n$ and $\exp_{k+1}(n) := 2^{\exp_k(n)}$. We write $\exp(n)$ instead of $\exp_1(n)$.

▶ **Definition 2.1.** For $k \geq 0$, ATIME-ALT$(\exp_k, \text{poly})$ is the class of problems decided by an alternating Turing machine with at most $p(n)$ alternations and runtime at most $\exp_k(p(n))$, for a polynomial p.

▶ **Definition 2.2.** TOWER(poly) is the class of problems that are decided by a deterministic Turing machine in time $\exp_{p(n)}(1)$ for some polynomial p.

The reader may verify that both ATIME-ALT$(\exp_k, \text{poly})$ and TOWER(poly) are closed under all Boolean operations and under polynomial time resp. logspace computable reductions.

First-order Team Logic

A vocabulary τ is a set of function symbols f and predicate symbols P, with their respective arity denoted by $\mathsf{arity}(f)$ and $\mathsf{arity}(P)$. τ is called relational if it contains no function symbols. We explicitly state $= \in \tau$ if we permit equality as part of the syntax. For obvious reasons, we require that a vocabulary always contains at least one predicate or $=$.

We fix a set $\mathsf{Var} = \{x_1, x_2, \ldots\}$ of first-order variables. If $\vec{t}$ is a tuple of τ-terms, $\mathsf{Var}(\vec{t})$ is the set of variables appearing in $\vec{t}$. Formulas are interpreted in τ-structures, denoted as pairs $\mathcal{A} = (A, \tau^{\mathcal{A}})$, with the domain A of $\mathcal{A}$ also written $\mathsf{dom}\,\mathcal{A}$. We sometimes identify $\mathcal{A}$ and $\mathsf{dom}\,\mathcal{A}$ if the meaning is clear. If $s\colon X \to \mathcal{A}$, t is a τ-term, and $\mathsf{dom}\,s \supseteq \mathsf{Var}(t)$, then $t\langle s\rangle \in \mathcal{A}$ is the evaluation of t in $\mathcal{A}$ under s. Likewise, if $\vec{t} = (t_1, \ldots, t_n)$, then $\vec{t}\langle s\rangle := (t_1\langle s\rangle, \ldots, t_n\langle s\rangle)$.

A *team* T (in $\mathcal{A}$) is a set of assignments $s\colon X \to \mathcal{A}$, where X is called domain of T. If $X \supseteq \mathsf{Var}(\vec{t}\,)$ and $\vec{t}$ is a tuple of terms, then $\vec{t}\langle T\rangle := \{\vec{t}\langle s\rangle \mid s \in T\}$. If T is a team with domain $X \supseteq Y$, then its restriction to Y is $T\restriction Y := \{s\restriction Y \mid s \in T\}$. In slight abuse of notation, we sometimes identify a tuple $\vec{x}$ with its underlying set, e.g., write $T\restriction\vec{x}$ for $T\restriction\{x_1, \ldots, x_n\}$.

If $s\colon X \to \mathcal{A}$ and $x \in \mathsf{Var}$, then $s^x_a\colon X \cup \{x\} \to \mathcal{A}$ is the assignment that maps x to a and $y \in X \setminus \{x\}$ to $s(y)$. If T is a team in $\mathcal{A}$ with domain X, then $f\colon T \to \mathfrak{P}(\mathcal{A}) \setminus \{\emptyset\}$ is called a *supplementing function* of T. It extends (or modifies) T to the *supplementing team* $T^x_f := \{\, s^x_a \mid s \in T, a \in f(s) \,\}$. If $f(s) = A$ is constant, we write T^x_A for T^x_f.

In this paper, we consider generalized dependencies in team semantics (cf. [26, 12]), but restrict ourselves to the special case of FO-definable dependencies. For this reason, in our setting, the definition boils down to the following.

▶ **Definition 2.3** (Dependencies). If P is a predicate and $\tau_P = \{P, =\}$, then a τ_P-FO-formula δ is called *dependency.* Furthermore, if $\mathsf{arity}(P) = k$, then δ is also called *k-ary dependency.*

Let $\mathcal{D} = \{\delta_1, \delta_2, \ldots\}$ be a (possibly infinite) set of dependencies. Then we consider special atoms $A_i\vec{t}$, called *generalized dependency atoms*, to represent the dependencies δ_i in the syntax. The logic τ-$\mathsf{FO}(\sim, \mathcal{D})$ extends τ-FO as follows:

$$\varphi ::= \alpha \mid A_i\vec{t} \mid {\sim}\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \exists x\, \varphi \mid \forall x\, \varphi,$$

where α is any τ-FO-formula, $\delta_i \in \mathcal{D}$ is a k-ary dependency, $\vec{t}$ is a k-tuple of τ-terms, and $x \in \mathsf{Var}$. For easier distinction, we usually call classical FO-formulas $\alpha, \beta, \gamma, \ldots$ and reserve $\varphi, \psi, \vartheta, \ldots$ for $\mathsf{FO}(\sim, \mathcal{D})$-formulas. If $\vec{t} = (t_1, \ldots, t_n)$ and $\vec{u} = (u_1, \ldots, u_n)$ are tuples of τ-terms, then we use the shorthand $\vec{t} = \vec{u}$ for $\bigwedge_{i=1}^n t_i = u_i$.

From now on, we usually omit τ. The $\sim$-free fragment of $\mathsf{FO}(\sim, \mathcal{D})$ is $\mathsf{FO}(\mathcal{D})$, and we abbreviate $\mathsf{FO}(\sim, \emptyset)$ as $\mathsf{FO}(\sim)$.

▶ **Example 2.4.** Let $\mathsf{dep} := \{\mathsf{dep}_1, \mathsf{dep}_2, \ldots\}$ be defined by

$$\mathsf{dep}_n(R) := \forall x_1 \cdots \forall x_{n-1} \forall y \forall z (R x_1 \cdots x_{n-1} y \wedge R x_1 \cdots x_{n-1} z \to y = z).$$

Then dep is set of dependencies, and the corresponding atom $A_n\vec{t}$ is called n-ary *dependence atom* and is also written $=\!(t_1, \ldots, t_n)$. It holds $(\mathcal{A}, T) \vDash\, =\!(t_1, \ldots, t_n)$ if and only if for all $s, s' \in T$ we have that $t_1\langle s\rangle = t_1\langle s'\rangle, \ldots, t_{n-1}\langle s\rangle = t_{n-1}\langle s'\rangle$ implies $t_n\langle s\rangle = t_n\langle s'\rangle$. Likewise, for the case $n = 1$, the atom $=\!(t)$ means that t is constant, i.e., $t\langle s\rangle = t\langle s'\rangle$ for all $s, s' \in T$.

In this notation, Väänänen's dependence logic D is $\mathsf{FO}(\mathsf{dep})$, and team logic TL is $\mathsf{FO}(\sim, \mathsf{dep})$ [38]. Many other important atoms are FO-definable, such as independence [17], inclusion and exclusion [10] (see also Durand et al. [8]).

If φ is a formula, $\mathsf{Fr}(\varphi)$ and $\mathsf{Var}(\varphi)$ denote the set of free resp. of all variables in φ, with $\mathsf{Fr}(A_i\vec{t}\,) := \mathsf{Var}(A_i\vec{t}\,) := \mathsf{Var}(\vec{t})$. If $\mathsf{Fr}(\varphi) = \emptyset$, then φ is called sentence. We write $\varphi(x_1, \ldots, x_n)$ to indicate that $x_1, \ldots, x_n$ are free in φ. The *width* $\mathsf{w}(\varphi)$ of φ is $|\mathsf{Var}(\varphi)|$.

The *quantifier rank* $\mathsf{qr}(\varphi)$ of φ is 0 if φ is atomic, and otherwise defined recursively as $\mathsf{qr}({\sim}\varphi) := \mathsf{qr}(\neg\varphi) := \mathsf{qr}(\varphi)$, $\mathsf{qr}(\varphi \wedge \psi) := \mathsf{qr}(\varphi \vee \psi) := \max\{\mathsf{qr}(\varphi), \mathsf{qr}(\psi)\}$, and $\mathsf{qr}(\exists x\, \varphi) := \mathsf{qr}(\forall x\, \varphi) := \mathsf{qr}(\varphi) + 1$, respectively. The fragment of FO with formulas of width at most n

and quantifier rank at most k is FO^n_k. The corresponding fragments D^n_k, TL^n_k, $\mathsf{FO}^n_k(\sim,\mathcal{D})$, $\mathsf{FO}^n_k(\sim)$ and $\mathsf{FO}^n_k(\mathcal{D})$ are defined analogously.

We evaluate τ-$\mathsf{FO}(\sim,\mathcal{D})$-formulas φ on pairs $(\mathcal{A},T)$ as follows, where $\mathcal{A}$ is a τ-structure and T a team in $\mathcal{A}$ with domain $X \supseteq \mathsf{Fr}(\varphi)$:

$$
\begin{aligned}
(\mathcal{A},T) \vDash \varphi &\Leftrightarrow \forall s \in T : (\mathcal{A},s) \vDash \varphi \text{ (in Tarski semantics), for } \varphi \in \mathsf{FO},\\
(\mathcal{A},T) \vDash A_i\vec{t} &\Leftrightarrow \mathcal{A} \vDash \delta_i(\vec{t}\langle T\rangle), \text{ where } \delta_i \in \mathcal{D},\\
(\mathcal{A},T) \vDash {\sim}\psi &\Leftrightarrow (\mathcal{A},T) \nvDash \psi,\\
(\mathcal{A},T) \vDash \psi \wedge \theta &\Leftrightarrow (\mathcal{A},T) \vDash \psi \text{ and } (\mathcal{A},T) \vDash \theta,\\
(\mathcal{A},T) \vDash \psi \vee \theta &\Leftrightarrow \exists S,U \subseteq T \text{ such that } T = S \cup U,\ (\mathcal{A},S) \vDash \psi, \text{ and } (\mathcal{A},U) \vDash \theta,\\
(\mathcal{A},T) \vDash \exists x\, \varphi &\Leftrightarrow (\mathcal{A},T^x_f) \vDash \varphi \text{ for some } f\colon T \to \mathfrak{P}(\mathsf{dom}\,\mathcal{A}) \setminus \{\emptyset\},\\
(\mathcal{A},T) \vDash \forall x\, \varphi &\Leftrightarrow (\mathcal{A},T^x_{\mathsf{dom}\,\mathcal{A}}) \vDash \varphi.
\end{aligned}
$$

A τ-formula φ is *satisfiable* if there exists a τ-structure $\mathcal{A}$ and team T with domain $X \supseteq \mathsf{Fr}(\varphi)$ in $\mathcal{A}$ such that $(\mathcal{A},T) \vDash \varphi$. Likewise, φ is *valid* if $(\mathcal{A},T) \vDash \varphi$ for all such τ-structures $\mathcal{A}$ and teams T.

The so-called *locality* property ensures that the truth of a formula, as in classical semantics, depends only on the assignments to variables that occur free in it.

▶ **Proposition 2.5** (Locality). *Let $\varphi \in \mathsf{FO}(\sim,\mathcal{D})$ and $X \supseteq \mathsf{Fr}(\varphi)$. If T is a team in $\mathcal{A}$ with domain $Y \supseteq X$, then $(\mathcal{A},T) \vDash \varphi$ if and only if $(\mathcal{A},T\upharpoonright X) \vDash \varphi$.*

Proof. Proof by induction on φ. The base case of FO-formulas and the inductive step for $\wedge$, $\vee$, $\exists$ and $\forall$ work similarly to Galliani's proof for inclusion/exclusion logic [10, Theorem 4.22], to which the $\sim$-case can be added in the obvious manner. It remains to consider the dependence atoms $A_i\vec{t}$. As $X \supseteq \mathsf{Fr}(A_i\vec{t}) = \mathsf{Var}(\vec{t})$, clearly $\vec{t}\langle s\rangle = \vec{t}\langle s\upharpoonright X\rangle$ for any $s \in T$, and consequently, $\vec{t}\langle T\rangle = \vec{t}\langle T\upharpoonright X\rangle$. Hence, $\mathcal{A} \vDash \delta_i(\vec{t}\langle T\rangle)$ iff $\mathcal{A} \vDash \delta_i(\vec{t}\langle T\upharpoonright X\rangle)$. ◀

Second-Order Logic

Second-order logic τ-SO (or simply SO) extends τ-FO by second-order quantifiers $\exists f$, $\forall f$, $\exists P$ and $\forall P$ for function and predicate variables. For an SO-formula α, the sets $\mathsf{Var}(\alpha)$ and $\mathsf{Fr}(\alpha)$ refer to all resp. all free variables in α (first-order or second-order). SO is evaluated on pairs $(\mathcal{A},\mathcal{J})$, where $\mathcal{A}$ is a structure and $\mathcal{J}$ maps first-order variables x to elements $\mathcal{J}(x) \in \mathcal{A}$, function variables f to functions $\mathcal{J}(f)\colon \mathcal{A}^{\mathsf{arity}(f)} \to \mathcal{A}$, and predicate variables P to relations $\mathcal{J}(P) \subseteq \mathcal{A}^{\mathsf{arity}(P)}$. The notation $\mathcal{J}^X_Y$ for a (first-order or second-order) variable X and an element resp. function resp. relation Y is defined as in the first-order setting. Instead of $(\mathcal{A},\mathcal{J}) \vDash \alpha(X_1,\ldots,X_n)$ and $\mathcal{J}(X_1) = \mathcal{X}_1,\ldots,\mathcal{J}(X_n) = \mathcal{X}_n$, we also write $\mathcal{A} \vDash \alpha(\mathcal{X}_1,\ldots,\mathcal{X}_n)$.

Second-order model checking, $\mathsf{MC}(\mathsf{SO})$, is decidable using a straightforward algorithm: Given a formula α and a finite input structure $\mathcal{A}$, evaluate α in recursive top-down manner, using non-deterministic guesses for the quantified elements, functions and relations, which are of exponential size with respect to $|\mathsf{dom}\,\mathcal{A}|$.

▶ **Proposition 2.6** ($\star$). $\mathsf{MC}(\mathsf{SO})$ *is decidable on input $(\mathcal{A},\mathcal{J},\alpha)$ in time $2^{n^{\mathcal{O}(1)}}$ and with $|\alpha|$ alternations.*

If the arity of quantified functions and relations is bounded by c, then each quantified function and relation has at most $|\mathsf{dom}\,\mathcal{A}|^c$ elements and hence takes only polynomial space:

▶ **Corollary 2.7.** *Let c-SO be the fragment of SO where all quantified functions and relations have arity at most c. Then $\mathsf{MC}(c\text{-}\mathsf{SO})$ is* PSPACE*-complete.*

3 From FO($\sim$) to SO: Upper bounds for model checking

In this section, we present upper bounds for the model checking problem of $\mathsf{FO}(\sim,\mathcal{D})$. On that account, we assume all first-order structures $\mathcal{A}$ and teams T to be finite and to have a suitable encoding. Instead of deciding $\mathsf{MC}(\mathsf{FO}(\sim,\mathcal{D}))$ directly, we reduce it to the corresponding problem of second-order logic, $\mathsf{MC}(\mathsf{SO})$. For this purpose, we build on top of a result of Väänänen [38], which roughly speaking states that TL-formulas can efficiently be translated to SO.

However, in Väänänen's original translation [38, Theorem 8.12, p. 159] from TL to SO it is assumed that the truth in a team is preserved when taking subteams (which is not the case if $\sim$ is available), and that all variables in a formula are quantified at most once. However, in fragments $\mathsf{FO}^n(\sim,\mathcal{D})$ of finite width n, re-quantification of variables cannot be avoided in general. In what follows, we adapt the translation accordingly. Furthermore, we extend it to include generalized dependency atoms.

Suppose $\vec{x} = (x_1,\ldots,x_n)$ is a tuple of variables. In order to avoid repetitions of variables, we define the notation $\vec{x};y$ as follows: $\vec{x};y := \vec{x}$ if $y \in \vec{x}$, and $\vec{x};y := (x_1,\ldots,x_n,y)$ if $y \notin \vec{x}$. Let now $\varphi \in \mathsf{FO}(\sim,\mathcal{D})$ such that $\mathsf{Fr}(\varphi) \subseteq \vec{x}$, and R be a n-ary predicate. Then we inductively define the SO-formula $\eta^{\vec{x}}_\varphi(R)$ as shown below.

- If φ is a classical first-order formula, then $\eta^{\vec{x}}_\varphi(R) := \forall\vec{x}(R\vec{x} \rightarrow \varphi)$.
- If $\varphi = A_i(\vec{t})$ and $\delta_i \in \mathcal{D}$ is k-ary, then let $\vec{z} = (z_1,\ldots,z_k)$ be pairwise distinct variables disjoint from $\vec{x}$ and $\eta^{\vec{x}}_\varphi(R) := \exists S\,\forall\vec{z}\,(S\vec{z} \leftrightarrow (\exists\vec{x}\,R\vec{x} \wedge \vec{t} = \vec{z}\,)) \wedge \delta_i(S)$.[1]
- If $\varphi = \sim\psi$, then $\eta^{\vec{x}}_\varphi(R) := \neg\eta^{\vec{x}}_\psi(R)$.
- If $\varphi = \psi \wedge \theta$, then $\eta^{\vec{x}}_\varphi(R) := \eta^{\vec{x}}_\psi(R) \wedge \eta^{\vec{x}}_\theta(R)$.
- If $\varphi = \psi \vee \theta$, then $\eta^{\vec{x}}_\varphi(R) := \exists S\,\exists U\;\forall\vec{x}(R\vec{x} \leftrightarrow (S\vec{x} \vee U\vec{x})) \wedge \eta^{\vec{x}}_\psi(S) \wedge \eta^{\vec{x}}_\theta(U)$.
- If $\varphi = \exists y\,\psi$, then $\eta^{\vec{x}}_\varphi(R) := \exists S\,\forall\vec{x}((\exists y R\vec{x}) \leftrightarrow (\exists y\, S\vec{x};y)) \wedge \eta^{\vec{x};y}_\psi(S)$.
- If $\varphi = \forall y\,\psi$, then $\eta^{\vec{x}}_\varphi(R) := \exists S\,\forall\vec{x}((\exists y R\vec{x}) \leftrightarrow (\exists y\, S\vec{x};y)) \wedge \eta^{\vec{x};y}_\psi(S) \wedge \forall\vec{x}(R\vec{x} \rightarrow \forall y\, S\vec{x};y)$.

By an inductive proof, the formulas φ and $\eta^{\vec{x}}_\varphi(R)$ can be shown equivalent, provided that the team T is represented as a relation $R := \vec{x}\langle T\rangle$:

▶ **Theorem 3.1** ($\star$). *Let $\varphi \in \mathsf{FO}(\sim,\mathcal{D})$, let $\vec{x} \supseteq \mathsf{Fr}(\varphi)$ be a tuple of variables, and T be a team in $\mathcal{A}$ with domain $Y \supseteq \vec{x}$. Then $(\mathcal{A},T) \vDash \varphi$ if and only if $\mathcal{A} \vDash \eta^{\vec{x}}_\varphi(\vec{x}\langle T\rangle)$.*

▶ **Definition 3.2.** We call a set $\mathcal{D} = \{\delta_1,\delta_2,\ldots\}$ of dependencies *p-uniform* if there is a polynomial time algorithm that for all i, when given $A_i\vec{t}$, computes δ_i.

▶ **Corollary 3.3.** *Let $\mathcal{D}$ be a p-uniform set of dependencies. Then $\mathsf{MC}(\mathsf{FO}(\sim,\mathcal{D}))$ is decidable on input $(\mathcal{A},T,\varphi)$ in time $2^{n^{\mathcal{O}(1)}}$ and with $|\varphi|^{\mathcal{O}(1)}$ alternations.*

Proof. First, we compute $\vec{x} := \mathsf{Fr}(\varphi)$, the formula $\eta^{\vec{x}}_\varphi$ and the relation $\vec{x}\langle T\rangle$ from φ and T in polynomial time. When translating the atoms $A_i\vec{t}$, we apply the p-uniformity of $\mathcal{D}$. Afterwards, we accept if and only if $\mathcal{A} \vDash \eta^{\vec{x}}_\varphi(\vec{x}\langle T\rangle)$. By Proposition 2.6, the latter can be checked by an algorithm with $|\eta^{\vec{x}}_\varphi|$ alternations and time exponential in $(\mathcal{A},\vec{x}\langle T\rangle,\eta^{\vec{x}}_\varphi)$. In total, this leads to $|\varphi|^{\mathcal{O}(1)}$ alternations and runtime exponential in the size of $(\mathcal{A},T,\varphi)$. ◀

[1] Note that the "obvious" translation $\eta^{\vec{x}}_\varphi(R) := \delta_i(R)$ does not work in general if $A_i(\vec{t})$ contains proper terms. For instance, any team T satisfies $=\!(c)$ for any constant term c, but R represents only $\vec{x}\langle T\rangle$, which might or might not satisfy δ_i. To properly reflect such atoms, we quantify $\vec{t}\langle T\rangle$ in the relation S; in our example, $S = \{(c)\}$) for $T \neq \emptyset$.

Sparse second-order logic

The complexity of the model checking problem of $\mathsf{FO}(\sim, \mathcal{D})$ significantly drops if either the number of variables or the quantifier rank is bounded by an arbitrary constant. To prove this, we introduce a fragment of SO that corresponds to these restricted fragments of $\mathsf{FO}(\sim, \mathcal{D})$. We call this fragment *sparse second-order logic*, based on *sparse quantifiers* $\exists^p$ and $\forall^p$:

$$(\mathcal{A}, \mathcal{J}) \vDash \exists^p P\, \psi \Leftrightarrow \text{there exists } R \subseteq \mathcal{A}^{\mathsf{arity}(P)} \text{ such that } |R| \leq p(|\mathcal{A}|) \text{ and } (\mathcal{A}, \mathcal{J}^P_R) \vDash \psi,$$
$$(\mathcal{A}, \mathcal{J}) \vDash \forall^p P\, \psi \Leftrightarrow \text{for all } R \subseteq \mathcal{A}^{\mathsf{arity}(P)} \text{ such that } |R| \leq p(|\mathcal{A}|) \text{ it holds } (\mathcal{A}, \mathcal{J}^P_R) \vDash \psi,$$

where $p \colon \mathbb{N} \to \mathbb{N}$ and $|\mathcal{A}| := |\mathsf{dom}\, \mathcal{A}| + \sum_{X \in \tau} |X^{\mathcal{A}}|$. In other words, all quantified relations have bounded cardinality relative to the underlying structure. For obvious reasons, there are no sparse function quantifiers.

The logic $\mathsf{SO}[p]$ is now defined as SO, but with only $\exists^p$ and $\forall^p$ as permitted second-order quantifiers. Consider the case where p is bounded by a polynomial. The interpretation of each quantified relation then contains at most $|\mathcal{A}|^{\mathcal{O}(1)}$ tuples. Consequently, on $\mathsf{SO}[p]$-formulas, the recursive model checking algorithm from Proposition 2.6 then runs in polynomial time:

▶ **Corollary 3.4.** *If p is bounded by a polynomial, then $\mathsf{MC}(\mathsf{SO}[p])$ is decidable on input $(\mathcal{A}, \mathcal{J}, \alpha)$ in polynomial time and with $|\alpha|$ alternations.*

It remains to show that the translation from team logic with bounded width or quantifier rank takes place in this fragment of SO. This can be seen as follows. Intuitively, every quantified relation in $\eta^{\vec{x}}_{\varphi}$ represents either a subteam of an existing team (for the $\vee$-case), or it is a supplementing team (for the $\exists$-case and $\forall$-case). For this reason, the cardinality of the quantified relations grows at most by a factor of $|\mathsf{dom}\, \mathcal{A}|$ for every occurrence of $\vee$, $\exists$ or $\forall$.

Now, for $p \colon \mathbb{N} \to \mathbb{N}$, define the $\mathsf{SO}[p]$-formula $\zeta^{\vec{x},p}_{\varphi}$ like $\eta^{\vec{x}}_{\varphi}$, but with all second-order quantifiers replaced by $\exists^p$. The next theorem states that $\zeta^{\vec{x},p}_{\varphi}$ is an appropriate translation of φ, similarly to $\eta^{\vec{x}}_{\varphi}$, if φ has sufficiently small width or quantifier rank:

▶ **Theorem 3.5** ($\star$)**.** *Let $\varphi \in \mathsf{FO}(\sim, \mathcal{D})$, let $\vec{x} \supseteq \mathsf{Fr}(\varphi)$ be a tuple of variables, and T be a team in $\mathcal{A}$ with domain $Y \supseteq \vec{x}$. If $p(n) \geq |T| \cdot n^{\mathsf{qr}(\varphi)}$ or $p(n) \geq n^{\mathsf{w}(\varphi)}$, then $(\mathcal{A}, T) \vDash \varphi$ if and only if $\mathcal{A} \vDash \zeta^{\vec{x},p}_{\varphi}(\vec{x}\langle T \rangle)$.*

Proof. By a careful analysis, it can be shown that all second-order quantifiers $\exists S$ in $\eta^{\vec{x}}_{\varphi}$ can be replaced by $\exists^p S$. See the appendix for details. As then $\eta^{\vec{x}}_{\varphi}$ and $\zeta^{\vec{x},p}_{\varphi}$ agree on $(\mathcal{A}, T)$, the claim follows by Theorem 3.1. ◀

▶ **Corollary 3.6.** *Let $\mathcal{D}$ be a p-uniform set of dependencies and $m < \omega$. $\mathsf{MC}(\mathsf{FO}^m_{\omega}(\sim, \mathcal{D}))$ and $\mathsf{MC}(\mathsf{FO}^{\omega}_m(\sim, \mathcal{D}))$ are then decidable on input $(\mathcal{A}, T, \varphi)$ in polynomial time with $|\varphi|^{\mathcal{O}(1)}$ alternations.*

Proof. Let $p(n) := n^{m+1}$. Analogously to Corollary 3.3, we reduce both $\mathsf{MC}(\mathsf{FO}^m_{\omega}(\sim, \mathcal{D}))$ and $\mathsf{MC}(\mathsf{FO}^{\omega}_m(\sim, \mathcal{D}))$ to $\mathsf{MC}(\mathsf{SO}[p])$. Assume that $(\mathcal{A}, T, \varphi)$ is the input, and that either $\mathsf{w}(\varphi) \leq m$ or $\mathsf{qr}(\varphi) \leq m$. Then the input is mapped to $(\mathcal{A}, \vec{x}\langle T \rangle, \zeta^{p,\vec{x}}_{\varphi})$, where $\vec{x} = \mathsf{Fr}(\varphi)$.

- If $\mathsf{w}(\varphi) \leq m$, then $(\mathcal{A}, T) \vDash \varphi$ if and only if $\mathcal{A} \vDash \zeta^{p,\vec{x}}_{\varphi}(\vec{x}\langle T \rangle)$ by Theorem 3.5.
- If $\mathsf{qr}(\varphi) \leq m$, then w.l.o.g. $|T| \leq |\mathcal{A}|$ (if necessary, pad $\mathcal{A}$ with a dummy predicate in polynomial time). Then $|T| \cdot |\mathcal{A}|^m \leq p(|\mathcal{A}|)$, and we can again apply Theorem 3.5. ◀

4 From $\mathsf{FO}^2(\sim)$ to FO^2: Upper bounds for satisfiability

In this section, we turn to the satisfiability problem of $\mathsf{FO}^2(\sim)$ and prove that it is complete for TOWER(poly) (cf. Definition 2.2). Our approach is to establish a finite model property for $\mathsf{FO}^2(\sim)$. However, instead of constructing a finite model directly, we reduce $\mathsf{FO}^2(\sim)$-formulas to FO^2-formulas, and use the exponential model property of FO^2 [19]. As a first step, we expand $\mathsf{FO}^2(\sim)$-formulas into a specific "disjunctive normal form" over $\wedge$ and $\sim$. Recall that $\vee$ is not the Boolean disjunction, which we instead define via $\varphi \otimes \psi := \sim(\sim\varphi \wedge \sim\psi)$. We also use the abbreviation $\mathsf{E}\beta := \sim\neg\beta$ ("at least one assignment in the team satisfies β").

▶ **Lemma 4.1** ($\star$). *Every τ-$\mathsf{FO}^n_k(\sim)$-formula φ is equivalent to a formula of the form*

$$\psi := \bigotimes_{i=1}^{n} \left(\alpha_i \wedge \bigwedge_{j=1}^{m_i} \mathsf{E}\beta_{i,j} \right)$$

such that $\{\alpha_1, \ldots, \alpha_n, \beta_{1,1}, \ldots, \beta_{n,m_n}\} \subseteq \tau\text{-}\mathsf{FO}^n_k$ and $|\psi| \leq \exp_{\mathcal{O}(|\varphi|)}(1)$.

Proof. The proof is by induction on φ, and consists of repeatedly applying the following distributive laws. Here, $\varphi \equiv \psi$ means that φ and ψ are logically equivalent. See the appendix for details.

$$\begin{aligned}
\alpha \wedge \bigwedge_{i=1}^{n} \mathsf{E}\beta_i &\equiv \bigvee_{i=1}^{n} (\alpha \wedge \mathsf{E}\beta_i) & \bigvee_{i=1}^{n} (\alpha_i \wedge \mathsf{E}\beta_i) &\equiv \Big(\bigvee_{i=1}^{n} \alpha_i\Big) \wedge \bigwedge_{i=1}^{n} \mathsf{E}(\alpha_i \wedge \beta_i)\\
(\vartheta_1 \otimes \vartheta_2) \vee \vartheta_3 &\equiv (\vartheta_1 \vee \vartheta_3) \otimes (\vartheta_2 \vee \vartheta_3) & \vartheta_1 \vee (\vartheta_2 \otimes \vartheta_3) &\equiv (\vartheta_1 \vee \vartheta_2) \otimes (\vartheta_1 \vee \vartheta_3)\\
(\vartheta_1 \otimes \vartheta_2) \wedge \vartheta_3 &\equiv (\vartheta_1 \wedge \vartheta_3) \otimes (\vartheta_2 \wedge \vartheta_3) & \vartheta_1 \wedge (\vartheta_2 \otimes \vartheta_3) &\equiv (\vartheta_1 \wedge \vartheta_2) \otimes (\vartheta_1 \wedge \vartheta_3)\\
\exists v\, (\vartheta_1 \otimes \vartheta_2) &\equiv (\exists v\, \vartheta_1) \otimes (\exists x\, \vartheta_2) & \exists v\, (\vartheta_1 \vee \vartheta_2) &\equiv (\exists v\, \vartheta_1) \vee (\exists v\, \vartheta_2)\\
\exists v\, (\alpha \wedge \mathsf{E}\beta) &\equiv (\exists v\, \alpha) \wedge \mathsf{E}\, \exists v\, (\alpha \wedge \beta) & \forall v\, (\vartheta_1 \wedge \vartheta_2) &\equiv (\forall v\, \vartheta_1) \wedge (\forall v\, \vartheta_2)\\
\forall v \sim\vartheta &\equiv \sim\forall v\, \vartheta
\end{aligned}$$

◀

▶ **Theorem 4.2.** *If τ is a relational vocabulary, then every satisfiable $\varphi \in \tau\text{-}\mathsf{FO}^2(\sim)$ has a model of size $\exp_{\mathcal{O}(|\varphi|)}(1)$.*

Proof. Let $\varphi \in \tau\text{-}\mathsf{FO}^2(\sim)$ be satisfiable. φ is equivalent to a disjunction of size $\exp_{\mathcal{O}(|\varphi|)}(1)$ as stated in Lemma 4.1. Clearly, this disjunction must have at least one satisfiable disjunct, which is of the form

$$\psi = \alpha \wedge \bigwedge_{i=1}^{m} \mathsf{E}\beta_i,$$

for $\{\alpha, \beta_1, \ldots, \beta_m\} \subseteq \tau\text{-}\mathsf{FO}^2$ and w.l.o.g. $\mathsf{Var}(\psi) \subseteq \{x, y\}$. Let $(\mathcal{A}, T)$ be a model of ψ. For every i, as ψ implies $\mathsf{E}(\alpha \wedge \beta_i)$, there exists $s \in T$ such that $(\mathcal{A}, s) \vDash \alpha \wedge \beta_i$. But then $\mathcal{A}$ also satisfies – in Tarski semantics – the classical FO^2-sentence

$$\gamma := \bigwedge_{i=1}^{m} \exists x\, \exists y\, \alpha \wedge \beta_i,$$

as $s(x)$ and $s(y)$ are witnesses for $\exists x$ and $\exists y$. However, by the exponential model property of FO^2 [19], there exists a model $\mathcal{B}$ of size $2^{\mathcal{O}(|\gamma|)}$ for γ. As for every i there is $\hat{s}_i \colon \{x, y\} \to \mathcal{B}$ such that $(\mathcal{B}, \hat{s}_i) \vDash \alpha \wedge \beta_i$, we conclude $(\mathcal{B}, \{\hat{s}_1, \ldots, \hat{s}_m\}) \vDash \psi$ by definition of team semantics. Clearly, this shows that ψ and hence φ has a model of size $\exp_{\mathcal{O}(|\varphi|)}(1)$. ◀

▶ **Corollary 4.3.** *If* τ *is a relational vocabulary, then* $\mathsf{SAT}(\tau\text{-}\mathsf{FO}^2(\sim))$ *and* $\mathsf{VAL}(\tau\text{-}\mathsf{FO}^2(\sim))$ *are in* $\mathrm{TOWER}(\mathrm{poly})$*.*

Proof. By Corollary 3.6, model checking for $\mathsf{FO}^2(\sim)$ is possible in alternating polynomial time, and hence in deterministic exponential time. The following is a $\mathrm{TOWER}(\mathrm{poly})$-algorithm for $\mathsf{SAT}(\tau\text{-}\mathsf{FO}^2(\sim))$. Given a formula φ, iterate over all interpretations $(\mathcal{A}, T)$ of size $\exp_{\mathcal{O}(|\varphi|)}(1)$ and accept if $(\mathcal{A}, T) \vDash \varphi$. The algorithm for $\mathsf{VAL}(\tau\text{-}\mathsf{FO}^2(\sim))$ is similar. ◀

5 From MTL to $\mathsf{FO}^2(\sim)$: A team-semantical standard translation

In order to prove the lower bounds for $\mathsf{FO}^2(\sim)$ and $\mathsf{FO}^n_k(\sim)$, we reduce from the corresponding satisfiability, validity and model checking problems of so-called *modal team logic* MTL. This logic was introduced by Müller [32], and extends classical modal logic ML by $\sim$ in the same fashion as $\mathsf{FO}(\sim)$ extends FO.

We fix a countably infinite set Φ of propositions. MTL is defined as follows, where φ denotes an MTL-formula, α an ML-formula, and p a proposition.

$$\varphi ::= {\sim}\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \Box\varphi \mid \Diamond\varphi \mid \alpha \qquad\qquad \alpha ::= \neg\alpha \mid \alpha \wedge \alpha \mid \alpha \vee \alpha \mid \Box\alpha \mid \Diamond\alpha \mid p$$

The modal depth $\mathsf{md}(\varphi)$ of φ is defined recursively, i.e., $\mathsf{md}(p) := 0$, $\mathsf{md}({\sim}\varphi) := \mathsf{md}(\neg\varphi) := \mathsf{md}(\varphi)$, $\mathsf{md}(\varphi \wedge \psi) := \mathsf{md}(\varphi \vee \psi) := \max\{\mathsf{md}(\varphi), \mathsf{md}(\psi)\}$, and $\mathsf{md}(\Box\varphi) := \mathsf{md}(\Diamond\varphi) := \mathsf{md}(\varphi)+1$. MTL_k is the fragment of MTL with modal depth at most k. The set of propositional variables occurring in $\varphi \in \mathsf{MTL}$ is written $\mathsf{Prop}(\varphi)$.

Let $X \subseteq \Phi$ be finite. Then, a Kripke structure (over X) is a tuple $\mathcal{K} = (W, R, V)$, where W is a set of *worlds* or *points*, (W, R) is a directed graph, and $V \colon X \to \mathfrak{P}(W)$. If $w \in W$, then $(\mathcal{K}, w)$ is called *pointed Kripke structure*.

ML is evaluated on pointed Kripke structures in the classical Kripke semantics, whereas MTL is evaluated on pairs $(\mathcal{K}, T)$, where $\mathcal{K}$ is a Kripke structure and – analogously to the first-order case – $T \subseteq W$ is called *team* (in $\mathcal{K}$). The team $RT := \{v \in W \mid \exists w \in T \colon Rwv\}$ is the *image* of T, and we write Rw instead of $R\{w\}$ for brevity. A successor team of T is a team S such that every $w \in T$ has at least one successor in S, and every $v \in S$ has at least one predecessor in T. The semantics of MTL is now defined as follows:

$$\begin{aligned}
(\mathcal{K}, T) &\vDash \varphi && \Leftrightarrow \forall w \in T \colon (\mathcal{K}, w) \vDash \varphi \text{ (in Kripke semantics) for } \varphi \in \mathsf{ML},\\
(\mathcal{K}, T) &\vDash {\sim}\psi && \Leftrightarrow (\mathcal{K}, T) \nvDash \psi,\\
(\mathcal{K}, T) &\vDash \psi \wedge \theta && \Leftrightarrow (\mathcal{K}, T) \vDash \psi \text{ and } (\mathcal{K}, T) \vDash \theta,\\
(\mathcal{K}, T) &\vDash \psi \vee \theta && \Leftrightarrow \exists S, U \subseteq T \text{ such that } T = S \cup U,\ (\mathcal{K}, S) \vDash \psi, \text{ and } (\mathcal{K}, U) \vDash \theta,\\
(\mathcal{K}, T) &\vDash \Diamond\psi && \Leftrightarrow \exists S \text{ such that } S \text{ is a successor team of } T \text{ and } (\mathcal{K}, S) \vDash \psi,\\
(\mathcal{K}, T) &\vDash \Box\psi && \Leftrightarrow (\mathcal{K}, RT) \vDash \psi.
\end{aligned}$$

A formula $\varphi \in \mathsf{MTL}$ is satisfiable if $(\mathcal{K}, T) \vDash \varphi$ for some Kripke structure $\mathcal{K}$ over $X \supseteq \mathsf{Prop}(\varphi)$ and team T in $\mathcal{K}$. Likewise, φ is valid if it is true in every such pair.

The modality-free fragment MTL_0 syntactically coincides with *propositional team logic* PTL [20, 21, 40]. The usual interpretations of the latter, i.e., via sets of Boolean assignments, can easily be simulated by teams in Kripke structures. For this reason, we identify PTL and MTL_0 in this paper.

The following lower bounds due to Hannula et al. [21] are logspace reductions.

▶ **Theorem 5.1** ([21])**.** $\mathsf{MC}(\mathsf{PTL})$ *is* PSPACE*-complete.*

▶ **Theorem 5.2** ([21]). $\mathsf{SAT}(\mathsf{PTL})$ *and* $\mathsf{VAL}(\mathsf{PTL})$ *are* ATIME-ALT(exp, poly)*-complete.*

For each increment in modal depth, the complexity of the satisfiability problem increases by an exponential, reaching the non-elementary class TOWER(poly) in the unbounded case:

▶ **Theorem 5.3** ([28]). $\mathsf{SAT}(\mathsf{MTL})$ *and* $\mathsf{VAL}(\mathsf{MTL})$ *are* TOWER(poly)*-complete.* $\mathsf{SAT}(\mathsf{MTL}_k)$ *and* $\mathsf{VAL}(\mathsf{MTL}_k)$ *are* ATIME-ALT($\exp_{k+1}$, poly)*-complete for every* $k < \omega$*.*

Next, let us demonstrate how MTL can be embedded into $\mathsf{FO}^2(\sim)$. More precisely, we present an extension of the well-known standard translation that embeds modal logic ML into FO^2. The underlying relational vocabulary usually is $\tau_{\mathrm{st}} = (R, P_1, P_2, \ldots)$, where $\mathsf{arity}(R) = 2$ and $\mathsf{arity}(P_i) = 1$ for all i. The translation of an ML-formula α is denoted by $\mathsf{st}_x(\alpha)$ resp. $\mathsf{st}_y(\alpha)$, and is defined by mutual recursion:

$$
\begin{aligned}
\mathsf{st}_x(p_i) &:= P_i x \quad \text{for } p_i \in \mathcal{PS} & \mathsf{st}_x(\neg\alpha) &:= \neg\mathsf{st}_x(\alpha)\\
\mathsf{st}_x(\Box\alpha) &:= \forall y\, Rxy \to \mathsf{st}_y(\alpha) & \mathsf{st}_x(\alpha \wedge \beta) &:= \mathsf{st}_x(\alpha) \wedge \mathsf{st}_x(\beta)\\
\mathsf{st}_x(\Diamond\alpha) &:= \exists y\, Rxy \wedge \mathsf{st}_y(\alpha) & \mathsf{st}_x(\alpha \vee \beta) &:= \mathsf{st}_x(\alpha) \vee \mathsf{st}_x(\beta),
\end{aligned}
$$

with $\mathsf{st}_y(\alpha)$ defined symmetrically via $\mathsf{st}_x(\alpha)$. The corresponding *first-order interpretation* of a Kripke structure $\mathcal{K} = (W, R', V)$ is the τ_{st}-structure $\mathcal{A}(\mathcal{K})$ defined by $\mathsf{dom}\,\mathcal{A}(\mathcal{K}) = W$, $R^{\mathcal{A}(\mathcal{K})} = R'$ and $P_i^{\mathcal{A}(\mathcal{K})} = V(p_i)$. For a world w, let $w^x \colon \{x\} \to W$ be defined by $w^x(x) = w$.

▶ **Theorem 5.4** (see, e.g., Blackburn et al. [5]). *Let* $(\mathcal{K}, w)$ *be a pointed Kripke structure and* $\alpha \in \mathsf{ML}$*. Then* $(\mathcal{K}, w) \vDash \alpha$ *if and only if* $(\mathcal{A}(\mathcal{K}), w^x) \vDash \mathsf{st}_x(\alpha)$*.*

Let us now turn to team semantics. On the model side, the first-order interpretation of a team T in a Kripke structure is straightforwardly $T^x := \{\, w^x \mid w \in T \,\}$. For the syntax, we require the additional operator $\hookrightarrow$. It was introduced by Galliani [12] and Kontinen and Nurmi [25] in the first-order setting, but was also adapted to the modal setting [28]. For $\alpha \in \mathsf{ML}$ and $\varphi \in \mathsf{MTL}$, define $\alpha \hookrightarrow \varphi := \neg\alpha \vee (\alpha \wedge \varphi)$. If $(\mathcal{K}, T)$ is a Kripke structure with team, let $T_\alpha := \{\, w \in T \mid (\mathcal{K}, w) \vDash \alpha \,\}$.

▶ **Proposition 5.5.** $(\mathcal{A}, T) \vDash \alpha \hookrightarrow \varphi$ *if and only if* $(\mathcal{A}, T_\alpha) \vDash \varphi$*.*

Proof. Straightforward. See also Galliani [12, Lemma 16]. ◀

We extend the above translation by an $\sim$-case, and in the $\Box$-case replace $\to$ by $\hookrightarrow$.[2] The standard translation for MTL, denoted by st^*_x resp. st^*_y, then becomes:

$$
\begin{aligned}
\mathsf{st}^*_x(\alpha) &:= \mathsf{st}_x(\alpha) \quad \text{for } \alpha \in \mathsf{ML} & \mathsf{st}^*_x(\sim\varphi) &:= \sim\mathsf{st}^*_x(\varphi)\\
\mathsf{st}^*_x(\Box\varphi) &:= \forall y\, Rxy \hookrightarrow \mathsf{st}^*_y(\varphi) & \mathsf{st}^*_x(\varphi \wedge \psi) &:= \mathsf{st}^*_x(\varphi) \wedge \mathsf{st}^*_x(\psi)\\
\mathsf{st}^*_x(\Diamond\varphi) &:= \exists y\, Rxy \wedge \mathsf{st}^*_y(\varphi) & \mathsf{st}^*_x(\varphi \vee \psi) &:= \mathsf{st}^*_x(\varphi) \vee \mathsf{st}^*_x(\psi),
\end{aligned}
$$

with $\mathsf{st}^*_y(\varphi)$ again defined symmetrically.

▶ **Theorem 5.6.** *For every Kripke structure* $\mathcal{K}$*, team* T *in* $\mathcal{K}$ *and* $\varphi \in \mathsf{MTL}$ *it holds* $(\mathcal{K}, T) \vDash \varphi$ *if and only if* $(\mathcal{A}(\mathcal{K}), T^x) \vDash \mathsf{st}^*_x(\varphi)$*.*

Proof. Proof by induction on φ. We omit $\mathcal{K}$ and $\mathcal{A}(\mathcal{K})$ and simply write, e.g., $T \vDash \varphi$.

---

[2] It is not hard to show that the "classical" translation of $\Box\varphi$ to $\forall y\, Rxy \to \mathsf{st}^*_y(\varphi) = \forall y\, (\neg Rxy \vee \mathsf{st}^*_y(\varphi))$ is unsound under team semantics.

- $\varphi \in \mathsf{ML}$: We have $T \vDash \varphi$ iff $\forall w \in T \colon w \vDash \varphi$ by definition of the semantics of MTL, which by Theorem 5.4 is equivalent to $\forall w^x \in T^x \colon w^x \vDash \mathsf{st}_x(\varphi)$. However, as $\mathsf{st}_x(\varphi) \in \mathsf{FO}$, the latter is equivalent to $T^x \vDash \mathsf{st}_x(\varphi)$ by the semantics of $\mathsf{FO}(\sim)$, and hence $T^x \vDash \mathsf{st}^*_x(\varphi)$.
- $\varphi = \psi \wedge \vartheta$ and $\varphi = {\sim}\psi$ are clear.
- $\varphi = \psi \vee \theta$: Suppose $T \vDash \psi \vee \theta$. Then $T = S \cup U$ such that $S \vDash \psi$ and $U \vDash \theta$. By induction hypothesis, $S^x \vDash \mathsf{st}^*_x(\psi)$ and $U^x \vDash \mathsf{st}^*_x(\theta)$. As $S \cup U = T$, clearly $S^x \cup U^x = T^x$. As a consequence, $T^x \vDash \mathsf{st}^*_x(\psi) \vee \mathsf{st}^*_x(\theta) = \mathsf{st}^*_x(\psi \vee \theta)$.

  For the other direction, suppose $T^x \vDash \mathsf{st}^*_x(\psi \vee \theta) = \mathsf{st}^*_x(\psi) \vee \mathsf{st}^*_x(\theta)$ by the means of some subteams $S' \cup U' = T^x$ such that $S' \vDash \mathsf{st}^*_x(\psi)$ and $U' \vDash \mathsf{st}^*_x(\theta)$. As T^x has domain $\{x\}$, there are unique $S, U \subseteq T$ such that $S' = S^x$ and $U' = U^x$. By induction hypothesis, $S \vDash \psi$ and $U \vDash \theta$. In order to prove $T \vDash \psi \vee \theta$, it remains to show $T \subseteq S \cup U$. For this purpose, let $w \in T$. As then $w^x \in T^x$, as least one of $w^x \in S'$ or $w^x \in U'$ holds. But then $w \in S$ or $w \in U$.
- $\varphi = \Box\psi$: We define subteams S and U of the duplicating team $(T^x)^y_W$ as follows: S contains all "outgoing edges": $S := \{s \in (T^x)^y_W \mid s(y) \in Rs(x)\}$. On the other hand, U contains all "non-edges": $U := \{s \in (T^x)^y_W \mid s(y) \notin Rs(x)\}$. Then clearly $(T^x)^y_W = S \cup U$, $S \vDash Rxy$ and $U \vDash \neg Rxy$. Moreover, the above division of $(T^x)^y_W$ into S and U is the *only* possible splitting of $(T^x)^y_W$ such that $S \vDash Rxy$ and $U \vDash \neg Rxy$.
  By induction hypothesis, clearly $T \vDash \Box\psi \Leftrightarrow (RT)^y \vDash \mathsf{st}^*_y(\psi)$. Moreover, by the above argument, $T^x \vDash \mathsf{st}^*_x(\Box\psi) \Leftrightarrow S \vDash \mathsf{st}^*_y(\psi)$. Consequently, it suffices to show that $(RT)^y$ and S agree on $\mathsf{st}^*_y(\psi)$. This follows from Proposition 2.5, since

$$\begin{aligned}(RT)^y &= \{s \colon \{y\} \to W \mid \exists w \in T : s(y) \in Rw\}\\ &= \{s{\upharpoonright}y \mid s \in (T^x)^y_W \text{ and } s(y) \in Rs(x)\} = S{\upharpoonright}y.\end{aligned}$$

- $\varphi = \Diamond\psi$: Suppose $T \vDash \Diamond\psi$, i.e., $S \vDash \psi$ for some successor team S of T. By induction hypothesis, $S^y \vDash \mathsf{st}^*_y(\psi)$. In order to prove $T^x \vDash \exists y\, Rxy \wedge \mathsf{st}^*_y(\psi)$, we define a supplementing function $f : T^x \to \mathfrak{P}(W) \setminus \{\emptyset\}$ such that $(T^x)^y_f \vDash Rxy \wedge \mathsf{st}^*_y(\psi)$.
  Let $f(w^x) := Rw \cap S$. Then $f(w^x)$ is non-empty for each w, as S is a successor team. Moreover, $(T^x)^y_f \vDash Rxy$. It remains to show that $(T^x)^y_f \vDash \mathsf{st}^*_y(\psi)$ follows from $S^y \vDash \mathsf{st}^*_y(\psi)$. Here, we combine Proposition A.1 and 2.5, since

$$y\langle S^y\rangle = S = \bigcup_{w \in T} Rw \cap S = \bigcup_{w^x \in T^x} f(w^x) = \{s(y) \mid s \in (T^x)^y_f\} = y\langle (T^x)^y_f\rangle.$$

  For the other direction, suppose $T^x \vDash \exists y\, Rxy \wedge \mathsf{st}^*_y(\psi)$ by the means of a supplementing function $f : T^x \to \mathfrak{P}(W) \setminus \{\emptyset\}$ such that $(T^x)^y_f \vDash Rxy \wedge \mathsf{st}^*_y(\psi)$.
  We define $S := \bigcup_{w \in T} f(w^x)$, and first prove that it is a successor team of T, i.e., that every $v \in S$ has a predecessor in T and that every $w \in T$ has a successor in S.
  Let $v \in S$. Then there exists $w \in T$ such that $v \in f(w^x)$. As a consequence, the assignment s given by $s(x) = w$ and $s(y) = v$ is in $(T^x)^y_f$, and hence satisfies Rxy. In other words, v has a predecessor in T. Conversely, if $w \in T$, then $f(w^x)$ is non-empty, i.e., contains an element v. As before, v is a successor of w. Since $v \in f(w^x)$, $v \in S$, so w has a successor in S. By a similar argument as above, $y\langle (T^x)^y_f\rangle = S = y\langle S^y\rangle$, hence $S^y \vDash \mathsf{st}^*_y(\psi)$, and consequently $S \vDash \psi$ by induction hypothesis. ◀

6 Lower bounds

As a first application of the extended standard translation from the previous section, we prove several complexity theoretic lower bounds.

▶ **Lemma 6.1.** $\mathsf{MC}(\tau\text{-}\mathsf{FO}^1_0(\sim))$ *is* PSPACE*-hard if* τ *contains infinitely many predicates.*

Proof. We reduce from $\mathsf{MC}(\mathsf{PTL})$, which is PSPACE-hard by Theorem 5.1. The reduction maps $(\mathcal{K}, T, \varphi)$ to $(\mathcal{A}(\mathcal{K}), T^x, \mathsf{st}^*_x(\varphi))$. W.l.o.g. τ contains unary predicates $P_0, P_1, \ldots$; otherwise they are easily simulated by predicates of higher arity. It is now easy to see that $\mathsf{st}^*_x(\varphi)$ is quantifier-free and contains only the variable x. Moreover, by Theorem 5.6, $(\mathcal{K}, T) \vDash \varphi$ if and only if $(\mathcal{A}(\mathcal{K}), T^x) \vDash \mathsf{st}^*_x(\varphi)$. ◀

▶ **Lemma 6.2.** $\mathsf{MC}(\tau\text{-}\mathsf{FO}^\omega_\omega(\sim))$ *is* ATIME-ALT(exp, poly)*-hard for all vocabularies* τ*, even on sentences and for a fixed* τ*-structure* $\mathcal{A}$ *with domain* $\{0,1\}$ *and a fixed team* $\{\emptyset\}$*.*

Proof. Here, we reduce from $\mathsf{SAT}(\mathsf{PTL})$, which is ATIME-ALT(exp, poly)-hard by Theorem 5.2. Given $\varphi \in \mathsf{PTL}$, suppose $\mathsf{Prop}(\varphi) = \{p_1, \ldots, p_n\}$. The idea is that a team of worlds (and their Boolean assignments to $p_1, \ldots, p_n$), are simulated by a team of first-order assignments $s \colon X \to B$, where $X = \{z, x_1, \ldots, x_n\}$ and $B := \{0, 1\}$. Here, the variable z acts as the constant 1, while x_i simulates p_i. For each $b \in B$, define the team $V_b := (\{\emptyset\}^z_{\{b\}})^{x_1}_B \cdots ^{x_n}_B$. In other words, V_b is the n-fold supplemented team of $\{\emptyset\}^z_{\{b\}} = \{\{z \mapsto b\}\}$.

In the remaining proof, we distinguish two cases based on τ. By definition of a vocabulary, either $= \in \tau$, or τ contains a predicate. First, we consider the case $= \in \tau$. We reduce via the mapping $\varphi \mapsto (\mathcal{A}, \{\emptyset\}, \psi)$, where $\mathcal{A}$ is a fixed τ-structure with $\mathsf{dom}\,\mathcal{A} = B$, $\psi := \exists z\, \forall x_1 \cdots \forall x_n\, \top \vee \varphi^*$, and φ^* is obtained from φ by replacing each p_i by $x_i = z$. We prove that the reduction is correct, and begin with the following equivalence:

$$\exists U \subseteq V_1 : (\mathcal{A}, U) \vDash \varphi^* \quad \text{iff} \quad (\mathcal{A}, V_1) \vDash \top \vee \varphi^* \quad \text{iff} \quad (\mathcal{A}, \{\emptyset\}) \vDash \psi. \tag{1}$$

Here, "⇒" follows from the semantics of $\vee$ and the definition of ψ. For "⇐", suppose $(\mathcal{A}, \{\emptyset\}) \vDash \psi$. Then, again by definition of ψ, we have $(\mathcal{A}, U) \vDash \varphi^*$ for some $U \subseteq V_0 \cup V_1$. In particular, the variable z can take the values 0, 1 or both in U. However, for all $s \in U \cap V_0$, we can simply flip the ones and zeroes of s. This leaves the truth of any atomic formula $x_i = z$ unchanged, and by induction preserves the semantics of φ^*.

Next, we proceed with the correctness of the reduction. Assume that φ is satisfiable, i.e., has a model $(\mathcal{K}, T)$. For each world $w \in T$, define $s_w \colon X \to B$ by $s_w(z) = 1$ and $s_w(x_i) = 1 \Leftrightarrow (\mathcal{K}, w) \vDash p_i$. Then $(\mathcal{K}, w) \vDash p_i$ if and only if $(\mathcal{A}, s_w) \vDash x_i = z$. By induction on the syntax of φ, we obtain $(\mathcal{A}, U) \vDash \varphi^*$, where $U := \{s_w \mid w \in T\}$. As $U \subseteq V_1$, the equivalence (1) yields $(\mathcal{A}, \{\emptyset\}) \vDash \psi$. The other direction is similar.

Next, consider the case where $= \notin \tau$; then τ contains a predicate P. We define $\mathcal{A}$ as above, but let $P^{\mathcal{A}} := \{(1, \ldots, 1)\}$. Furthermore, $\psi := \forall x_1 \cdots \forall x_n\, \top \vee \varphi^*$, and φ^* is now as φ, with p_i replaced by $P(x_i, \ldots, x_i)$. The remaining proof is similar to the previous one. ◀

Clearly, the standard translation of satisfiable formulas is itself satisfiable. A converse result holds as well. Loosely speaking, from a first-order structure (and team) for $\mathsf{st}^*_x(\varphi)$ we can reconstruct a Kripke model (and team) for φ.

▶ **Lemma 6.3.** *If* $\varphi \in \mathsf{MTL}$*, then* φ *is satisfiable if and only if* $\mathsf{st}^*_x(\varphi)$ *is satisfiable.*

Proof. As Theorem 5.6 implies "⇒", we show "⇐". Suppose $(\mathcal{B}, S) \vDash \mathsf{st}^*_x(\varphi)$. Then $\mathcal{B}$ interprets the binary predicate R and unary predicates $P_1, P_2, \ldots$. By Proposition 2.5, w.l.o.g. S has domain $\{x\}$, i.e., $S = (x\langle S\rangle)^x$. Define now the Kripke structure $\mathcal{K} = (\mathsf{dom}\,\mathcal{B}, R^{\mathcal{B}}, V)$ such that $V(p_i) := P_i^{\mathcal{B}}$. Then clearly $\mathcal{A}(\mathcal{K}) = \mathcal{B}$. By Theorem 5.6, $(\mathcal{K}, x\langle S\rangle) \vDash \varphi$. ◀

Finally, with the above lower bounds, let us gather the completeness results for the satisfiability, validity and model checking problems.

▶ **Theorem 6.4.** *Let $\mathcal{D}$ be any p-uniform set of dependencies and τ any vocabulary.*

- $\mathsf{MC}(\tau\text{-}\mathsf{FO}^{\omega}_{\omega}(\sim,\mathcal{D}))$ *is* ATIME-ALT(exp, poly)*-complete, with hardness already on sentences and for a fixed τ-structure $\mathcal{A}$ with domain $\{0,1\}$ and a fixed team $\{\emptyset\}$.*
- *If τ contains infinitely many relations and at least one of $k \geq 0, n \geq 1$ is finite, then* $\mathsf{MC}(\tau\text{-}\mathsf{FO}^{n}_{k}(\sim,\mathcal{D}))$ *is* PSPACE*-complete.*

Proof. The upper bounds are due to Corollary 3.3 and 3.6, since alternating polynomial time coincides with PSPACE [7]. The lower bounds are due to Lemma 6.1 and 6.2. ◀

▶ **Corollary 6.5.** $\mathsf{MC}(\tau\text{-}\mathsf{SO})$ *is* ATIME-ALT(exp, poly)*-complete for all vocabularies τ, with hardness already on sentences and with a fixed τ-interpretation $\mathcal{A}$ with* $\mathsf{dom}\,\mathcal{A} = \{0,1\}$*.*

Proof. The upper bound is by Proposition 2.6. The lower bound is by the previous theorem and reduction from $\mathsf{MC}(\tau\text{-}\mathsf{FO}(\sim))$. Let R be a nullary predicate variable. In the spirit of Corollary 3.3, we map $(\mathcal{A}, \{\emptyset\}, \varphi)$ to $(\mathcal{A}, \emptyset, \exists R\, \eta^{\emptyset}_{\varphi}(R) \wedge R)$, where φ w.l.o.g. is a sentence. ◀

The next theorem settles the complexity of the satisfiability and validity problem of $\mathsf{FO}^2(\sim)$, and provides lower bounds for $\mathsf{FO}^1_0(\sim)$ and $\mathsf{FO}^2_k(\sim)$.

▶ **Theorem 6.6.** *Let τ contain at least one binary predicate, infinitely many unary predicates, and no functions. Then the problems* $\mathsf{SAT}(\tau\text{-}\mathsf{FO}^n_k(\sim))$ *and* $\mathsf{VAL}(\tau\text{-}\mathsf{FO}^n_k(\sim))$ *are*

- TOWER(poly)*-complete for $n = 2$ and $k = \omega$,*
- ATIME-ALT($\exp_{k+1}$, poly)*-hard for $n = 2$ and $0 \leq k < \omega$,*
- ATIME-ALT(exp, poly)*-hard for $n = 1$ and $0 \leq k < \omega$.*

Proof. The upper bound for $\tau\text{-}\mathsf{FO}^2(\sim)$ is by Corollary 4.3. For the lower bounds, the mapping $\varphi \mapsto \mathsf{st}^*_x(\varphi)$ is a reduction from $\mathsf{SAT}(\mathsf{MTL})$ resp. $\mathsf{SAT}(\mathsf{MTL}_k)$ (see Theorem 5.3 and Lemma 6.3). Finally, the validity cases follow since the logic is closed under negation. ◀

Let us contrast the above decidable cases with the following negative result, where a single unary dependence atom is added to the logic (cf. p. 4).

▶ **Theorem 6.7.** *There is a vocabulary τ such that* $\mathsf{SAT}(\mathcal{L})$ *is* Π^0_1*-hard and* $\mathsf{VAL}(\mathcal{L})$ *is* Σ^0_1*-hard, where* $\mathcal{L} = \tau\text{-}\mathsf{FO}^2_2(\sim, \{\mathsf{dep}_1\})$.

Proof. Kontinen et al. [24] showed that $\mathsf{VAL}(\mathsf{D}^2_2)$ is Σ^0_1-hard, and their reduction in fact uses only unary and binary dependence atoms. Moreover, the binary dependence atom $=\!(x,y)$ can equivalently be rewritten as $\sim(\top \vee (=\!(x) \wedge \sim=\!(y)))$, where $\top$ is an arbitrary tautology. Intuitively, this formula stipulates that every subteam constant in x is also constant in y. This concludes the reduction to $\mathsf{VAL}(\tau\text{-}\mathsf{FO}^2_2(\sim, \{\mathsf{dep}_1\}))$. Again, the proof for the satisfiability problem is analogous. ◀

7 Conclusion

In this paper, we proved that the logic $\mathsf{FO}^2(\sim)$ is complete for the class TOWER(poly) and hence decidable. In particular, it has the finite model property, but exhibits non-elementary succinctness compared to classical FO^2, which enjoys an exponential model property [19].

For $\mathsf{FO}^n_k(\sim,\mathcal{D})$, where $n \geq 1$ and $k \geq 0$, we proved a dichotomy regarding its model checking complexity: It is ATIME-ALT(exp, poly)-complete if $n = k = \omega$, and otherwise PSPACE-complete. This only requires that $\mathcal{D}$ is a p-uniformly FO-definable set of generalized dependency atoms (cf. Definition 3.2), which covers first-order team logic TL as well as independence [17] and inclusion logic [10] augmented with Boolean negation.

We conclude with some open questions:

- Can the translation from $\mathsf{FO}^n_k(\sim,\mathcal{D})$ to $\mathsf{SO}[p]$ be inverted, i.e., can we translate every $\mathsf{SO}[p]$-formula to $\mathsf{FO}^n_k(\sim,\mathcal{D})$ for suitable n and k? This would be an interesting generalization of the translation from SO to TL given by Kontinen and Nurmi [25].
- What is the exact complexity of $\mathsf{SAT}(\mathsf{FO}^2_k(\sim))$? In the modal setting, every satisfiable MTL_k-formula has a $(k+1)$-fold exponential model. It would be interesting to learn whether the same holds for $\mathsf{FO}^2_k(\sim)$. Due to Corollary 3.6, a positive answer would immediately yield a tight ATIME-ALT$(\exp_{k+1}, \text{poly})$ upper bound.
- It is a well-known fact that the standard translation of an ML-formula is in the two-variable guarded fragment GF^2. It is conceivable to consider a similar fragment $\mathsf{GF}^2(\sim)$ for the standard translation of MTL. Studying the corresponding fragments $\mathsf{GF}^2_k(\sim)$ of bounded quantifier rank could also be a first step towards finding the complexity of $\mathsf{FO}^2_k(\sim)$.

References

1 Samson Abramsky, Juha Kontinen, Jouko Väänänen, and Heribert Vollmer, editors. *Dependence Logic, Theory and Applications.* Springer, 2016. doi:10.1007/978-3-319-31803-5.

2 Hajnal Andréka, István Németi, and Johan van Benthem. Modal languages and bounded fragments of predicate logic. *J. Philosophical Logic*, 27(3):217–274, 1998. doi:10.1023/A:1004275029985.

3 Vince Bárány and Mikołaj Bojańczyk. Finite satisfiability for guarded fixpoint logic. *Inf. Process. Lett.*, 112(10):371–375, 2012. doi:10.1016/j.ipl.2012.02.005.

4 Leonard Berman. The complexitiy of logical theories. *Theor. Comput. Sci.*, 11:71–77, 1980. doi:10.1016/0304-3975(80)90037-7.

5 Patrick Blackburn, Maarten de Rijke, and Yde Venema. *Modal logic.* Cambridge University Press, New York, NY, USA, 2001.

6 Egon Börger, Erich Grädel, and Yuri Gurevich. *The Classical Decision Problem.* Perspectives in Mathematical Logic. Springer, 1997.

7 Ashok K. Chandra, Dexter C. Kozen, and Larry J. Stockmeyer. Alternation. *J. ACM*, 28(1):114–133, 1981. doi:10.1145/322234.322243.

8 Arnaud Durand, Juha Kontinen, and Heribert Vollmer. Expressivity and Complexity of Dependence Logic. In Samson Abramsky, Juha Kontinen, Jouko Väänänen, and Heribert Vollmer, editors, *Dependence Logic*, pages 5–32. Springer International Publishing, 2016. doi:10.1007/978-3-319-31803-5_2.

9 Johannes Ebbing, Lauri Hella, Arne Meier, Julian-Steffen Müller, Jonni Virtema, and Heribert Vollmer. Extended modal dependence logic. In *Logic, Language, Information, and Computation - 20th International Workshop, WoLLIC 2013.*, pages 126–137, 2013. doi:10.1007/978-3-642-39992-3_13.

10 Pietro Galliani. Inclusion and exclusion dependencies in team semantics - on some logics of imperfect information. *Ann. Pure Appl. Logic*, 163(1):68–84, 2012. doi:10.1016/j.apal.2011.08.005.

11 Pietro Galliani. General Models and Entailment Semantics for Independence Logic. *Notre Dame Journal of Formal Logic*, 54(2):253–275, 2013. doi:10.1215/00294527-1960506.

12 Pietro Galliani. Upwards closed dependencies in team semantics. *Inf. Comput.*, 245:124–135, 2015. doi:10.1016/j.ic.2015.06.008.

13 Pietro Galliani. On strongly first-order dependencies. In *Dependence Logic, Theory and Applications*, pages 53–71. Springer, 2016. doi:10.1007/978-3-319-31803-5_4.

14 Erich Grädel. Model-checking games for logics of imperfect information. *Theor. Comput. Sci.*, 493:2–14, 2013. `doi:10.1016/j.tcs.2012.10.033`.

15 Erich Grädel, Phokion G. Kolaitis, Leonid Libkin, Maarten Marx, Joel Spencer, Moshe Y. Vardi, Yde Venema, and Scott Weinstein. *Finite Model Theory and Its Applications*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2007. `doi:10.1007/3-540-68804-8`.

16 Erich Grädel, Martin Otto, and Eric Rosen. Two-variable logic with counting is decidable. In *Proceedings, 12th Annual IEEE Symposium on Logic in Computer Science, 1997*, pages 306–317, 1997. `doi:10.1109/LICS.1997.614957`.

17 Erich Grädel and Jouko Väänänen. Dependence and independence. *Studia Logica*, 101(2):399–410, 2013. `doi:10.1007/s11225-013-9479-2`.

18 Erich Grädel and Igor Walukiewicz. Guarded fixed point logic. In *14th Annual IEEE Symposium on Logic in Computer Science*, pages 45–54, 1999. `doi:10.1109/LICS.1999.782585`.

19 Erich Grädel, Phokion G. Kolaitis, and Moshe Y. Vardi. On the decision problem for two-variable first-order logic. *Bulletin of Symbolic Logic*, 3(1):53–69, 1997. `doi:10.2307/421196`.

20 Miika Hannula, Juha Kontinen, Martin Lück, and Jonni Virtema. On quantified propositional logics and the exponential time hierarchy. In *Proceedings of the Seventh International Symposium on Games, Automata, Logics and Formal Verification, GandALF 2016.*, pages 198–212, 2016. `doi:10.4204/EPTCS.226.14`.

21 Miika Hannula, Juha Kontinen, Jonni Virtema, and Heribert Vollmer. Complexity of Propositional Logics in Team Semantic. *ACM Transactions on Computational Logic*, 19(1):1–14, jan 2018. `doi:10.1145/3157054`.

22 Wilfrid Hodges. Compositional semantics for a language of imperfect information. *Logic Journal of IGPL*, 5(4):539–563, 1997. `doi:10.1093/jigpal/5.4.539`.

23 Juha Kontinen, Antti Kuusisto, Peter Lohmann, and Jonni Virtema. Complexity of two-variable dependence logic and IF-logic. *Information and Computation*, 239:237–253, 2014. `doi:10.1016/j.ic.2014.08.004`.

24 Juha Kontinen, Antti Kuusisto, and Jonni Virtema. Decidability of Predicate Logics with Team Semantics. In *41st International Symposium on Mathematical Foundations of Computer Science (MFCS 2016)*, volume 58 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 60:1–60:14, 2016. `doi:10.4230/LIPIcs.MFCS.2016.60`.

25 Juha Kontinen and Ville Nurmi. Team logic and second-order logic. *Fundam. Inform.*, 106(2-4):259–272, 2011. `doi:10.3233/FI-2011-386`.

26 Antti Kuusisto. A Double Team Semantics for Generalized Quantifiers. *Journal of Logic, Language and Information*, 24(2):149–191, 2015. `doi:10.1007/s10849-015-9217-4`.

27 Leopold Löwenheim. Über möglichkeiten im relativkalkül. *Mathematische Annalen*, 76:447–470, 1915. URL: `http://eudml.org/doc/158703`.

28 Martin Lück. Canonical Models and the Complexity of Modal Team Logic. Computer Science Logic (CSL) 2018. To appear.

29 Martin Lück. Axiomatizations of team logics. *Annals of Pure and Applied Logic*, 169(9):928–969, 2018. `doi:10.1016/j.apal.2018.04.010`.

30 Martin Lück. On the Complexity of Team Logic and its Two-Variable Fragment. *CoRR*, abs/1804.04968, 2018. URL: `https://arxiv.org/abs/1804.04968`.

31 Michael Mortimer. On languages with two variables. *Math. Log. Q.*, 21(1):135–140, 1975. `doi:10.1002/malq.19750210118`.

32 Julian-Steffen Müller. *Satisfiability and model checking in team based logics*. PhD thesis, University of Hanover, 2014. URL: `http://d-nb.info/1054741921`.

33 Ian Pratt-Hartmann. The two-variable fragment with counting revisited. In *Logic, Language, Information and Computation, 17th International Workshop, WoLLIC 2010.*, pages 42–54, 2010. doi:10.1007/978-3-642-13824-9_4.

34 Frank P. Ramsey. *On a Problem of Formal Logic*, pages 1–24. Birkhäuser Boston, Boston, MA, 1987. doi:10.1007/978-0-8176-4842-8_1.

35 Dana Scott. A decision method for validity of sentences in two variables. *Journal of Symbolic Logic*, 27(4):477, 1962.

36 Thomas Sturm, Marco Voigt, and Christoph Weidenbach. Deciding first-order satisfiability when universal and existential variables are separated. In *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2016*, pages 86–95, 2016. doi:10.1145/2933575.2934532.

37 Marco Voigt. A fine-grained hierarchy of hard problems in the separated fragment. In *32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2017*, pages 1–12, 2017. doi:10.1109/LICS.2017.8005094.

38 Jouko Väänänen. *Dependence logic: A New Approach to Independence Friendly Logic*. Number 70 in London Mathematical Society student texts. Cambridge University Press, Cambridge ; New York, 2007.

39 Fan Yang. *On extensions and variants of dependence logic.* PhD thesis, University of Helsinki, 2014. URL: http://www.math.helsinki.fi/logic/people/fan.yang/dissertation_fyang.pdf.

40 Fan Yang and Jouko Väänänen. Propositional team logics. *Ann. Pure Appl. Logic*, 168(7):1406–1441, 2017. doi:10.1016/j.apal.2017.01.007.

Algorithm 1: Decision procedure for MC(SO).

Algorithm: $\mathsf{check}(\alpha, \mathcal{A}, \mathcal{J})$ for $\alpha \in \tau$-SO in negation normal form, a τ-structure $\mathcal{A}$, and a second-order interpretation $\mathcal{J}$ of $\mathsf{Fr}(\alpha)$.

if *α is an atomic formula or the negation of an atomic formula* **then**
  return `true` *if* $(\mathcal{A}, \mathcal{J}) \vDash \alpha$ *and* `false` *otherwise*;
else if $\alpha = \gamma_1 \vee \gamma_2$ **then** existentially choose $i \in \{1, 2\}$ and let $\alpha := \gamma_i$
else if $\alpha = \gamma_1 \wedge \gamma_2$ **then** universally choose $i \in \{1, 2\}$ and let $\alpha := \gamma_i$
else if $\alpha = \eth X \gamma$ *for* $\eth \in \{\exists, \forall\}$ *and* $X \in \mathsf{Var}(\alpha)$ **then**
  $\alpha := \gamma$
  if $X \in \mathsf{Fr}(\gamma)$ **then**
    if $\eth = \exists$ **then** switch to existential branching **else** switch to universal branching
    if *X is a first-order variable* **then**
      non-deterministically choose $a \in \mathcal{A}$ and let $\mathcal{J}(X) := a$
    else if *X is a function variable* **then**
      non-deterministically choose $F \colon \mathcal{A}^{\mathsf{arity}(X)} \to \mathcal{A}$ and let $\mathcal{J}(X) := F$
    else if *X is a relation variable* **then**
      non-deterministically choose $R \subseteq \mathcal{A}^{\mathsf{arity}(X)}$ and let $\mathcal{J}(X) := R$
return $\mathsf{check}(\alpha, \mathcal{A}, \mathcal{J}{\restriction}\mathsf{Fr}(\alpha))$

A Appendix

Proof of Proposition 2.6

▶ **Proposition 2.6.** MC(SO) *is decidable on input* $(\mathcal{A}, \mathcal{J}, \alpha)$ *in time* $2^{n^{\mathcal{O}(1)}}$ *and with* $|\alpha|$ *alternations.*

Proof. W.l.o.g., $\neg$ appears in α only in front of atomic formulas, and $\mathsf{dom}\,\mathcal{J} = \mathsf{Fr}(\alpha)$. Let $A := \mathsf{dom}\,\mathcal{A}$. We abbreviate

$$|\mathcal{J}| := \sum_{\substack{X \in \mathsf{dom}\,\mathcal{J} \\ X \text{ second-order}}} |\mathcal{J}(X)|,$$

i.e., the sum of the cardinalities of functions and relations in $\mathcal{J}$. Since for any second-order variable X it holds $|\mathcal{J}(X)| \leq |A|^{\mathsf{arity}(X)} \leq |A|^{|\alpha|}$, and furthermore $|\mathsf{dom}\,\mathcal{J}| = |\mathsf{Fr}(\alpha)| \leq |\alpha|$, the sum $|\mathcal{J}|$ is at most $|\alpha| \cdot |A|^{|\alpha|}$.

Now we run Algorithm 1. It performs at most $|\alpha|$ recursive calls, and clearly at most $|\alpha|$ alternations. Furthermore, the i-th recursive call is of the form $\mathsf{check}(\alpha_i, \mathcal{A}, \mathcal{J}_i)$ with $|\alpha_i| \leq |\alpha|$ and, by the same argument as before, $|\mathcal{J}_i| \leq |\alpha| \cdot |A|^{|\alpha|}$. For this reason, it is easy to see that the overall runtime is polynomial in $|\mathcal{J}|$ and $|A|^{|\alpha|}$, and consequently exponential in the input size. ◀

Proofs of Theorem 3.1 and 3.5

We require the next propositions in order to prove Theorem 3.1 and 3.5.

▶ **Proposition A.1.** *Let $\mathcal{A}$ be a structure, $\vec{t}$ a tuple of terms, and $X \supseteq \mathsf{Fr}(\vec{t})$. For $i \in \{1,2\}$, let T_i be a team in $\mathcal{A}$ with domain $X_i \supseteq X$. Then $T_1 \restriction X = T_2 \restriction X$ implies $\vec{t}\langle T_1\rangle = \vec{t}\langle T_2\rangle$. Furthermore, for any tuple $\vec{x} \subseteq X$ of variables, $\vec{x}\langle T_1\rangle = \vec{x}\langle T_2\rangle$ iff $T_1 \restriction \vec{x} = T_2 \restriction \vec{x}$.*

Proof. For the first part of the proposition, assume $T_1 \restriction X = T_2 \restriction X$. Exploiting symmetry, we only show that $\vec{t}\langle T_1\rangle \subseteq \vec{t}\langle T_2\rangle$. Hence, let $\vec{a} \in \vec{t}\langle T_1\rangle$ be arbitrary. Then $\vec{a} = \vec{t}\langle s\rangle$ for some $s \in T_1$. By assumption, there is $s' \in T_2$ such that $s \restriction X = s' \restriction X$. Since $\mathsf{Fr}(\vec{t}) \subseteq X$, clearly $\vec{t}\langle s\rangle = \vec{t}\langle s'\rangle$. Consequently, $\vec{a} \in \vec{t}\langle T_2\rangle$.

For the second part, suppose $\vec{x}\langle T_1\rangle = \vec{x}\langle T_2\rangle$ and let $s \in T_1 \restriction \vec{x}$ be arbitrary. We show $s \in T_2 \restriction \vec{x}$, which again suffices due to symmetry. Clearly, $s = s' \restriction \vec{x}$ for some $s' \in T_1$. Then $\vec{x}\langle s\rangle = \vec{x}\langle s'\rangle \in \vec{x}\langle T_1\rangle = \vec{x}\langle T_2\rangle$, and consequently, $\vec{x}\langle s\rangle \in \vec{x}\langle T_2\rangle$. But then $\vec{x}\langle s\rangle = \vec{x}\langle s''\rangle$ for some $s'' \in T_2$, which implies $s = s'' \restriction \vec{x}$, and hence $s \in T_2 \restriction \vec{x}$. ◀

▶ **Proposition A.2.** *Let $\mathcal{A}$ be a structure, $\vec{x}$ a tuple of variables, and $V := \{s : \vec{x} \to \mathcal{A}\}$. Then $\mathfrak{P}(V)$ is the set of all teams in $\mathcal{A}$ with domain $\vec{x}$, and the mapping $r : S \mapsto \vec{x}\langle S\rangle$ is an order isomorphism between $(\mathfrak{P}(V), \subseteq)$ and $(\mathfrak{P}((\mathsf{dom}\,\mathcal{A})^{|\vec{x}|}), \subseteq)$.*

Proof. Let $n := |\vec{x}|$. Clearly, every team with domain $\vec{x}$ is in $\mathfrak{P}(V)$. It is easy to show that r is surjective: Given $A \subseteq (\mathsf{dom}\,\mathcal{A})^n$, define the team $S := \{s \in V \mid \vec{x}\langle s\rangle \in A\}$. Then $r(S) = \vec{x}\langle S\rangle = \{\vec{x}\langle s\rangle \mid s \in V \text{ and } \vec{x}\langle s\rangle \in A\} = A$.

Moreover, r preserves $\subseteq$ in both directions: Suppose $S \subseteq S'$ and let $\vec{a} = (a_1, \dots, a_n) \in r(S)$ be arbitrary. We show $\vec{a} \in r(S')$, which proves $r(S) \subseteq r(S')$. Since $\vec{a} \in r(S) = \vec{x}\langle S\rangle$, there exists $s \in S$ such that $\vec{x}\langle s\rangle = \vec{a}$. By assumption, $s \in S'$. Consequently, $\vec{a} \in \vec{x}\langle S'\rangle = r(S')$.

Conversely, suppose $r(S) \subseteq r(S')$ and let $s \in S$ be arbitrary. As $\vec{x}\langle s\rangle \in r(S) \subseteq r(S') = \vec{x}\langle S'\rangle$, there exists an assignment $s' \in S'$ such that $\vec{x}\langle s\rangle = \vec{x}\langle s'\rangle$. However, as $\mathsf{dom}\, s = \mathsf{dom}\, s' = \vec{x}$, necessarily $s = s'$, i.e., $s \in S'$. As $S \subseteq S' \Leftrightarrow r(S) \subseteq r(S')$, and r is surjective, we conclude that r is also injective and hence an order isomorphism. ◀

As an alternative definition of supplementing functions, Galliani [11] coined the term *x-variations*, which are teams that "agree" on all variables but x:

▶ **Proposition A.3.** *Let T be a team with domain X and S a team with domain $X \cup \{x\}$ (with possibly $x \in X$), and let $X' := X \setminus \{x\}$. Then $S \restriction X' = T \restriction X'$ if and only if there is a supplementing function f such that $S = T^x_f$.*

Proof. Let $\mathcal{A}$ be the underlying structure.

"$\Rightarrow$": Suppose $S \restriction X' = T \restriction X'$. First, we show that for every $s' \in S$ there is $s \in T$ such that $s' = s^x_a$ for some a. By assumption, $s' \restriction X' = s \restriction X'$ for some $s \in T$. But then $s' = s^x_{s'(x)}$. We define the function $f(s) := \{\, a \in \mathcal{A} \mid s^x_a \in S \,\}$, and prove that it is a supplementing function of T. Here, it suffices to show that $f(s) \neq \emptyset$ for all $s \in T$, i.e., that for every $s \in T$ there exists $a \in \mathcal{A}$ such that $s^x_a \in S$. This follows again by $S \restriction X' = T \restriction X'$. Moreover, $T^x_f = \{s^x_a \mid s \in T, a \in f(s)\} = \{s^x_a \mid s \in T, s^x_a \in S\}$, which equals S by the above argument.

"$\Leftarrow$": First, we show "$\subseteq$", i.e., that $s \in T \restriction X'$ for arbitrary $s \in S \restriction X'$. By definition, for such s we have $s = s' \restriction X'$ for some $s' \in S$. Since $S = T^x_f$, there exists $s'' \in T$ and $a \in \mathcal{A}$ such that $s' = (s'')^x_a$. As $x \notin X'$, we have $s = s' \restriction X' = s'' \restriction X' \in T \restriction X'$, as desired.

For the other direction, "$\supseteq$", let $s \in T \restriction X'$ be arbitrary. Then $s = s' \restriction X'$ for some $s' \in T$. As $S = T^x_f$, there exists some $s'' \in S$ and $a \in \mathcal{A}$ such that $s'' = (s')^x_a$. Again we have $s = s' \restriction X' = s'' \restriction X'$, i.e., $s \in S \restriction X'$. ◀

▶ **Lemma A.4.** *Let T have domain $\vec{x}$ and S have domain $\vec{x} \cup \{y\}$ (with possibly $y \in X$), and let $X' := \vec{x} \setminus \{y\}$. Then $T \restriction X' = S \restriction X'$ if and only if $\mathcal{A} \vDash \pi(\vec{x}\langle T\rangle, \vec{x};y\langle S\rangle)$, where $\pi(T, S) := \forall \vec{x}((\exists y T\vec{x}) \leftrightarrow (\exists y S \vec{x};y))$.*

Proof. First, let us consider the case $y \notin X$, i.e., $X' = X$. Then:

$$\begin{aligned} & T{\upharpoonright}X' = S{\upharpoonright}X' \\ \Leftrightarrow\ & \vec{x}\langle T\rangle = \vec{x}\langle S\rangle && \text{(by Proposition A.1)} \\ \Leftrightarrow\ & \forall \vec{a} : (\vec{a} \in \vec{x}\langle T\rangle \Leftrightarrow \exists b : (\vec{a}, b) \in (\vec{x}; y)\langle S\rangle) && \text{(since } T \text{ has domain } \vec{x}) \\ \Leftrightarrow\ & \mathcal{A} \vDash \pi(\vec{x}\langle T\rangle, \vec{x};y\langle S\rangle) && \text{(since } \exists y T\vec{x} \equiv T\vec{x}) \end{aligned}$$

Next, assume $y \in X$ and w.l.o.g. $y = x_n$. Then $\vec{x};y = \vec{x}$ and $X' = \{x_1, \ldots, x_{n-1}\}$. Let $\vec{x}' = (x_1, \ldots, x_{n-1})$. Analogously as before, we have:

$$\begin{aligned} & T{\upharpoonright}X' = S{\upharpoonright}X' \\ \Leftrightarrow\ & \vec{x}'\langle T\rangle = \vec{x}'\langle S\rangle \\ \Leftrightarrow\ & \forall \vec{a} : \big((\exists b : (\vec{a}, b) \in \vec{x}\langle T\rangle) \Leftrightarrow (\exists b : (\vec{a}, b) \in \vec{x};y\langle S\rangle)\big) \\ \Leftrightarrow\ & \mathcal{A} \vDash \pi(\vec{x}\langle T\rangle, \vec{x};y\langle S\rangle) \end{aligned}$$ ◀

▶ **Theorem 3.1.** *Let $\varphi \in \mathsf{FO}(\sim, \mathcal{D})$, let $\vec{x} \supseteq \mathsf{Fr}(\varphi)$ be a tuple of variables, and T be a team in $\mathcal{A}$ with domain $Y \supseteq \vec{x}$. Then $(\mathcal{A}, T) \vDash \varphi$ if and only if $\mathcal{A} \vDash \eta^{\vec{x}}_{\varphi}(\vec{x}\langle T\rangle)$.*

Proof. Note that $(\mathcal{A}, T) \vDash \varphi \Leftrightarrow (\mathcal{A}, T{\upharpoonright}\vec{x}) \vDash \varphi$ by Proposition 2.5, and $\vec{x}\langle T\rangle = \vec{x}\langle T{\upharpoonright}\vec{x}\rangle$. For this reason, we can assume that T has domain $\vec{x}$. The proof is now by induction on φ.

- If φ is first-order, clearly $(\mathcal{A}, T) \vDash \varphi$ iff $\mathcal{A} \vDash \varphi(\vec{a})$ for all $\vec{a} \in \vec{x}\langle T\rangle$ iff $\mathcal{A} \vDash \eta^{\vec{x}}_{\varphi}(\vec{x}\langle T\rangle)$.
- If $\varphi = A_i(\vec{t})$ and $\delta_i \in \mathcal{D}$ is a k-ary dependency, then $(\mathcal{A}, T) \vDash A_i(\vec{t})$ iff $\mathcal{A} \vDash \delta_i(\vec{t}\langle T\rangle)$. We prove that this is again equivalent to $\mathcal{A} \vDash \exists S\, \rho(\vec{x}\langle T\rangle, S) \land \delta_i(S)$, where $\rho(R, S) := \forall \vec{z}\, (S\vec{z} \leftrightarrow (\exists \vec{x}\, R\vec{x} \land \vec{t} = \vec{z}\,))$.
  It suffices to show that $\mathcal{A} \vDash \rho(\vec{x}\langle T\rangle, S)$ if and only if $S = \vec{t}\langle T\rangle$. As it is straightforward that $\mathcal{A} \vDash \rho(\vec{x}\langle T\rangle, \vec{t}\langle T\rangle)$ holds, let us focus on the direction from left to right. Recall that $\vec{x} \cap \{z_1, \ldots, z_k\} = \emptyset$ and that the z_i are pairwise distinct. On that account, suppose $\mathcal{A} \vDash \rho(\vec{x}\langle T\rangle, S)$ and $\vec{a} = (a_1, \ldots, a_k) \in \mathcal{A}^k$. By definition of the formula, $\vec{a} \in S$ iff $\mathcal{A} \vDash \exists \vec{x}\, R\vec{x} \land \vec{t} = \vec{a}$. However, this is the case iff $\vec{t}\langle s\rangle = \vec{a}$ for some $s \in T$, i.e., $\vec{a} \in \vec{t}\langle T\rangle$.
- The cases $\varphi = {\sim}\psi$ and $\varphi = \psi \land \theta$ immediately follow by induction hypothesis.
- If $\varphi = \psi \lor \theta$, then by induction hypothesis, $(\mathcal{A}, T) \vDash \varphi$ iff there are $S, U \subseteq T$ such that $T = S \cup U$ and $\mathcal{A} \vDash \eta^{\vec{x}}_{\psi}(\vec{x}\langle S\rangle) \land \eta^{\vec{x}}_{\theta}(\vec{x}\langle U\rangle)$. Let $R := \vec{x}\langle T\rangle$.
  Then due to Proposition A.2, the above is equivalent to the existence of $P, Q \subseteq \mathcal{A}^n$ such that $R = P \cup Q$ and $\mathcal{A} \vDash \eta^{\vec{x}}_{\psi}(P) \land \eta^{\vec{x}}_{\theta}(Q)$, and consequently to $\mathcal{A} \vDash \exists S\, \exists U\, \forall \vec{x} (R\vec{x} \leftrightarrow (S\vec{x} \lor U\vec{x})) \land \eta^{\vec{x}}_{\psi}(S) \land \eta^{\vec{x}}_{\theta}(U)$.
- If $\varphi = \exists y\, \psi$, by Proposition A.3, then $(\mathcal{A}, T) \vDash \varphi$ iff $(\mathcal{A}, S) \vDash \psi$ for some team S with domain $\vec{x} \cup \{y\}$ such that $T{\upharpoonright}X' = S{\upharpoonright}X'$, where $X' := \vec{x} \setminus \{y\}$. By Lemma A.4 and by induction hypothesis, this is the case iff $(\mathcal{A}, \vec{x}\langle T\rangle) \vDash \exists S\, \forall \vec{x} ((\exists y R\vec{x}) \leftrightarrow (\exists y S\vec{x}; y)) \land \eta^{\vec{x};y}_{\psi}(S)$.
- The case $\varphi = \forall y\, \psi$ is proven analogously to $\exists$. The additional clause $(R\vec{x} \to \forall y\, S\vec{x};y)$ ensures that the supplementing function is constant and $f(s) = \mathsf{dom}\, \mathcal{A}$. ◀

▶ **Theorem 3.5.** *Let $\varphi \in \mathsf{FO}(\sim, \mathcal{D})$, let $\vec{x} \supseteq \mathsf{Fr}(\varphi)$ be a tuple of variables, and T be a team in $\mathcal{A}$ with domain $Y \supseteq \vec{x}$. If $p(n) \geq |T| \cdot n^{\mathsf{qr}(\varphi)}$ or $p(n) \geq n^{\mathsf{w}(\varphi)}$, then $(\mathcal{A}, T) \vDash \varphi$ if and only if $\mathcal{A} \vDash \zeta^{\vec{x},p}_{\varphi}(\vec{x}\langle T\rangle)$.*

Proof for $p(n) \geq |T| \cdot n^{\mathsf{qr}(\varphi)}$. Assume $\mathcal{A}$, T as above, let $m := \mathsf{qr}(\varphi)$ and $p(n) \geq n^m$. The idea is to show that $\eta^{\vec{x}}_{\varphi}$ and $\zeta^{\vec{x},p}_{\varphi}$ agree on $(\mathcal{A}, \mathcal{J})$ for all "sufficiently sparse" $\mathcal{J}$ (cf. Theorem 3.1). Formally, let $\ell \leq m$ and let $(\mathcal{A}, \mathcal{J})$ be a second-order interpretation such that

$|\mathcal{J}(R)| \leq |T| \cdot |\mathcal{A}|^{\ell}$ for all relations $R \in \mathsf{dom}\,\mathcal{J}$. Then we prove for all $\varphi \in \mathsf{FO}(\sim, \mathcal{D})$ with $\mathsf{qr}(\varphi) \leq m - \ell$ and $\vec{x} \supseteq \mathsf{Fr}(\varphi)$ that $(\mathcal{A}, \mathcal{J}) \vDash \eta_{\varphi}^{\vec{x}}$ if and only if $(\mathcal{A}, \mathcal{J}) \vDash \zeta_{\varphi}^{\vec{x},p}$. For $\ell = 0$, this yields the theorem, since $|\vec{x}\langle T\rangle| \leq |T| \cdot |\mathcal{A}|^{0}$.

The proof is by induction on φ. We distinguish the following cases.

- If $\varphi \in \mathsf{FO}$, then there is nothing to prove, as $\eta_{\varphi}^{\vec{x}} = \zeta_{\varphi}^{\vec{x},p}$.
- If $\varphi = {\sim}\psi$ or $\varphi = \psi \wedge \theta$, then the inductive step is clear.
- If $\varphi = A_i\vec{t}$ for some k-ary $\delta_i \in \mathcal{D}$, then $\zeta_{\varphi}^{\vec{x},p}(R) = \exists^p S\, \rho(R, S)$ and $\eta_{\varphi}^{\vec{x}}(R) = \exists S\, \rho(R, S)$, where

  $$\rho(R, S) = \forall \vec{z}\, (S\vec{z} \leftrightarrow (\exists \vec{x}\, R\vec{x} \wedge \vec{t} = \vec{z}\,)) \wedge \delta_i(S).$$

  We show that $\mathcal{A} \vDash \eta_{\varphi}^{\vec{x}}(R)$ implies $\mathcal{A} \vDash \zeta^{\vec{x},p}(R)$, as the other direction is trivial.
  On that account, suppose $\mathcal{A} \vDash \rho(R, S)$ for some $S \subseteq \mathcal{A}^k$. We prove that necessarily $|S| \leq |R|$ by constructing some injective $f\colon S \to R$. Then $\mathcal{A} \vDash \exists^p S\, \rho(R, S)$, as by assumption, $|S| \leq |R| \leq |T| \cdot |\mathcal{A}|^{\ell} \leq |T| \cdot |\mathcal{A}|^{m} \leq p(|\mathcal{A}|)$.

  We define f as follows. For every $\vec{a} \in S$, let $f(\vec{a})$ be some $\vec{b} \in R$ such that $\vec{t}\langle\{\vec{x} \mapsto \vec{b}\}\rangle = \vec{a}$. By $\rho(R, S)$, such $\vec{b}$ must exist. Clearly, f is injective.
- If $\varphi = \psi \vee \theta$, then $\zeta_{\varphi}^{\vec{x},p}(R) = \exists^p S\, \exists^p U\, \rho$ and $\eta_{\varphi}^{\vec{x}}(R) = \exists S\, \exists U\, \rho'$, where

  $$\rho(R, S, U) = \forall \vec{x}(R\vec{x} \leftrightarrow (S\vec{x} \vee U\vec{x})) \wedge \zeta_{\psi}^{\vec{x},p}(S) \wedge \zeta_{\theta}^{\vec{x},p}(U),$$
  $$\rho'(R, S, U) = \forall \vec{x}(R\vec{x} \leftrightarrow (S\vec{x} \vee U\vec{x})) \wedge \eta_{\psi}^{\vec{x}}(S) \wedge \eta_{\theta}^{\vec{x}}(U).$$

  Suppose $|R| \leq |T| \cdot |\mathcal{A}|^{\ell}$ and $\mathsf{qr}(\varphi) \leq m - \ell$. Clearly $\mathsf{qr}(\psi), \mathsf{qr}(\theta) \leq \mathsf{qr}(\varphi)$.
  Let $\mathcal{A} \vDash \eta_{\varphi}^{\vec{x}}(R)$, i.e., $\mathcal{A} \vDash \rho'(R, S, U)$ for some $S, U \subseteq \mathcal{A}^{|\vec{x}|}$.
  It is easy to see that ρ' forces $|S|, |U| \leq |R|$. Since $|R| \leq |T| \cdot |\mathcal{A}|^{\ell}$ by assumption, we can apply the induction hypothesis to $\eta_{\psi}^{\vec{x}}(S)$ and $\eta_{\theta}^{\vec{x}}(U)$ and derive $\mathcal{A} \vDash \rho(R, S, U)$ from $\mathcal{A} \vDash \rho'(R, S, U)$. Since in particular $|S|, |U| \leq p(|\mathcal{A}|)$, we conclude $\mathcal{A} \vDash \zeta_{\varphi}^{\vec{x},p}(R)$. The other direction is trivial due to the inductivion hypothesis, since $\rho(R, S)$ entails $\rho'(R, S)$.
- If $\varphi = \exists y\, \psi$, then $\zeta_{\varphi}^{\vec{x},p}(R) = \exists^p S\, \rho(R, S)$ and $\eta_{\varphi}^{\vec{x}}(R) = \exists S\, \rho'(R, S)$, where

  $$\rho(R, S) = \forall \vec{x}((\exists y R\vec{x}) \leftrightarrow (\exists y\, S\vec{x};y)) \wedge \zeta_{\psi}^{\vec{x};y,p}(S),$$
  $$\rho'(R, S) = \forall \vec{x}((\exists y R\vec{x}) \leftrightarrow (\exists y\, S\vec{x};y)) \wedge \eta_{\psi}^{\vec{x};y}(S).$$

  Suppose $|R| \leq |T| \cdot |\mathcal{A}|^{\ell}$ and $\mathsf{qr}(\varphi) \leq m - \ell$. We show that $\mathcal{A} \vDash \eta_{\varphi}^{\vec{x}}(R)$ implies $\mathcal{A} \vDash \zeta_{\varphi}^{\vec{x},p}(R)$. The other direction is then again similar.
  Assuming $\mathcal{A} \vDash \eta_{\varphi}^{\vec{x}}(R)$, there exists $S \subseteq \mathcal{A}^{|\vec{x};y|}$ such that $\mathcal{A} \vDash \rho'(R, S)$. As a first step, we erase unnecessary elements from S. Note that S occurs in ρ' only in atomic formulas $S\vec{x};y$, i.e., with a fixed argument tuple $\vec{x};y$. Let $(v_1, \ldots, v_r) := \vec{x};y$. If now $v_i = v_j$ for some $1 \leq i < j \leq r$, then every tuple $(a_1, \ldots, a_r)$ with $a_i \neq a_j$ can be safely deleted from S. Formally, if $S^* := \vec{x};y\langle V\rangle \cap S$, where $V = \{s\colon \vec{x} \cup \{y\} \to \mathcal{A}\}$ is the full team with domain $\vec{x} \cup \{y\}$, then $\mathcal{A} \vDash \rho'(R, S)$ if and only if $\mathcal{A} \vDash \rho'(R, S^*)$, which can be shown by straightforward induction.

  Note that $\mathsf{qr}(\psi) = \mathsf{qr}(\varphi) - 1 \leq m - (\ell + 1)$. Consequently, to apply the induction hypothesis, we prove $|S^*| \leq |R| \cdot |\mathcal{A}| \leq |T| \cdot |\mathcal{A}|^{\ell+1}$ by presenting some injective $f\colon S^* \to R \times \mathcal{A}$.
  If $y \notin \vec{x}$, let f be the identity, as ρ' ensures that $(\vec{a}, b) \in S^*$ implies $\vec{a} \in R$. However, if $y \in \vec{x}$, then we define $f(\vec{a})$ as follows. By construction, $\vec{a} \in S^*$ equals $\vec{x}\langle s\rangle$ for some $s\colon \vec{x} \to \mathcal{A}$. Again by ρ', there is $\hat{s}\colon \vec{x} \to \mathcal{A}$ such that $\vec{x}\langle\hat{s}\rangle \in R$ and $s = \hat{s}_{s(y)}^{y}$. Let now $f(\vec{a}) := (\vec{x}\langle\hat{s}\rangle, s(y))$. Then f is injective.

Hence, by induction hypothesis, we can replace $\eta_\varphi^{\vec{x}}$ by $\zeta_\varphi^{\vec{x},p}$ and obtain $\mathcal{A} \vDash \rho(R, S^*)$. Since $|S^*| \leq |\mathcal{A}|^{\ell+1} \leq p(|\mathcal{A}|)$, we obtain $\mathcal{A} \vDash \exists^p S\, \rho(R, S)$.

- The case $\varphi = \forall y\, \psi$ is proven similarly to $\varphi = \exists y\, \psi$. ◀

Proof for $p(n) \geq n^{\mathsf{w}(\varphi)}$. We can apply the same argument as in the $\exists$-case of the previous proof. Suppose S is a second-order variable. Then S appears in $\eta_\varphi^{\vec{x}}$ only in atomic formulas of the form $S\vec{t}$ for a fixed $\vec{t}$. Accordingly, it suffices to let $\exists S$ range over subsets of $\vec{t}\langle V\rangle$, where $\vec{y} := \mathsf{Var}(\vec{t})$ and $V := \{s \colon \vec{y} \to \mathcal{A}\}$.

(We consider terms $\vec{t}$ instead of only variables to account for the translations of dependencies, where S can have terms as arguments.)

Since $\vec{y}$ contains at most $\mathsf{w}(\varphi)$ distinct variables, $|V| \leq |\mathcal{A}|^{\mathsf{w}(\varphi)} \leq p(|\mathcal{A}|)$. Consequently, every second-order quantifier $\exists S$ can be replaced by $\exists^p S$, which implies $\mathcal{A} \vDash \eta_\varphi^{\vec{x}}(\vec{x}\langle T\rangle) \Leftrightarrow \mathcal{A} \vDash \zeta_\varphi^{\vec{x},p}(\vec{x}\langle T\rangle)$. ◀

Proof of Lemma 4.1

▶ **Lemma A.5.** *The following laws hold for* $\mathsf{FO}(\sim)$*:*

$$\alpha \wedge \bigwedge_{i=1}^{n} \mathsf{E}\beta_i \equiv \bigvee_{i=1}^{n} (\alpha \wedge \mathsf{E}\beta_i) \tag{2}$$

$$\bigvee_{i=1}^{n} (\alpha_i \wedge \mathsf{E}\beta_i) \equiv \left(\bigvee_{i=1}^{n} \alpha_i\right) \wedge \bigwedge_{i=1}^{n} \mathsf{E}(\alpha_i \wedge \beta_i) \tag{3}$$

$$(\vartheta_1 \varovee \vartheta_2) \vee \vartheta_3 \equiv (\vartheta_1 \vee \vartheta_3) \varovee (\vartheta_2 \vee \vartheta_3) \tag{4}$$

$$\vartheta_1 \vee (\vartheta_2 \varovee \vartheta_3) \equiv (\vartheta_1 \vee \vartheta_2) \varovee (\vartheta_1 \vee \vartheta_3) \tag{5}$$

$$\exists x\, (\vartheta_1 \varovee \vartheta_2) \equiv (\exists x\, \vartheta_1) \varovee (\exists x\, \vartheta_2) \tag{6}$$

$$\exists x\, (\vartheta_1 \vee \vartheta_2) \equiv (\exists x\, \vartheta_1) \vee (\exists x\, \vartheta_2) \tag{7}$$

$$\exists x\, (\alpha \wedge \mathsf{E}\beta) \equiv (\exists x\, \alpha) \wedge \mathsf{E}\, \exists x\, (\alpha \wedge \beta) \tag{8}$$

$$\forall x\, (\vartheta_1 \wedge \vartheta_2) \equiv (\forall x\, \vartheta_1) \wedge (\forall x\, \vartheta_2) \tag{9}$$

$$\forall x \sim\!\vartheta \equiv \sim\!\forall x\, \vartheta \tag{10}$$

Proof. For (2), (3) and (7), see Lück [29, Lemma 4.13, 4.14 and D.1], respectively. For (4)–(6), see Galliani [13, Proposition 5]. For (9)–(10), see Väänänen [38, Chapter 8].

For (8), the direction "$\vDash$" is clear, as $\alpha \wedge \mathsf{E}\beta$ implies both α and $\mathsf{E}(\alpha \wedge \beta)$. For the converse direction, suppose $(\mathcal{A}, T) \vDash \exists x \alpha$ and $(\mathcal{A}, \hat{s}) \vDash \exists x(\alpha \wedge \beta)$ for some $\hat{s} \in T$. Then there are $f \colon T \to \mathfrak{P}(\mathcal{A}) \setminus \{\emptyset\}$ and $b \in \mathcal{A}$ such that $(\mathcal{A}, T^x_f) \vDash \alpha$ and $(\mathcal{A}, \hat{s}^x_b) \vDash \alpha \wedge \beta$. Define $g(\hat{s}) = f(\hat{s}) \cup \{b\}$ and $g(s) = f(s)$ for $s \in T \setminus \{\hat{s}\}$. Then $T^x_g = T^x_f \cup \{s^x_b\}$. Consequently, $(\mathcal{A}, T^x_g) \vDash \alpha \wedge \mathsf{E}\beta$, hence $(\mathcal{A}, T) \vDash \exists x\, (\alpha \wedge \mathsf{E}\beta)$. ◀

▶ **Lemma 4.1.** *Every* τ-$\mathsf{FO}^n_k(\sim)$*-formula* φ *is equivalent to a formula of the form*

$$\psi := \bigovee_{i=1}^{n} \left(\alpha_i \wedge \bigwedge_{j=1}^{m_i} \mathsf{E}\beta_{i,j}\right)$$

such that $\{\alpha_1, \ldots, \alpha_n, \beta_{1,1}, \ldots, \beta_{n,m_n}\} \subseteq \tau$-$\mathsf{FO}^n_k$ *and* $|\psi| \leq \exp_{\mathcal{O}(|\varphi|)}(1)$.

In what follows, *disjunctive normal form* (DNF) refers to formulas in the above form.

Proof. We construct the formula ψ by induction on φ. In each inductive step, it grows at most exponentially.

- If φ is a Boolean combination of FO^n_k-formulas (i.e., over $\sim$ and $\wedge$), then we obtain a DNF of size $\leq |\varphi| \cdot 2^{|\varphi|}$ similarly as for ordinary propositional logic.
- If $\varphi = \vartheta_1 \vee \vartheta_2$ for ϑ_1, ϑ_2 in DNF, then

$$\begin{aligned}
\varphi &= \bigovee_{i=1}^{n} \left(\alpha_i \wedge \bigwedge_{j=1}^{m_i} \mathsf{E}\beta_{i,j}\right) \vee \bigovee_{i=1}^{k} \left(\gamma_i \wedge \bigwedge_{j=1}^{\ell_i} \mathsf{E}\delta_{i,j}\right) \\
&\equiv \bigovee_{i=1}^{n} \bigvee_{j=1}^{m_i} (\alpha_i \wedge \mathsf{E}\beta_{i,j}) \vee \bigovee_{i=1}^{k} \bigvee_{j=1}^{\ell_i} (\gamma_i \wedge \mathsf{E}\delta_{i,j}) && \text{(Lemma A.5, (2))} \\
&\equiv \bigovee_{\substack{1 \leq i_1 \leq n \\ 1 \leq i_2 \leq k}} \bigvee_{j=1}^{m_{i_1}} (\alpha_{i_1} \wedge \mathsf{E}\beta_{i_1,j}) \vee \bigvee_{j=1}^{\ell_{i_2}} (\gamma_{i_2} \wedge \mathsf{E}\delta_{i_2,j}) && \text{(Lemma A.5, (4) and (5))} \\
&\equiv \bigovee_{i=1}^{n \cdot k} \bigvee_{j=1}^{o_i} (\mu_{i,j} \wedge \mathsf{E}\nu_{i,j}) && \text{(for some } \mu_{i,j}, \nu_{i,j} \in \mathsf{FO}^n_k\text{)} \\
&\equiv \bigovee_{i=1}^{n \cdot k} \Big(\bigvee_{j=1}^{o_i} \mu_{i,j}\Big) \wedge \bigwedge_{j=1}^{o_i} \mathsf{E}(\mu_{i,j} \wedge \nu_{i,j}), && \text{(Lemma A.5, (3))}
\end{aligned}$$

where the final DNF has size polynomial in $|\vartheta_1| + |\vartheta_2| \leq |\varphi|$.
- If $\varphi = \exists x\, \vartheta$ for ϑ in DNF, then

$$\begin{aligned}
\varphi &\equiv \exists x \bigovee_{i=1}^{n} \bigvee_{j=1}^{m_i} (\alpha_i \wedge \mathsf{E}\beta_{i,j}) && \text{(Lemma A.5, (2))} \\
&\equiv \bigovee_{i=1}^{n} \bigvee_{j=1}^{m_i} \exists x\, (\alpha_i \wedge \mathsf{E}\beta_{i,j}) && \text{(Lemma A.5, (6) and (7))} \\
&\equiv \bigovee_{i=1}^{n} \bigvee_{j=1}^{m_i} \Big((\exists x\, \alpha_i) \wedge \mathsf{E}\exists x\, (\alpha_i \wedge \beta_{i,j})\Big) && \text{(Lemma A.5, (8))} \\
&\equiv \bigovee_{i=1}^{n} \Big(\bigvee_{j=1}^{m_i} \exists x\, \alpha_i\Big) \wedge \bigwedge_{j=1}^{m_i} \mathsf{E}\exists x\, (\alpha_i \wedge \beta_{i,j})), && \text{(Lemma A.5, (3))}
\end{aligned}$$

which is again a DNF of polynomial size.
- Finally, the $\forall$ case is by repeated application of (9) and (10) of Lemma A.5. ◀