Polyteam Semantics [*]

Miika Hannula[1,2], Juha Kontinen[2], and Jonni Virtema[2,3,4]

[1] University of Auckland, New Zealand
[2] University of Helsinki, Finland, {miika.hannula,juha.kontinen}@helsinki.fi
[3] Hasselt University, Belgium
[4] Hokkaido University, Japan, jonni.virtema@let.hokudai.ac.jp

Abstract. Team semantics is the mathematical framework of modern logics of dependence and independence in which formulae are interpreted by sets of assignments (teams) instead of single assignments as in first-order logic. In order to deepen the fruitful interplay between team semantics and database dependency theory, we define *Polyteam Semantics* in which formulae are evaluated over a family of teams. We begin by defining a novel polyteam variant of dependence atoms and give a finite axiomatisation for the associated implication problem. We relate polyteam semantics to team semantics and investigate in which cases logics over the former can be simulated by logics over the latter. We also characterise the expressive power of poly-dependence logic by properties of polyteams that are downwards closed and definable in existential second-order logic (ESO). The analogous result is shown to hold for poly-independence logic and all ESO-definable properties. We also relate poly-inclusion logic to greatest fixed point logic.

Keywords: team semantics, dependence, independence, expressive power, existential second-order logic

1 Introduction

Team semantics is the mathematical framework of modern logics of dependence and independence. The origin of team semantics goes back to [18] but its development to its current form began with the publication of the monograph [27]. In team semantics formulae are interpreted by sets of assignments (teams) instead of single assignments as in first-order logic. The reason for this change is that statements such as *the value of a variable x depends on the value of y* do not really make sense for single assignments. Team semantics has interesting connections with database theory and database dependencies [13,15,16,21]. In order to facilitate the exchange between team semantics and database theory, we introduce a generalisation of team semantics in which formulae are evaluated over a family of teams. We identify a natural notion of poly-dependence that generalises dependence atoms to polyteams and give a finite axiomatisation for its implication problem. We also define polyteam versions of independence, inclusion and exclusion atoms, and characterise the expressive power of logics using these novel atoms.

A team X is a set of assignments with a common finite domain $x_1, \ldots, x_n$ of variables. Such a team can be viewed as a database table with $x_1, \ldots, x_n$ as its attributes. Dependence logic extends the language of first-order logic with atomic formulae $=(\overline{x}, y)$ called *dependence atoms* expressing that the value of the variable y is functionally determined by the values of the variables in $\overline{x}$. On the other hand, *independence atoms* $\overline{y} \perp_{\overline{x}} \overline{z}$ [11] express that, for any fixed value of $\overline{x}$, knowing the value of $\overline{z}$ does not tell us anything new about the value of $\overline{y}$. By viewing a team as a database table, the atoms $=(\overline{x}, y)$ and $\overline{y} \perp_{\overline{x}} \overline{z}$ correspond to the widely studied functional and embedded multivalued dependencies. Furthermore, inclusion atoms $\overline{x} \subseteq \overline{y}$ and exclusion atoms $\overline{x}|\overline{y}$ of [7] inherit their semantics from the corresponding database dependencies.

Independence, inclusion, and exclusion atoms have very interesting properties in the team semantics setting. For example, inclusion atoms give rise to a variant of dependence logic that corresponds to the complexity class PTIME over finite ordered structures [9] whereas all the other atoms above (and their combinations) give rise to logics that are equi-expressive with existential second-order logic and the complexity class NP. The complexity theoretic aspects of logics in team semantics have been studied extensively during the past few years (see [5] for a survey).

A multiset version of team semantics was recently defined in [3]. Multiteam semantics is motivated by the fact that multisets are widely assumed in database theory and occur in applications. Multiteam semantics widens the applicability of team semantics for the study of qualitative (e.g., functional dependence) and quantitative (e.g., probabilistic independence) dependencies in a unified framework. Recently multiteam semantics was further generalised to so-called probabilistic team semantics in which information on exact multiplicities are replaced by a (probability) distribution over the assignments [4,14].

The aim of this work is similar to that of [3], i.e., we want to extend the applicability of team semantics. In database theory dependencies are often expressed by so-called embedded dependencies. An *embedded dependency* is a sentence of first-order logic with equality of the form

$$\forall x_1 \ldots \forall x_n \big(\phi(x_1, \ldots, x_n) \to \exists y_1 \ldots \exists y_k \psi(x_1, \ldots, x_n, y_1, \ldots, y_k) \big),$$

where ϕ and ψ are conjunctions of relational atoms $R(x_1, \ldots, x_n)$ and equalities $x = y$. In the literature embedded dependencies have been thoroughly classified stemming from real life applications. Examples of well-known subclasses include *full*, *uni-relational*, *1-head*, *tuple-generating*, and *equality-generating*. For example, an embedded dependency is called *tuple-generating* if it is equality free (for further details see, e.g., [19, Section 3]). The uni-relational dependencies can be studied also in the context of team semantics as generalised dependencies [24]. However in many applications, especially in the area of data exchange and data integration, it is essential to be able to express dependencies between different relations.

In the context of data exchange (see, e.g., [6]) the relational database is divided into a set of source relations $\mathcal{S}$ and a set of target relations $\mathcal{T}$. Dependencies are used to describe what kind of properties should hold when data is transferred from the source schema to the target schema. In this setting a new taxonomy of embedded dependencies rises: An embedded dependency $\forall \overline{x} \big(\phi(\overline{x}) \to \exists \overline{y} \psi(\overline{x}, \overline{y}) \big)$ is *source-to-target* if the relation symbols occurring in ϕ and ψ are from $\mathcal{S}$ and $\mathcal{T}$, respectively. The embedded dependency is *target* if the relation symbols occurring in it are from $\mathcal{T}$. There is no direct way to study these classes of dependencies in the uni-relational setting of team semantics. In this paper we propose a general framework in which these inherently poly-relational dependencies can be studied.

In Section 2 we lay the foundations of polyteam semantics. The shift to polyteams is exemplified in Section 2.2 by the definition of poly-dependence atoms and an Armstrong type axiomatisation for the associated implication problem. In Section 3 polyteam semantics is extended from atoms to complex formulae. Section 4 studies the relationship between polyteam and team semantics. We show that some polyteam logics can be simulated with team logics. Section 5 examines the expressive power of the new logics over polyteams. We characterise poly-independence (poly-dependence) logic as the maximal logic capable of defining all (downwards closed) properties of polyteams definable in existential second-order logic. Finally we relate poly-inclusion logic to a fragment of greatest fixed point logic.

2 From uni-dependencies to poly-dependencies

We start by defining the familiar dependency notions from the team semantics literature. In Section 2.2 we introduce a novel poly-relational version of dependence atoms and establish a finite axiomatisation of its implication problem. We then continue to present poly-relational versions of inclusion, exclusion, and independence atoms, and a general notion of a poly-relational dependency atom. We conclude this section by relating the embedded dependencies studied in database theory to our new setting.

2.1 Dependencies in team semantics

Vocabularies τ are sets of relation symbols with prescribed arities. For each $R \in \tau$, let $\mathrm{ar}(R) \in Z_+$ denote the arity of R. A τ-structure is a tuple $\mathfrak{A} = \left(A, (R_i^{\mathfrak{A}})_{R_i \in \tau}\right)$, where A is a set and each $R_i^{\mathfrak{A}}$ is an $\mathrm{ar}(R_i)$-ary relation on A (i.e., $R_i^{\mathfrak{A}} \subseteq A^{\mathrm{ar}(R_i)}$). We use $\mathfrak{A}$, $\mathfrak{B}$, etc. to denote τ-structures and A, B, etc. to denote the corresponding domains.

Let D be a finite set of first-order variables and A be a nonempty set. A function $s \colon D \to A$ is called an *assignment*. For a variable x and $a \in A$, the assignment $s(a/x) \colon D \cup \{x\} \to A$ is obtained from s as follows:

$$s(a/x)(y) := \begin{cases} a & \text{if } y = x, \\ s(y) & \text{otherwise.} \end{cases}$$

A *team* X is a set of assignments with a common domain D and codomain A. If $\mathfrak{A}$ is a τ-structure and X a team with codomain A, then we say that X is a team of $\mathfrak{A}$. Let $\overline{x} = (x_1, \ldots, x_n)$ be a sequence of variables, and s an assignment. We write $s(\overline{x})$ to denote the sequence $\left(s(x_1), \ldots, s(x_n)\right)$, and $X(\overline{x})$ for the set of values $\{s(\overline{x}) \mid s \in X\}$. For a set (or sequence) of variables V, we write $X \upharpoonright V$ for the restriction of X to V.

The following dependency atoms were introduced in [27,7,11].

Definition 1 (Dependency atoms). *Let $\mathfrak{A}$ be a structure and X a team with codomain A. If $\overline{x}, \overline{y}$ are variable sequences, then $=(\overline{x}, \overline{y})$ is a dependence atom with the truth condition:*

$\mathfrak{A} \models_X =(\overline{x}, \overline{y})$ *if for all $s, s' \in X$ s.t. $s(\overline{x}) = s'(\overline{x})$, it holds that $s(\overline{y}) = s'(\overline{y})$.*

If $\overline{x}, \overline{y}$ are variable sequences of the same length, then $\overline{x} \subseteq \overline{y}$ is an inclusion atom and $\overline{x} \mid \overline{y}$ an exclusion atom with satisfaction defined as follows:

$\mathfrak{A} \models_X \overline{x} \subseteq \overline{y}$ *if for all $s \in X$ there exists $s' \in X$ such that $s(\overline{x}) = s'(\overline{y})$.*
$\mathfrak{A} \models_X \overline{x} \mid \overline{y}$ *if for all $s, s' \in X : s(\overline{x}) \neq s'(\overline{y})$.*

If $\overline{x}, \overline{y}, \overline{z}$ are variable sequences, then $\overline{y} \perp_{\overline{x}} \overline{z}$ is a conditional independence atom with satisfaction defined by

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

Note that in the previous definitions it is allowed that some or all of the vectors of variables have length 0. For example, $\mathfrak{A} \models_X =(\overline{y})$ holds iff $\forall s \in X : s(\overline{y}) = \overline{c}$ holds for some fixed tuple $\overline{c}$, where $=(\overline{y})$ denotes the dependence atom $=(\overline{x}, \overline{y})$ such that $\overline{x}$ is of length 0. Also, $\mathfrak{A} \models_X \overline{y} \perp_{\overline{x}} \overline{z}$ holds always if either of the vectors $\overline{y}$ or $\overline{z}$ is of length 0.

All the aforementioned dependency atoms have corresponding variants in relational databases. One effect of this relationship is that the axiomatic properties of these dependency atoms trace back to well-known results in database theory. Armstrong's axioms for functional dependencies constitute a finite axiomatisation for dependence atoms [1,11], and inclusion atoms can be finitely axiomatised using the axiomatisation for inclusion dependencies [2]. On the other hand, the non-axiomatisability and undecidability of the (finite and unrestricted) implication problem for embedded multivalued dependencies both carry over to conditional independence atoms [17,25,26]. Restricting attention to the so-called *pure independence atoms*, i.e., atoms of the form $\overline{x} \perp_\emptyset \overline{y}$, a finite axiomatisation is obtained by relating to marginal independence in statistics [10,21].

2.2 The notion of poly-dependence

For each $i \in \mathbb{N}$, let $\mathrm{Var}(i)$ denote a distinct countable set of first-order variable symbols. We say that these variables are of sort i. Relating to databases, sorts correspond to table names. Usually we set $\mathrm{Var}(i) = \{x_j^i \mid j \in \mathbb{N}\}$. We write x^i, y^i, x_j^i to denote variables from $\mathrm{Var}(i)$, and $\overline{x}^i$ to denote tuples of variables from $\mathrm{Var}(i)$. Sometimes we drop the index i and write simply x and $\overline{x}$ instead of x^i and $\overline{x}^i$, respectively. Note that $\overline{x}$ is always a tuple of variables of a single sort. In order to simplify notation, we sometimes write $\overline{x}^i$ and $\overline{x}^j$ to denote arbitrary tuples of variables of sort i and j, respectively. We emphasise that $\overline{x}^i$ and $\overline{x}^j$ might be of different length and may consist of distinct variables. Let $\mathfrak{A}$ be a τ-structure and let $D_i \subseteq \mathrm{Var}(i)$ for all $i \in \mathbb{N}$. A tuple $\overline{X} = (X_i)_{i \in \mathbb{N}}$ is a *polyteam* of $\mathfrak{A}$ with domain $\overline{D} = (D_i)_{i \in \mathbb{N}}$, if X_i is a team with domain D_i and co-domain A for each $i \in \mathbb{N}$. We identify $\overline{X}$ with $(X_0, \ldots, X_n)$ if X_i is the singleton team consisting of the empty assignment for all i greater than n. Let $\overline{X} = (X_i)_{i \in \mathbb{N}}$ and $\overline{Y} = (Y_i)_{i \in \mathbb{N}}$ be two polyteams. We say that $\overline{X}$ is a *subteam* of $\overline{Y}$ if $X_i \subseteq Y_i$ for all $i \in \mathbb{N}$. By the *union* (resp. *intersection*) of $\overline{X}$ and $\overline{Y}$ we denote the polyteam $(X_i \cup Y_i)_{i \in \mathbb{N}}$ (resp. $(X_i \cap Y_i)_{i \in \mathbb{N}}$). By a slight abuse of notation we write $\overline{X} \cup \overline{Y}$ (resp. $\overline{X} \cap \overline{Y}$) for the union (resp. intersection) of $\overline{X}$ and $\overline{Y}$, and $\overline{X} \subseteq \overline{Y}$ to denote that $\overline{X}$ is a subteam of $\overline{Y}$. For a tuple $\overline{V} = (V_i)_{i \in \mathbb{N}}$ where $V_i \subseteq \mathrm{Var}(i)$, the *restriction* of $\overline{X}$ to $\overline{V}$, written $\overline{X} \restriction \overline{V}$, is defined as $(X_i \restriction V_i)_{i \in \mathbb{N}}$.

Next we generalise dependence atoms to the polyteam setting. In contrast to the standard dependence atoms, poly-dependence atoms declare functional dependence of variables over two teams.

Poly-dependence. Let $\overline{x}^i \overline{y}^i$ and $\overline{u}^j \overline{v}^j$ be sequences of variables such that $\overline{x}^i$ and $\overline{u}^j$, and $\overline{y}^i$ and $\overline{v}^j$ have the same length, respectively. Then $= (\overline{x}^i, \overline{y}^i / \overline{u}^j, \overline{v}^j)$ is a *poly-dependence atom* whose satisfaction relation $\models_{\overline{X}}$ is defined as follows:

$$\mathfrak{A} \models_{\overline{X}} = (\overline{x}^i, \overline{y}^i / \overline{u}^j, \overline{v}^j) \Leftrightarrow \forall s \in X_i \forall s' \in X_j : s(\overline{x}^i) = s'(\overline{u}^j) \text{ implies } s(\overline{y}^i) = s'(\overline{v}^j).$$

Note that the atom $= (\overline{x}, \overline{y} / \overline{x}, \overline{y})$ corresponds to the dependence atom $= (\overline{x}, \overline{y})$. For empty tuples $\overline{x}^i$ and $\overline{u}^j$ the poly-dependence atom reduces to a "poly-constancy atom" $= (\overline{y}^i / \overline{v}^j)$. We will later show (Remark 14) that poly-dependence atoms of the form $= (\overline{x}^i, \overline{y}^i / \overline{u}^i, \overline{v}^i)$ can be expressed with formulae using only ordinary dependence atoms. Thus poly-dependence atoms of this form are considered as primitive notions only when $\overline{x}^i \overline{y}^i = \overline{u}^i \overline{v}^i$; otherwise $= (\overline{x}^i, \overline{y}^i / \overline{u}^i, \overline{v}^i)$ is considered as a shorthand for the equivalent formula obtained from Remark 14.

The ability to reason about database dependencies facilitates many data management tasks such as schema design, query optimisation, and integrity maintenance. Keys, inclusion dependencies, and functional dependencies in particular have a crucial role in all of these processes. A traditional way to approach the interaction between dependencies has been the

utilisation of proof systems similar to natural deduction systems in logic. The most significant of all these systems is Armstrong's axiomatisation for functional dependencies. This inference system consists of only three rules which we depict below using the standard notation for functional dependencies, i.e., $X \to Y$ denotes that an attribute set X functionally determines another attribute set Y.

Definition 2 (Armstrong's axiomatisation [1]).

- *Reflexivity: If $Y \subseteq X$, then $X \to Y$*
- *Augmentation: if $X \to Y$, then $XZ \to YZ$*
- *Transitivity: if $X \to Y$ and $Y \to Z$, then $X \to Z$*

Our first objective is to generalise Armstrong's axiomatisation to the poly-dependence setting. To this end, we assemble the three rules of Armstrong and introduce three auxiliary rules: Union, Symmetry, and Weak Transitivity. Contrary to the Armstrong's proof system, here Union is not reducible to Transitivity and Augmentation because we operate with sequences instead of sets of variables or attributes. Symmetry in turn is imposed by the sequential notation employed by the poly-dependence atom. Weak Transitivity exhibits transitivity of equalities on the right-hand side of a poly-dependence atom, a phenomenon that arises only in the polyteam setting.

Definition 3 (Axiomatisation for poly-dependence atoms).

- *Reflexivity: $= \bigl(\overline{x}^i, pr_k(\overline{x}^i)/\overline{y}^j, pr_k(\overline{y}^j)\bigr)$, where $k = 1, \ldots, |\overline{x}^i|$ and pr_k takes the kth projection of a sequence.*
- *Augmentation: if $= \bigl(\overline{x}^i, \overline{y}^i/\overline{u}^j, \overline{v}^j\bigr)$, then $= \bigl(\overline{x}^i\overline{z}^i, \overline{y}^i\overline{z}^i/\overline{u}^j\overline{w}^j, \overline{v}^j\overline{w}^j\bigr)$*
- *Transitivity: if $= \bigl(\overline{x}^i, \overline{y}^i/\overline{u}^j, \overline{v}^j\bigr)$ and $= \bigl(\overline{y}^i, \overline{z}^i/\overline{v}^j, \overline{w}^j\bigr)$, then $= \bigl(\overline{x}^i, \overline{z}^i/\overline{u}^j, \overline{w}^j\bigr)$*
- *Union: if $= \bigl(\overline{x}^i, \overline{y}^i/\overline{u}^j, \overline{v}^j\bigr)$ and $= \bigl(\overline{x}^i, \overline{z}^i/\overline{u}^j, \overline{w}^j\bigr)$ then $= \bigl(\overline{x}^i, \overline{y}^i\overline{z}^i/\overline{u}^j, \overline{v}^j\overline{w}^j\bigr)$*
- *Symmetry: if $= \bigl(\overline{x}^i, \overline{y}^i/\overline{u}^j, \overline{v}^j\bigr)$, then $= \bigl(\overline{u}^j, \overline{v}^j/\overline{x}^i, \overline{y}^i\bigr)$*
- *Weak Transitivity: if $= \bigl(\overline{x}^i, \overline{y}^i\overline{z}^i\overline{z}^i/\overline{u}^j, \overline{v}^j\overline{v}^j\overline{w}^j\bigr)$, then $= \bigl(\overline{x}^i, \overline{y}^i/\overline{u}^j, \overline{w}^j\bigr)$*

This proof system forms a complete characterisation of logical implication for poly-dependence atoms. We use $\models$ to refer to logical implication, i.e., we write $\Sigma \models \sigma$ if $\mathfrak{A} \models_{\overline{X}} \Sigma$ implies $\mathfrak{A} \models_{\overline{X}} \sigma$ for all models $\mathfrak{A}$ and polyteams $\overline{X}$. Given an *axiomatisation $\mathcal{R}$*, that is, a set of axioms and inference rules, we write $\Sigma \vdash_{\mathcal{R}} \sigma$ if $\mathcal{R}$ yields a proof of σ from Σ. Given a class of dependency atoms $\mathcal{C}$, we then say that $\mathcal{R}$ is sound (complete, resp.) for $\mathcal{C}$ if for all finite sets of dependency atoms $\Sigma \cup \{\sigma\}$ from $\mathcal{C}$, $\Sigma \vdash_{\mathcal{R}} \sigma$ implies (is implied by, resp.) $\Sigma \models \sigma$.

Theorem 4. *The axiomatisation of Def. 3 is sound and complete for poly-dependence atoms.*

Proof. The proof of soundness is straightforward and omitted. We show that the axiomatisation is complete, i.e., that $\Sigma \models \sigma$ implies $\Sigma \vdash \sigma$ for a set $\Sigma \cup \{\sigma\}$ of poly-dependence atoms. Assume σ is $= \bigl(\overline{x}^i, \overline{y}^i/\overline{x}^j, \overline{y}^j\bigr)$. First we consider the case where $i = j$ in which case σ is a standard dependence atom. Let Σ^* be the subset of Σ consisting of all standard dependence atoms over $\mathrm{Var}(i)$. Since all teams satisfying Σ^* can be extended to a polyteam satisfying Σ by introducing new empty teams, we have that $\Sigma^* \models \sigma$ in the team semantics setting. Since dependence atoms $= (\overline{x}, \overline{y})$ in team semantics correspond to functional dependencies $\{x \in \overline{x}^i\} \to \{y \in \overline{y}^i\}$ in relational databases (see e.g. [11]), Armstrong's complete axiomatisation from Definition 2 yields a deduction of σ_0 from Σ_0^* where Σ_0^* and $\{\sigma_0\}$ are obtained from Σ^* and σ by replacing dependence atoms with their corresponding functional dependencies. Since dependence atoms are provably order-independent (i.e. one derives $= (\overline{x}_0, \overline{x}_1)$

from $= (\overline{y}_0, \overline{y}_1)$ by Reflexivity, Union, and Transitivity if $\overline{x}_i$ and $\overline{y}_i$ list the same variables), the deduction in Armstrong's system can be simulated with the rules in Definition 3. This proves the case $i = j$.

Let us then consider the case $i \neq j$. We will show that $\Sigma \nvdash \sigma$ implies $\Sigma \nvDash \sigma$. Assume $\Sigma \nvdash \sigma$. Define first a binary relation $\sim$ on $\mathrm{Var}(i) \cup \mathrm{Var}(j)$ such that $a^i \sim a^j$ if $\Sigma \vdash= \left(\overline{x}^i, a^i/\overline{x}^j, a^j\right)$, $a^j \sim a^i$ if $\Sigma \vdash= \left(\overline{x}^j, a^j/\overline{x}^i, a^i\right)$, and $a^i \sim b^i$ ($a^j \sim b^j$, resp.) if $a^i = b^i$ or $\Sigma \vdash= \left(\overline{x}^i, a^ib^i/\overline{x}^j, a^ja^j\right)$ for some a^j ($a^j = b^j$ or $\Sigma \vdash= \left(\overline{x}^j, a^jb^j/\overline{x}^i, a^ia^i\right)$ for some a^i, resp.). We show that $\sim$ is an equivalence relation.

- Reflexivity: Holds by definition.
- Symmetry: First note that $a^i \sim a^j$ and $a^j \sim a^i$ are derivably equivalent by the symmetry rule. Assume then that $a^i \sim b^i$ in which case $= \left(\overline{x}^i, a^ib^i/\overline{x}^j, a^ja^j\right)$ is derivable for some a^j. Then derive $= (a^ib^i, b^i/a^ja^j, a^j)$ and $= (a^ib^i, a^i/a^ja^j, a^j)$ by using the reflexivity rule, and then $= \left(\overline{x}^i, b^i/\overline{x}^j, a^j\right)$ and $= \left(\overline{x}^i, a^i/\overline{x}^j, a^j\right)$ by using the transitivity rule. Finally derive $= \left(\overline{x}^i, b^ia^i/\overline{x}^j, a^ja^j\right)$ by using the union rule.
- Transitivity: Assume first that $a^i \sim b^i \sim c^i$, where a^i, b^i, c^i and are pairwise distinct. Then $= \left(\overline{x}^i, a^ib^i/\overline{x}^j, a^ja^j\right)$ and $= \left(\overline{x}^i, b^ic^i/\overline{x}^j, b^jb^j\right)$ are derivable for some a^j and b^j. Then analogously to the previous case assemble $= \left(\overline{x}^i, a^ib^ib^i/\overline{x}^j, a^ja^jb^j\right)$ which admits $= \left(\overline{x}^i, a^i/\overline{x}^j, b^j\right)$ by weak transitivity, and detach $= \left(\overline{x}^i, c^i/\overline{x}^j, b^j\right)$ from $= \left(\overline{x}^i, b^ic^i/\overline{x}^j, b^jb^j\right)$. By the union rule we then obtain $= \left(\overline{x}^i, a^ic^i/\overline{x}^j, b^jb^j\right)$ and thus that $a^i \sim c^i$. Since all the other cases are analogous, we observe that $\sim$ is transitive.

Let s be a function that maps each $x \in \mathrm{Var}(i) \cup \mathrm{Var}(j)$ that appears in $\Sigma \cup \{\sigma\}$ to the equivalence class $x/\sim$. We define $\overline{X} = (X_i, X_j)$ where $X_k = \{s \restriction \mathrm{Var}(k)\}$ for $k = i, j$. First notice that $\overline{X} \nvDash \sigma$ for, by union, it cannot be the case that $\mathrm{pr}_k(\overline{y}^i) \sim \mathrm{pr}_k(\overline{y}^j)$ for all $k = 1, \ldots, |\overline{y}^i|$. It suffices to show that $\overline{X}$ satisfies each $= (\overline{u}^m, \overline{v}^m/\overline{u}^n, \overline{v}^n)$ in Σ. If $m = n$ or $\{m, n\} \neq \{i, j\}$, the atom is trivially satisfied. Hence, and by symmetry, we may assume that the atom is of the form $= \left(\overline{u}^i, \overline{v}^i/\overline{u}^j, \overline{v}^j\right)$. Assume that $s(\overline{u}^i) = s(\overline{u}^j)$, that is, $\mathrm{pr}_k(\overline{u}^i) \sim \mathrm{pr}_k(\overline{u}^j)$ for all $k = 1, \ldots, |\overline{u}^i|$. We obtain by the union rule that $= \left(\overline{x}^i, \overline{u}^i/\overline{x}^j, \overline{u}^j\right)$ is derivable, and hence by the transitivity rule that $= \left(\overline{x}^i, \overline{v}^i/\overline{x}^j, \overline{v}^j\right)$ is also derivable. Therefore, by using the reflexivity and transitivity rules we conclude that $s(\overline{v}^i) = s(\overline{v}^j)$. $\qquad\square$

2.3 A general notion of a poly-dependency

Next we consider suitable polyteam generalisations for the dependencies discussed in Section 2.1 and also define a general notion of poly-dependency. This generalisation is immediate for inclusion atoms which are inherently multi-relational; relational database management systems maintain referential integrity by enforcing inclusion dependencies specifically between two distinct tables. With poly-inclusion atoms these multi-relational features can now be expressed.

Poly-inclusion. Let $\overline{x}^i$ and $\overline{y}^j$ be sequences of variables of the same length. Then $\overline{x}^i \subseteq \overline{y}^j$ is a *poly-inclusion atom* whose satisfaction relation $\models_{\overline{X}}$ is defined as follows:

$$\mathfrak{A} \models_{\overline{X}} \overline{x}^i \subseteq \overline{y}^j \Leftrightarrow \forall s \in X_i \exists s' \in X_j : s(\overline{x}^i) = s'(\overline{y}^j).$$

If $i = j$, then the atom is the standard inclusion atom.

Poly-exclusion. Let $\overline{x}^i$ and $\overline{y}^j$ be sequences of variables of the same length. Then $\overline{x}^i \mid \overline{y}^j$ is a *poly-exclusion atom* whose satisfaction relation $\models_{\overline{X}}$ is defined as follows:

$$\mathfrak{A} \models_{\overline{X}} \overline{x}^i \mid \overline{y}^j \Leftrightarrow \forall s \in X_i, s' \in X_j : s(\overline{x}^i) \neq s'(\overline{y}^j).$$

If $i = j$, then the atom is the standard exclusion atom.

Poly-independence Let $\overline{x}^i$, $\overline{y}^i$, $\overline{a}^j$, $\overline{b}^j$, $\overline{u}^k$, $\overline{v}^k$, and $\overline{w}^k$ be tuples of variables such that $|\overline{x}^i| = |\overline{a}^j| = |\overline{u}^k|$, $|\overline{y}^i| = |\overline{v}^k|$, $|\overline{b}^j| = |\overline{w}^k|$. Then $\overline{y}^i/\overline{v}^k \perp_{\overline{x}^i,\overline{a}^j/\overline{u}^k} \overline{b}^j/\overline{w}^k$ is a *poly-independence atom* whose satisfaction relation $\models_{\overline{X}}$ is defined as follows:

$$\mathfrak{A} \models_{\overline{X}} \overline{y}^i/\overline{v}^k \perp_{\overline{x}^i,\overline{a}^j/\overline{u}^k} \overline{b}^j/\overline{w}^k \Leftrightarrow \forall s \in X_i, s' \in X_j : s(\overline{x}^i) = s'(\overline{a}^j) \text{ implies}$$

$$\exists s'' \in X_k : s''(\overline{u}^k\overline{v}^k) = s(\overline{x}^i\overline{y}^i) \text{ and } s''(\overline{w}^k) = s'(\overline{b}^j).$$

The atom $\overline{y}/\overline{y} \perp_{\overline{x},\overline{x}/\overline{x}} \overline{z}/\overline{z}$, where all variables are of the same sort, corresponds to the standard independence atom $\overline{y} \perp_{\overline{x}} \overline{z}$. Furthermore, a *pure poly-independence atom* is an atom of the form $\overline{y}^i/\overline{v}^k \perp_{\emptyset,\emptyset/\emptyset} \overline{b}^j/\overline{w}^k$, written using a shorthand $\overline{y}^i/\overline{v}^k \perp \overline{b}^j/\overline{w}^k$.

Poly-independence atoms are closely related to equi-join operators of relational databases as the next example exemplifies.

Example 5. A relational database schema

$$\mathrm{P}(\textsc{rojects}) = \{\texttt{project},\texttt{team}\}, \quad \mathrm{T}(\textsc{eams}) = \{\texttt{team},\texttt{employee}\},$$
$$\mathrm{E}(\textsc{mployees}) = \{\texttt{employee},\texttt{team},\texttt{project}\},$$

stores information about distribution of employees for teams and projects in a workplace. The poly-independence atom

$$\mathrm{P}[\texttt{project}]/\mathrm{E}[\texttt{project}] \perp_{\mathrm{P}[\texttt{team}],\mathrm{T}[\texttt{team}]/\mathrm{E}[\texttt{team}]} \mathrm{T}[\texttt{employee}]/\mathrm{E}[\texttt{employee}] \tag{1}$$

expresses that the relation $\textsc{Employees}$ includes as a subrelation the natural join of $\textsc{Projects}$ and $\textsc{Teams}$. If furthermore $\mathrm{E}[\texttt{project},\texttt{team}] \subseteq \mathrm{P}[\texttt{project},\texttt{team}]$ and $\mathrm{E}[\texttt{team},\texttt{employee}] \subseteq \mathrm{T}[\texttt{team},\texttt{employee}]$ hold, then $\textsc{Employees}$ is exactly this natural join.

In addition to the poly-atoms described above, we define the notion of generalised poly-atoms that is analogous to the notion of generalised atoms of [24].

Generalised poly-atoms. Let $n \in \mathbb{N}$, and let $(j_1,\ldots,j_n)$ be a sequence of positive integers. A *generalised quantifier* of type $(j_1,\ldots,j_n)$ is a collection Q of relational structures $(A, R_1,\ldots,R_n)$ (where each R_i is j_i-ary) that is closed under isomorphisms. For every sequence $(\overline{x}_1,\ldots,\overline{x}_n)$, where $\overline{x}_i$ are length j_i tuples of variables from some $\mathrm{Var}(l_i)$, $\alpha_Q(\overline{x}_1,\ldots,\overline{x}_n)$ is a *generalised poly-atom* of type $(j_1,\ldots,j_n)$ and of sort $\{l_1,\ldots,l_n\}$. For a structure $\mathfrak{A}$ and polyteam $\overline{X}$ where $\overline{x}_i \subseteq \mathsf{Dom}(X_{l_i})$, the satisfaction relation with respect to $\alpha_Q(\overline{x}_1,\ldots,\overline{x}_n)$ is defined as follows:

$$\mathfrak{A} \models_{\overline{X}} \alpha_Q(\overline{x}_1,\ldots,\overline{x}_n) \Leftrightarrow \Big(\mathsf{Dom}(\mathfrak{A}), R_1 := \mathrm{rel}(X_{l_1},\overline{x}_1)\ldots, R_n := \mathrm{rel}(X_{l_n},\overline{x}_n)\Big) \in Q.$$

By $\mathrm{rel}(X,\overline{x})$, for $\overline{x} = (x_1,\ldots,x_m)$, we denote the relation $\{(s(x_1),\ldots,s(x_m)) \mid s \in X\}$. A generalised poly-atom $\alpha_Q(\overline{x}_1,\ldots,\overline{x}_n)$ that has a singleton sort is called a *uni-atom*. When referring to the set of all poly-atoms of the form $\alpha_Q(\overline{x}_1,\ldots,\overline{x}_n)$, for a fixed Q, we omit the tuples $\overline{x}_1,\ldots,\overline{x}_n$ and write simply poly-atom α_Q. We say that an atom α_Q is definable in a logic $\mathcal{L}$ if the class Q is definable in $\mathcal{L}$. For instance, we notice that poly-inclusion atoms of the form $(x^i,y^i) \subseteq (u^j,v^j)$ are first-order definable generalised poly-atoms of type $(2,2)$.

2.4 Database dependencies as poly-atoms

Embedded dependencies in a multi-relational context can now be studied with the help of generalised poly-atoms and polyteam semantics. Conversely, strong results obtained in the study of database dependencies can be transferred and generalised for stronger results in

the polyteam setting. In particular, each embedded dependency can be seen as a defining formula for a generalised poly-atom, and hence the classification of embedded dependencies naturally yield a corresponding classification of generalised poly-atoms. For example, the class

$$\mathcal{C} := \{\alpha_Q(\overline{x}_1, \ldots, \overline{x}_n) \mid Q \text{ is definable by an } \mathsf{FO}(R_1, \ldots, R_n)\text{-sentence in}$$
$$\text{the class of equality-generating dependencies}\}$$

is the class of *equality-generating* poly-atoms. The defining formula of the generalised atom of type (2,2) that captures the poly-dependence atom of type $=(x^i, y^i/u^j, v^j)$ is

$$\forall x_1 \forall x_2 \forall y_1 \forall y_2 \big((R_1(x_1, x_2) \land R_2(y_1, y_2) \land x_1 = y_1) \to x_2 = y_2 \big).$$

Thus poly-dependence atoms are included in the class of equality-generating poly-atoms.

In order to study data exchange in the polyteam setting, we first need to define the notions of *source-to-target* and *target* poly-atoms. This classification of poly-atoms requires some more care as it is not enough to consider the defining formulae of the corresponding atoms, but also the variables that the atom is instantiated with. We will return to this topic briefly after we have given semantics for logics that work on polyteams.

3 Polyteam semantics for complex formulae

We next delineate a version of team semantics suitable for the polyteam context. We note here that it is not a priori clear what sort of modifications for connectives and quantifiers one should entertain when shifting from teams to the polyteam setting.

3.1 Syntax and semantics

Definition 6. *Let τ be a set of relation symbols. The syntax of* poly first-order logic $\mathsf{PFO}(\tau)$ *is given by the following grammar rules:*

$$\phi ::= x = y \mid x \neq y \mid R(\overline{x}) \mid \neg R(\overline{x}) \mid (\phi \land \phi) \mid (\phi \lor \phi) \mid (\phi \lor^j \phi) \mid \exists x \phi \mid \forall x \phi,$$

where $R \in \tau$ is a k-ary relation symbol, $j \in \mathbb{N}$, $\overline{x} \subseteq Var(i)^k$ and $x, y \in Var(i)$ for some $i, k \in \mathbb{N}$.

We say that $\lor$ is a *global disjunction* whereas $\lor^i$ is a *local disjunction*. A literal is said to be of sort i if its variables are of sort i. Note that in the definition the scope of negation is restricted to atomic formulae. Note also that the restriction of $\mathsf{PFO}(\tau)$ to formulae without the connective $\lor^j$ and using only variables of a single fixed sort is $\mathsf{FO}(\tau)$.

For the definition of polyteam semantics of PFO, recall the definitions of teams and polyteams from Sections 2.1 and 2.2, respectively. Let X be a team, A a non-empty set, and $F \colon X \to \mathcal{P}(A) \setminus \{\emptyset\}$ a function. We denote by $X[A/x]$ the modified team $\{s(a/x) \mid s \in X, a \in A\}$, and by $X[F/x]$ the team $\{s(a/x) \mid s \in X, a \in F(s)\}$. Moreover let $\overline{X}$ be a polyteam. Then $\overline{X}[X/X_i]$ denotes the polyteam $(\ldots, X_{i-1}, X, X_{i+1}, \ldots)$.

Note that if restricted to the aforementioned single-sort fragment of $\mathsf{PFO}(\tau)$ the polyteam semantics below coincides with traditional team semantics, see e.g. [5] for a definition. Thus for $\mathsf{FO}(\tau)$-formulae we may write $\mathfrak{A} \models_{X_i} \phi$ instead of $\mathfrak{A} \models_{(X_i)} \phi$.

Definition 7 (Lax polyteam semantics). *Let $\mathfrak{A}$ be a τ-structure and $\overline{X}$ a polyteam of $\mathfrak{A}$. The satisfaction relation $\models_{\overline{X}}$ for poly first-order logic is defined as follows:*

8

$$\mathfrak{A} \models_{\overline{X}} x = y \quad \Leftrightarrow \textit{if } x, y \in \textit{Var}(i) \textit{ then } \forall s \in X_i : s(x) = s(y)$$
$$\mathfrak{A} \models_{\overline{X}} x \neq y \quad \Leftrightarrow \textit{if } x, y \in \textit{Var}(i) \textit{ then } \forall s \in X_i : s(x) \neq s(y)$$
$$\mathfrak{A} \models_{\overline{X}} R(\overline{x}) \quad \Leftrightarrow \textit{if } \overline{x} \in \textit{Var}(i)^k \textit{ then } \forall s \in X_i : s(\overline{x}) \in R^{\mathfrak{A}}$$
$$\mathfrak{A} \models_{\overline{X}} \neg R(\overline{x}) \quad \Leftrightarrow \textit{if } \overline{x} \in \textit{Var}(i)^k \textit{ then } \forall s \in X_i : s(\overline{x}) \notin R^{\mathfrak{A}}$$
$$\mathfrak{A} \models_{\overline{X}} (\psi \wedge \theta) \quad \Leftrightarrow \mathfrak{A} \models_{\overline{X}} \psi \textit{ and } \mathfrak{A} \models_{\overline{X}} \theta$$
$$\mathfrak{A} \models_{\overline{X}} (\psi \vee \theta) \quad \Leftrightarrow \mathfrak{A} \models_{\overline{Y}} \psi \textit{ and } \mathfrak{A} \models_{\overline{Z}} \theta \textit{ for some } \overline{Y}, \overline{Z} \subseteq \overline{X} \textit{ s.t. } \overline{Y} \cup \overline{Z} = \overline{X}$$
$$\mathfrak{A} \models_{\overline{X}} (\psi \vee^j \theta) \Leftrightarrow \mathfrak{A} \models_{\overline{X}[Y_j/X_j]} \psi \textit{ and } \mathfrak{A} \models_{\overline{X}[Z_j/X_j]} \theta,$$
$$\textit{for some } Y_j, Z_j \subseteq X_j \textit{ s.t. } Y_j \cup Z_j = X_j$$
$$\mathfrak{A} \models_{\overline{X}} \forall x \psi \quad \Leftrightarrow \mathfrak{A} \models_{\overline{X}[X_i[A/x]/X_i]} \psi, \textit{ when } x \in \textit{Var}(i)$$
$$\mathfrak{A} \models_{\overline{X}} \exists x \psi \quad \Leftrightarrow \mathfrak{A} \models_{\overline{X}[X_i[F/x]/X_i]} \psi \textit{ holds for some } F \colon X_i \to \mathcal{P}(A) \setminus \{\emptyset\},$$
$$\textit{when } x \in \textit{Var}(i)$$

Remark 8. Note that whereas the global disjunction is both commutative and associative, the local disjunction is only commutative. In particular $(\phi \vee^i \psi) \vee^j \theta$ is not, in general, equivalent with $\phi \vee^i (\psi \vee^j \theta)$. However the local disjunction is associative with respect to local disjunctions of the same sort, i.e., $(\phi \vee^i \psi) \vee^i \theta$ and $\phi \vee^i (\psi \vee^i \theta)$ are equivalent.

The truth of a *sentence* ϕ (i.e., a formula with no free variables) in a structure $\mathfrak{A}$ is defined as: $\mathfrak{A} \models \phi$ if $\mathfrak{A} \models_{(\{\emptyset\})} \phi$, where $(\{\emptyset\})$ denotes the polyteam consisting only singleton teams of the empty assignment. We write $\text{Fr}(\phi)$ for the set of free variables in ϕ, and $\text{Fr}_i(\phi)$ for $\text{Fr}(\phi) \cap \text{Var}(i)$.

Polyteam semantics is a conservative extension of team semantics in the same fashion as team semantics is a conservative extension of Tarski semantics [27].

Proposition 9. *Let $\phi \in \mathsf{FO}(\tau)$ whose variables are all of sort $i \in \mathbb{N}$. Let $\mathfrak{A}$ be a τ-structure and $\overline{X}$ a polyteam of $\mathfrak{A}$. Then*

$$\mathfrak{A} \models_{\overline{X}} \phi \Leftrightarrow \mathfrak{A} \models_{X_i} \phi \Leftrightarrow \forall s \in X_i : \mathfrak{A} \models_s \phi,$$

where $\models_s$ denotes the ordinary satisfaction relation of first-order logic.

Example 10. A relational database schema

$$\begin{aligned} \text{PATIENT} &= \{\texttt{patient\_id}, \texttt{patient\_name}\}, \\ \text{CASE} &= \{\texttt{case\_id}, \texttt{patient\_id}, \texttt{diagnosis\_id}, \texttt{confirmation}\}, \\ \text{TEST} &= \{\texttt{diagnosis\_id}, \texttt{test\_id}\}, \\ \text{RESULTS} &= \{\texttt{patient\_id}, \texttt{test\_id}, \texttt{result}\} \end{aligned}$$

stores information about patient cases and their related laboratory tests. In order to maintain consistency of the stored data, database management systems support the use of integrity constraints that are based on functional and inclusion dependencies. For instance, on relation schema PATIENT the key $\texttt{patient\_id}$ (i.e. the dependence atom $= (\texttt{patient\_id}, \texttt{patient\_name})$) ensures that no patient id can refer to two different patient names. On CASE the foreign key $\texttt{patient\_id}$ referring to $\texttt{patient\_id}$ on PATIENT (i.e. the inclusion atom CASE$[\texttt{patient\_id}] \subseteq$ PATIENT$[\texttt{patient\_id}]$) enforces that patient ids on CASE refer to real patients. The introduction of poly-dependence logics opens up possibilities for more expressive data constraints. The poly-inclusion formula

$$\phi_0 := \texttt{confirmation} \neq \textit{positive} \vee^{\text{CASE}} \exists x_1 x_2 \big(x_1 \neq x_2 \wedge$$
$$\bigwedge_{i=1,2} (\text{CASE}[\texttt{diagnosis\_id}, x_i] \subseteq \text{TEST}[\texttt{diagnosis\_id}, \texttt{test\_id}] \wedge$$
$$\text{CASE}[\texttt{patient\_id}, x_i, \textit{positive}] \subseteq \text{RESULTS}[\texttt{patient\_id}, \texttt{test\_id}, \texttt{result}]))$$

ensures that a diagnosis may be confirmed only if it has been affirmed by two different appropriate tests. The poly-exclusion formula

$$\phi_1 := \text{confirmation} \neq negative \vee^{\text{CASE}}$$
$$\forall x \big(\text{CASE}[\texttt{diagnosis\_id}, x] \mid \text{TEST}[\texttt{diagnosis\_id,test\_id}] \vee^{\text{CASE}}$$
$$\text{CASE}[\texttt{patient\_id}, x, positive] \mid \text{RESULTS}[\texttt{patient\_id,test\_id,result}] \big)$$

makes sure that a diagnosis may obtain a negative confirmation only if it has no positive indication by any suitable test. Note that both formulae employ local disjunction and quantified variables that refer to CASE. Interestingly, the illustrated expressive gain is still computationally feasible as both ϕ_0 and ϕ_1 can be enforced in polynomial time. For ϕ_0 note that the data complexity of poly-inclusion logic is in PTIME because this logic can be translated to fixed-point logic (see Theorem 36). For ϕ_1 observe that satisfaction of a formula of the form $\overline{x}^1 \mid \overline{y}^2 \vee^1 \overline{x}^1 \mid \overline{z}^3$ can be decided in PTIME as well.

Poly-dependence logics. *Poly-dependence, poly-independence, poly-inclusion,* and *poly-exclusion logics* (PFO(pdep), PFO(pind), PFO(pinc), and PFO(pexc), resp.) are obtained by extending PFO with poly-dependence, poly-independence, poly-inclusion, and poly-exclusion atoms, respectively. In general, given a set of atoms $\mathcal{C}$ we denote by $\text{PFO}(\mathcal{C})$ the logic obtained by extending PFO with the atoms of $\mathcal{C}$. We also consider poly-atoms in the team semantics setting; by $\text{FO}(\mathcal{C})$ we denote the extension of first-order logic by the poly-atoms in $\mathcal{C}$. Similarly, it is also possible to consider atoms of Section 2.1 in the polyteam setting by requiring that the variables used with each atom are of a single sort. For two logics $\mathcal{L}$ and $\mathcal{L}'$ in polyteam setting, we write $\mathcal{L} \leq \mathcal{L}'$ if for all $\phi \in \mathcal{L}$ there is $\phi' \in \mathcal{L}'$ such that $\mathfrak{A} \models_{\overline{X}} \phi$ if and only if $\mathfrak{A} \models_{\overline{X}} \phi'$, for all structures $\mathfrak{A}$ and polyteams $\overline{X}$. We also write $\mathcal{L} \equiv \mathcal{L}'$ if $\mathcal{L} \leq \mathcal{L}'$ and $\mathcal{L}' \leq \mathcal{L}$. We define "$\leq$" and "$\equiv$" analogously for two logics in the team setting.

3.2   Basic properties

A polyteam $\overline{X}$ is called *strictly non-empty*, if none of the teams X_i, $i \in \mathbb{N}$, is empty. We say that a formula ϕ is *local* in polyteam semantics if for all $\overline{V} = (V_i)_{i \in \mathbb{N}}$ where $\text{Fr}_i(\phi) \subseteq V_i$ for $i \in \mathbb{N}$, and all structures $\mathfrak{A}$ and polyteams $\overline{X}$, we have

$$\mathfrak{A} \models_{\overline{X}} \phi \Leftrightarrow \mathfrak{A} \models_{\overline{X} \restriction \overline{V}} \phi.$$

The truth value of a local formula depends only on the free variables at each coordinate, including those coordinates where the set of free variables is empty. For instance, the formula $\exists x^1 (x^1 \neq x^1)$ is true if and only if the first coordinate team is empty. The truth value of a local formula on a strictly non-empty polyteam depends only on the values of its free variables. We now call a logic $\mathcal{L}$ *local* if all its formulae are local.

Proposition 11 (Locality). *For any set $\mathcal{C}$ of generalised poly-atoms $\text{PFO}(\mathcal{C})$ is local.*

Furthermore, the downward closure of dependence logic as well as the union closure of inclusion logic generalise to polyteams.

Proposition 12 (Downward Closure and Union Closure). *Let ϕ be a formula of* $\text{PFO}(\text{pdep})$*, ψ a formula of* $\text{PFO}(\text{pinc})$*, $\mathfrak{A}$ a model, and $\overline{X}, \overline{Y}$ two polyteams. Then $\mathfrak{A} \models_{\overline{X}} \phi$ and $\overline{Y} \subseteq \overline{X}$ implies that $\mathfrak{A} \models_{\overline{Y}} \phi$, and $\mathfrak{A} \models_{\overline{X}} \psi$ and $\mathfrak{A} \models_{\overline{Y}} \psi$ implies that $\mathfrak{A} \models_{\overline{X} \cup \overline{Y}} \psi$.*

The following proposition shows that the replacement of independence (dependence) atoms with any (downwards closed) class of atoms definable in existential second-order logic

(ESO) results in no expressive gain, if the empty team is ignored. Note that dependence and independence logic formulae are always true for the empty team. We say that a formula ϕ over team semantics has the *empty team property* if $\mathfrak{A} \models_\emptyset \phi$ for all models $\mathfrak{A}$, and a logic $\mathcal{L}$ has the empty team property is all of its formulae have it.

Proposition 13. *Let $\mathcal{C}$ ($\mathcal{D}$, resp.) be the class of all (all downwards closed, resp.)* ESO-*definable poly-atoms. With respect to non-empty teams,* $\mathsf{FO}(\mathcal{C}) \equiv \mathsf{FO}(\mathrm{ind})$, $\mathsf{FO}(\mathcal{D}) \equiv \mathsf{FO}(\mathrm{dep})$. *If the atoms in $\mathcal{C}$ and $\mathcal{D}$ have the empty team property, the restriction to non-empty teams can be lifted. Moreover* $\mathsf{FO}(\mathrm{pinc}) \equiv \mathsf{FO}(\mathrm{inc})$.

Proof. The claim $\mathsf{FO}(\mathrm{pinc}) \equiv \mathsf{FO}(\mathrm{inc})$ follows directly from the observation that in the team semantics setting poly-inclusion atoms are exactly inclusion atoms. Note that $\mathsf{FO}(\mathrm{ind})$ ($\mathsf{FO}(\mathrm{dep})$, resp.) captures all (all downwards closed, resp.) ESO-definable properties of teams which include the empty team (see Theorem 24). It is easy to show (cf. [20, Lemma 5]) that every property of teams definable in $\mathsf{FO}(\mathcal{C})$ ($\mathsf{FO}(\mathcal{D})$, resp.) is ESO-definable (ESO-definable and downwards closed, resp.). Thus since $\mathrm{ind} \in \mathcal{C}$ and $\mathrm{dep} \in \mathcal{D}$, we obtain that $\mathsf{FO}(\mathcal{C}) \equiv \mathsf{FO}(\mathrm{ind})$ and $\mathsf{FO}(\mathcal{D}) \equiv \mathsf{FO}(\mathrm{dep})$ with respect to non-empty teams. Finally note that if all atoms in $\mathcal{C}$ and $\mathcal{D}$ have the empty team property, the logics $\mathsf{FO}(\mathcal{C})$ and $\mathsf{FO}(\mathcal{D})$ have it as well. □

Remark 14. In particular it follows from the previous proposition that, in the polyteam setting, each occurrence of any (any downwards closed, resp.) ESO-definable poly-atom (with the empty team property) that takes variables of a single sort as parameters may be equivalently expressed by a formula of $\mathsf{PFO}(\mathrm{ind})$ ($\mathsf{PFO}(\mathrm{dep})$, resp.) that only uses variables of the same single sort.

We end this section by considering the relationship of global and local disjunctions. In particular, we observe that either one of the two disjunctions could be omitted from PFO without influencing the expressivity of the logic. To facilitate our construction, we here allow the use of disjunctions of type $\vee^I$, where I is a set of indices, with the obvious semantics. We then show that $\vee$ can be replaced by $\vee^I$ and $\vee^I$ by $\vee^i$.

Proposition 15. *For every formula ϕ of PFO there exists an equivalent formula ψ_l (ψ_g, resp.) of PFO in which all disjunctions are local (global, resp.).*

Proof. **Local case.** Let ϕ be a formula of PFO and let I list the sorts of all the variables that occur in ϕ. First, we let ϕ' denote the formula obtained from ϕ by substituting all occurrences of $\vee$ by $\vee^I$. It is a direct consequence of the locality property that ϕ and ϕ' are equivalent.

We will next show how to eliminate disjunctions of type $\vee^I$. Without loss of generality, we restrict to models of cardinality at least two. Let $\phi = \psi \vee^I \theta$ be a formula of PFO and let $I = \{i_1, \ldots, i_n\}$. Define

$$\phi^+ := \bigsqcup_{i \in I} x^i y^i (\xi_l \wedge \xi_r),$$

where, for each $i \in I$, the variables x^i and y^i are fresh and distinct, and

$$\xi_l := (x^{i_1} = y^{i_1} \vee^{i_1} (x^{i_1} \neq y^{i_1} \wedge (x^{i_2} = y^{i_2} \vee^{i_2} (x^{i_2} \neq y^{i_2}$$
$$\wedge (\ldots \wedge (x^{i_n} = y^{i_n} \vee^{i_n} (x^{i_n} \neq y^{i_n} \wedge \psi) \ldots),$$
$$\xi_r := (x^{i_1} \neq y^{i_1} \vee^{i_1} (x^{i_1} = y^{i_1} \wedge (x^{i_2} \neq y^{i_2} \vee^{i_2} (x^{i_2} = y^{i_2}$$
$$\wedge (\ldots \wedge (x^{i_n} \neq y^{i_n} \vee^{i_n} (x^{i_n} = y^{i_n} \wedge \theta) \ldots).$$

The idea above is that the variables x^{i_j}, y^{i_j} are used to encode a (possibly overlapping) split of the team X_j. Using locality it is easy to see that ϕ and ϕ^+ are equivalent over structures of cardinality at least two. From this the claim follows in a straightforward manner.

Global case. We show how a single local disjunction is eliminated. From this the result follows. Again, without loss of generality, we restrict to structures of cardinality at least two. Let $\phi = \psi \vee^j \theta$ be a formula of PFO and let I list the sorts of all the variables that occur in ϕ except j. Define ϕ^* as

$$\forall_{i \in I} x^i y^i \big((\psi \wedge \bigwedge_{i \in I} x^i = y^i) \vee (\theta \wedge \bigwedge_{i \in I} x^i \neq y^i) \big),$$

where, for each $i \in I$, the variables x^i and y^i are fresh and distinct. The idea in ϕ^* is that on structures with at least two elements, the evaluation of the universal quantifiers $\forall x^i y^i$ on a polyteam $\overline{X}$ duplicates each assignment in X_i in at least two ways: in some duplicates $x^i = y^i$ holds and in others $x^i \neq y^i$ holds. It then follows from locality that ϕ^* and ϕ are equivalent. $\qquad\square$

We wish to point out that, in contrast to the previous result, it is easy to see that there is no single formula in PFO without local (global, reps.) disjunction that defines global (local, resp.) disjunction. Moreover it's worth noticing that the translations produced above are quite involved and rely on the use of quantifiers. It is easy to define natural fragments of PFO where the disjunctions cannot be expressed with another.

3.3 Data exchange in the polyteam setting

As promised, we now return to the topic of modelling data exchange in our new setting. In this section we restrict our attention to poly-atoms that are embedded dependencies. Our first goal is to define the notions of *source-to-target* and *target* poly-atoms. For this purpose we define a normal form for embedded dependencies. We call an embedded dependency $\forall \overline{x} \big(\phi(\overline{x}) \to \exists \overline{y} \psi(\overline{x}, \overline{y}) \big)$ *separated* if the relation symbols that occur in ϕ and ψ are distinct. A poly-atom is called *separated*, if the defining formula is a separated embedded dependency. In the polyteam setting this is just a technical restriction as non-separated poly-atoms can be always simulated by separated ones. Below we use the syntax $A(\overline{x}_1, \ldots, \overline{x}_l, \overline{y}_1, \ldots, \overline{y}_k)$ for separated poly-atoms. The idea is that $\overline{x}_i$s project extensions for relations used in the antecedent and $\overline{y}_j$s in the consequent of the defining formula.

Let $\mathcal{S}$ and $\mathcal{T}$ be a set of source relations and target relations from some data exchange instance, respectively. Let $\overline{X} = (S_1, \ldots S_n, T_1, \ldots, T_m)$ be a polyteam that encodes $\mathcal{S}$ and $\mathcal{T}$ in the obvious manner. We say that an instance of a separated atom $A(\overline{x}_1, \ldots, \overline{x}_l, \overline{y}_1, \ldots, \overline{y}_k)$ is *source-to-target* if each $\overline{x}_i$ is a tuple of variables of the sort of S_j, for some j, and each $\overline{y}_i$ is a tuple of variables of the sort of T_j, for some j. Analogously the instance $A(\overline{x}_1, \ldots, \overline{x}_l, \overline{y}_1, \ldots, \overline{y}_k)$ is *target* if each $\overline{x}_i$ and $\overline{y}_j$ is a tuple of variables of the sort of T_p for some p.

Data exchange problems can now be directly studied in the polyteam setting. For example the *existence-of-solution* problem can be reduced to a model checking problem by using first-order quantifiers to *guess* a solution for the problem while the rest of the formula describes the dependences required to be fulfilled in the data exchange problem.

Example 16. A relational database schemas

$$\begin{aligned} \mathcal{S}: \quad & \text{P(\textsc{rojects})} = \{\texttt{name, employee, employee\_position}\}, \\ \mathcal{T}: \quad & \text{E(\textsc{mployees})} = \{\texttt{name, project\_1, project\_2}\} \end{aligned}$$

are used to store information about employees positions in different projects. We wish to check whether for a given instance of the schema $\mathcal{S}$ there exists an instance of the schema $\mathcal{T}$ that does not lose any information about for which projects employees are tasked to work and that uses the attribute `name` as a key. The $\mathsf{PFO}(\mathrm{pinc}, \mathrm{dep})$-formula

$$\phi := \exists x_1 \exists x_2 \exists x_3 \Big(\big(\mathrm{P}[\texttt{employee, name}] \subseteq \mathrm{E}[x_1, x_2]$$

$$\vee^{\mathrm{P}} \mathrm{P}[\texttt{employee, name}] \subseteq \mathrm{E}[x_1, x_3] \big) \wedge = (x_1, (x_2, x_3)) \Big),$$

when evaluated on a polyteam that encodes an instance of the schema $\mathcal{S}$, expresses that a solution for the data exchange problem exists. The variables x_1, x_2, and x_3 above are of sort E and are used to encode attribute names `name, project_1`, and `project_2`, respectively. The dependence atom above enforces that the attribute `name` is a key.

4 Relationship between polyteam and team semantics

Before embarking on an analysis of the expressive power of polyteam logics, we consider the relationship between polyteam and team semantics. One might ask is polyteam semantics really necessary in order to model the polyrelational case. Is it not possible to embed and interpret polyteams within team semantics? We will next shed some light on this question. Independence logic $\mathsf{FO}(\mathrm{ind})$ is as expressive as existential second-order logic ESO when the team is encoded as a relation [7]. Moreover it is relatively clear that $\mathsf{PFO}(\mathrm{pind})$ translates to ESO when the polyteam is encoded as a tuple of relations. Furthermore in ESO multiple relations can be encoded into a single relation of large enough arity. Theorem 25 establishes that, analogously to the case with ESO, $\mathsf{PFO}(\mathrm{pind})$ can also be simulated using $\mathsf{FO}(\mathrm{ind})$. On the other hand, such a result does not seem likely to hold for $\mathsf{PFO}(\mathrm{pdep})$ and $\mathsf{FO}(\mathrm{dep})$. It seems that there is no suitable way to encode multiple relations into a single relation that is also compatible with downward closure; it is known that $\mathsf{FO}(\mathrm{dep})$ characterises downward closed ESO-definable properties of teams [22]. Section 4.1 lays out the groundwork for relating polyteam semantics to team semantics. Our results will be shown in Section 4.2.

4.1 Team representation

First let us formulate precisely how polyteams can be represented by teams. Let X be a non-empty team whose domain takes variables from $\mathrm{Var}(1), \ldots, \mathrm{Var}(n)$. Then X *represents* the polyteam $\overline{X} = (X_1, \ldots, X_n)$ where $X_i = X \upharpoonright \mathrm{Var}(i)$. Note that any strictly non-empty polyteam $(X_1, \ldots, X_n)$, where $\bigcup \mathsf{Dom}(X_i)$ is finite, can be represented by a team. In order to deal with the situations where some coordinate team X_i of $\overline{X}$ is empty, we next introduce the concepts of *contraction* and *emptification*.

Definition 17 (Contraction and emptification). *Let Q be a generalised quantifier of type $(j_1, \ldots, j_n)$. The generalised quantifier Q' of type $(j_1, \ldots, j_{i-1}, j_{i+1}, j_n)$ is the i-contraction of Q if the equivalence*

$$(A, R_1, \ldots, R_{i-1}, \emptyset, R_{i+1}, R_n) \in Q \Leftrightarrow (A, R_1, \ldots, R_{i-1}, R_{i+1}, R_n) \in Q'$$

holds for every $A, R_1, \ldots, R_n$. For $I \subseteq \{1, \ldots, n\}$, the I-contraction of Q is defined analogously. We say that a class of generalised poly-atoms $\mathcal{C}$ is closed under contraction if, for any poly-atom $\alpha_Q \in \mathcal{C}$ and any index i, $A_{Q'} \in \mathcal{C}$, where Q' is the i-contraction of Q.

Let $\phi = \alpha_Q(\overline{x}_1^{j_1}, \ldots, \overline{x}_n^{j_n})$ be a poly-atom, $i \in \mathbb{N}$ a natural number, $I = \{k \mid 1 \leq k \leq n, j_k = i\}$ a set of indices, and Q' the I-contraction of Q. Furthermore let $\overline{\chi}$ list the variable

tuples in $\overline{x}_1^{j_1}, \ldots, \overline{x}_n^{j_n}$ that are not of sort i. We let $\phi_{i \mapsto \emptyset}$ denote the poly-atom $\alpha_{Q'}(\overline{\chi})$, and call it the i-emptification of ϕ.

For instance, the 1-emptification of the dependence atom $= (x^1, y^1)$ is equivalent to $\top$. It is easy to see that in polyteam setting, the i-emptification of $\alpha(\overline{x})$ can be equivalently expressed with a $\mathsf{PFO}(\alpha)$-formula. However the same is not, in general, true in the team semantics setting.

Next we show how statements on polyteams that have empty coordinate teams can be transformed to equivalent statements over strictly non-empty polyteams.

Definition 18 (Emptification of complex formulae). *Let $\mathcal{C}$ be a set of poly-atoms, ϕ a formula of $\mathsf{PFO}(\mathcal{C})$, and $i \in \mathbb{N}$. We denote by $\phi_{i \mapsto \emptyset}$ the formula obtained from ϕ by simultaneously replacing all first-order literals of sort i with $\top$, all disjunctions $\vee^i$ with $\wedge$, and all poly-atoms α with $\alpha_{i \mapsto \emptyset}$, and by eliminating all quantifiers of sort i. We call $\phi_{i \mapsto \emptyset}$ the i-emptification of ϕ. Note that $\phi_{i \mapsto \emptyset}$ is free of variables and local disjuctions of sort i.*

The following proposition can now be shown by straightforward structural induction.

Proposition 19. *Let $\phi \in \mathsf{PFO}(\mathcal{C})$, where $\mathcal{C}$ is any set of poly-atoms. Then for all models $\mathfrak{A}$ and polyteams $\overline{X}$,*

$$\mathfrak{A} \models_{\overline{X}} \phi_{i \mapsto \emptyset} \iff \mathfrak{A} \models_{\overline{X}_{i=\emptyset}} \phi,$$

where $\overline{X}_{i=\emptyset}$ is obtained from $\overline{X}$ by substituting the empty team for X_i.

Next we introduce a trick that enables us to change a representation of a polyteam on the fly.

Lemma 20. *Let $\mathcal{C}$ be a set of poly-atoms and X a non-empty team representing $\overline{X} = (X_1, \ldots, X_n)$. For every $\phi \in \mathsf{FO}(\mathcal{C}, \mathrm{inc})$ there exists a formula $\phi^{\exists r} \in \mathsf{FO}(\mathcal{C}, \mathrm{inc})$ such that*

$$\mathfrak{A} \models_X \phi^{\exists r} \Leftrightarrow \mathfrak{A} \models_Y \phi, \text{ for some team representation } Y \text{ of } \overline{X}.$$

Proof. Let ϕ, $\overline{X}$, and X be as described above. For each $1 \leq i \leq n$, let $\overline{x}^i = x_1^i, \ldots, x_k^i$ be the variable domain of X_i and let $\overline{y}^i = y_1^i, \ldots, y_k^i$ be fresh and distinct variables. Let $\overline{x}$ and $\overline{y}$ denote $\overline{x}^1, \ldots, \overline{x}^n$ and $\overline{y}^1, \ldots, \overline{y}^n$, respectively. Define

$$\phi^{\exists r} := \exists \overline{y}\Big(\bigwedge_{1 \leq i \leq n} (\overline{x}^i \subseteq \overline{y}^i \wedge \overline{y}^i \subseteq \overline{x}^i) \wedge \exists \overline{x}(\overline{x} \subseteq \overline{y} \wedge \overline{y} \subseteq \overline{x} \wedge \phi)\Big).$$

It is easy to check, using locality, that $\phi^{\exists r}$ defined as above satisfies the claim of the lemma. $\square$

4.2 Translation to team semantics

Using the concepts and results from the previous section we will now move to translations from polyteam semantics to team semantics. The next theorem reveals that polyteam semantics can be simulated with team semantics using inclusion atoms and *classical disjunction* $\varovee$:

$$\mathfrak{A} \models_X \phi \varovee \psi \text{ iff } \mathfrak{A} \models_X \phi \text{ or } \mathfrak{A} \models_X \psi.$$

Theorem 21. *Let ϕ be a formula in $\mathsf{PFO}(\mathcal{C})$, where $\mathcal{C}$ is a contraction closed set of poly-atoms. Then there is a formula $\phi^* \in \mathsf{FO}(\mathcal{C}, \mathrm{inc}, \varovee)$ such that for any structure $\mathfrak{A}$, with at least two elements, and any non-empty team X representing $\overline{X} = (X_1, \ldots, X_n)$,*

$$\mathfrak{A} \models_{\overline{X}} \phi \Leftrightarrow \mathfrak{A}^* \models_X \phi^*,$$

where $\mathfrak{A}^*$ is an expansion of $\mathfrak{A}$ with two distinct constants.

Proof. Define a rank r of a formula ϕ as follows. For a poly-atom $\alpha(\overline{x}_1^{i_1}, \ldots, \overline{x}_n^{i_n})$, $\mathrm{r}(\alpha) := n$. For first-order literals α, $\mathrm{r}(\alpha) := 1$. For complex formulae, $\mathrm{r}(Q\psi) := \mathrm{r}(\psi) + 1$ and $\mathrm{r}(\psi C\theta) := \mathrm{r}(\psi) + \mathrm{r}(\theta) + 1$, where $Q \in \{\exists, \forall\}$ and $C \in \{\wedge, \vee, \vee^i\}$. We prove the claim by induction on $\mathrm{r}(\phi)$ for a mapping $\phi \mapsto \phi^*$ defined recursively. This mapping is the identity on literals, and homomorphism on conjunction and universal quantification; proving the induction claim is straightforward for these cases and thus omitted. By Proposition 15 we can also exclude the case of global disjunction. Thus it suffices to consider local disjunction and existential quantification.

Local disjunction. Suppose $\phi = \psi \vee^i \theta$. We define

$$\phi^* := ((\psi_{i\mapsto\emptyset})^* \wedge \theta^*) \oslash (\psi^* \wedge (\theta_{i\mapsto\emptyset})^*) \oslash \Phi^{\exists r}, \tag{2}$$

where $^{\exists r}$ is as defined in Lemma 20 and

$$\Phi := \exists z \left(\overline{x}0 \subseteq \overline{x}z \wedge \overline{x}1 \subseteq \overline{x}z \wedge \left((\psi^* \wedge z = 0) \vee (\theta^* \wedge z = 1) \right) \right), \tag{3}$$

where z is a fresh variable and $\overline{x}$ lists all the free variables of ϕ which are not of the sort i. Furthermore, 0 and 1 are the two distinct constants, and $\overline{x}0 \subseteq \overline{y}z$ and $\overline{x}0 \subseteq \overline{y}z$ are shorthands for $\exists v(v = 0 \wedge \overline{x}v \subseteq \overline{y}z)$ and $\exists v(v = 1 \wedge \overline{x}v \subseteq \overline{y}z)$, respectively.

Assume first that $\mathfrak{A} \models_{\overline{X}} \phi$. Then there are $Y_i \cup Z_i = X_i$ such that $\mathfrak{A} \models_{\overline{Y}} \psi$ and $\mathfrak{A} \models_{\overline{Z}} \theta$, where $\overline{Y}$ and $\overline{Z}$ are obtained from $\overline{X}$ by substituting respectively Y_i and Z_i for X_i. Suppose Y_i is empty. Then by Proposition 19 $\mathfrak{A} \models_{\overline{X}} \psi_{i\mapsto\emptyset}$. Thus by induction hypothesis $\mathfrak{A} \models_X (\psi_{i\mapsto\emptyset})^*$. Furthermore $\mathfrak{A} \models_X \theta^*$ follows from $\mathfrak{A} \models_{\overline{Z}} \theta$ by the induction hypothesis, for $\overline{Z} = \overline{X}$ since $Y_i = \emptyset$. Thus the first disjunct of ϕ^* holds. If Z_i is empty, we similarly obtain that the second disjunct holds.

Suppose then that both Y_i and Z_i are non-empty. Let X^* be a representation of $\overline{X}$ obtained by taking the Cartesian product of the teams $X_1, \ldots, X_n$. For $s \in X^*$, define

$$G(s) := \begin{cases} \{0, 1\} & \text{if } s \upharpoonright \mathrm{Var}(i) \in Y_i \cap Z_i \\ \{0\} & \text{if } s \upharpoonright \mathrm{Var}(i) \in Y_i \setminus Z_i \\ \{1\} & \text{if } s \upharpoonright \mathrm{Var}(i) \in Z_i \setminus Y_i. \end{cases}$$

Clearly, the two left-most conjuncts in (3) are satisfied by $X^*[G/z]$. Recall that $\overline{Y}$ and $\overline{Z}$ were obtained from $\overline{X}$ by substituting respectively Y_i and Z_i for X_i. Let Y' and Z' consist of those assignments of $X^*[G/z]$ whose restriction to $\mathrm{Var}(i)$ belongs to Y_i and Z_i, respectively. Note that if the variable z is disregarded from Y' and Z', team representations of $\overline{Y}$ and $\overline{Z}$ are obtained, respectively. Now by induction hypothesis, locality (Proposition 11), and the selection of G,

$$\mathfrak{A} \models_{\overline{Y}} \psi \Leftrightarrow \mathfrak{A} \models_{Y'} \psi^* \Leftrightarrow \mathfrak{A} \models_{Y'} \psi^* \wedge z = 0. \tag{4}$$

Similarly we obtain that $\mathfrak{A} \models_{Z'} \theta^* \wedge z = 1$. Since $Y' \cup Z' = X^*[G/z]$, we conclude that $\mathfrak{A} \models_{X^*} \Phi$. Finally, by Lemma 20, we obtain that $\mathfrak{A} \models_X \Phi^{\exists r}$ and hence that $\mathfrak{A} \models_X \phi^*$.

For the converse direction, suppose $\mathfrak{A} \models_X \phi^*$. Assume first that $\mathfrak{A} \models_X (\psi_{i\mapsto\emptyset})^* \wedge \theta^*$. By the induction hypothesis, it follows that $\mathfrak{A} \models_{\overline{X}} \theta$ and $\mathfrak{A} \models_{\overline{X}} \psi_{i\mapsto\emptyset}$. By Proposition 19 together with the semantics of local disjunctions, it follows that $\mathfrak{A} \models_{\overline{X}} \phi$. Similarly, $\mathfrak{A} \models_{\overline{X}} \phi$ if $\mathfrak{A} \models_X (\theta_{i\mapsto\emptyset})^* \wedge \psi^*$. Assume then that $\mathfrak{A} \models_X \Phi^{\exists r}$. Then $\mathfrak{A} \models_Y \Phi$ for some team Y representing $\overline{X}$. Let $F \colon Y \to \mathcal{P}(A) \setminus \{\emptyset\}$ be a mapping such that $Y[F/z]$ satisfies the quantifier-free part of (3), and let U and V consist of those assignments of $Y[F/z]$ that map z to 0 and 1, respectively. The disjunction in (3) now guarantees that $U \cup V = Y[F/z]$, $\mathfrak{A} \models_U \psi^*$, and $\mathfrak{A} \models_V \theta^*$. By the induction hypothesis, $\mathfrak{A} \models_{\overline{U}} \psi$ and $\mathfrak{A} \models_{\overline{V}} \theta$, where $\overline{U} = (U_1, \ldots, U_n)$ and

$\overline{V} = (V_1, \ldots, V_n)$ are the polyteams represented by U and V, respectively. Since X and Y represent the same polyteam, and $Y(\overline{x}) = U(\overline{x}) = V(\overline{x})$ by the inclusion atoms in (3), we obtain that $U_j = V_j = X_j$ for $j \neq i$. Moreover, $U_i \cup V_i = Y_i = X_i$, and thus we obtain that $\mathfrak{A} \models_{\overline{X}} \psi \vee^i \theta$.

Existential quantification. Suppose $\phi = \exists x \psi$ where $x \in \mathrm{Var}(i)$. We define $\phi^* := \exists x \psi^*$. Suppose first $\mathfrak{A} \models_{\overline{X}} \phi$. Then $\mathfrak{A} \models_{\overline{X}[X_i[F_i/x]/X_i]} \psi$ for some $F_i \colon X_i \to \mathcal{P}(A) \setminus \{\emptyset\}$. If $F \colon X \to \mathcal{P}(A) \setminus \{\emptyset\}$ is defined as $F(s) := F_i(s \upharpoonright \mathrm{Var}(i))$, it follows that the team $X[F/x]$ represents the polyteam $\overline{X}[X_i[F_i/x]/X_i]$. Hence by induction hypothesis $\mathfrak{A} \models_{X[F/x]} \psi^*$, and thus $\mathfrak{A} \models_X \phi^*$.

For the converse direction, suppose $\mathfrak{A} \models_X \phi^*$. Then we find $F \colon X \to \mathcal{P}(A) \setminus \{\emptyset\}$ such that $\mathfrak{A} \models_{X[F/x]} \psi^*$. Setting $F_i \colon X_i \to \mathcal{P}(A) \setminus \{\emptyset\}$ as $F_i(s) := \bigcup\{F(s') \mid s' \in X, s' \upharpoonright \mathrm{Var}(i) = s\}$, it follows that $\overline{X}[X_i[F_i/x]/X_i]$ is represented by $X[F/x]$. Hence, we obtain by the induction hypothesis, that $\mathfrak{A} \models_{\overline{X}[X_i[F_i/x]/X_i]} \psi$. Thus $\mathfrak{A} \models_{\overline{X}} \phi$. $\qquad\square$

For poly-inclusion logic, the following corollary now follows immediately.

Corollary 22. *Let ϕ be a formula in* $\mathsf{PFO}(\mathrm{pinc})$. *Then there is a formula $\phi^* \in \mathsf{FO}(\mathrm{inc}, \varolessthan)$ such that, for any structure $\mathfrak{A}$ with at least two elements, and any non-empty team X representing $\overline{X}$,*

$$\mathfrak{A} \models_{\overline{X}} \phi \Leftrightarrow \mathfrak{A}^* \models_X \phi^*,$$

where $\mathfrak{A}^*$ is an expansion of $\mathfrak{A}$ with two distinct constants.

Proof. By the previous theorem ϕ^* can be found from $\mathsf{FO}(\mathcal{C}, \mathrm{inc}, \varolessthan)$, where $\mathcal{C}$ is the closure of poly-inclusion atoms under contraction. Since poly-inclusion atoms are just inclusion atoms in team semantics, and the closure only adds poly-atoms equivalent to $\top$ or $\bot$, we obtain that $\phi^* \in \mathsf{FO}(\mathrm{inc}, \varolessthan)$. $\qquad\square$

Next we show that the polyteam logics defined in terms of ESO-definable poly-atoms can be represented in independence logic. First, we observe that ESO-definable poly-atoms are closed under contraction.

Proposition 23. *The set of ESO-definable poly-atoms is closed under contraction.*

Proof. Let α_Q be a generalised ESO-definable poly-atom of type $(j_1, \ldots, j_n)$, and let $i \in \{1, \ldots, n\}$. We show that the i-contraction of α_Q is definable in ESO. Without loss of generality $i = n$, in which case the i-contraction of α_Q, written $\alpha_{Q'}$, is of type of $(j_1, \ldots, j_{n-1})$. Let $\phi(R_1, \ldots, R_n)$ be an ESO formula which defines α_Q, and let $\phi'(R_1, \ldots, R_{n-1})$ be obtained from ϕ by replacing all relational atoms of the form $R_n(\overline{t})$ with $\bot$. Then for all models $\mathfrak{A} = (A, R_1^{\mathfrak{A}}, \ldots, R_{n-1}^{\mathfrak{A}})$,

$$(A, R_1^{\mathfrak{A}}, \ldots, R_{n-1}^{\mathfrak{A}}) \in Q' \; \Leftrightarrow \; (A, R_1^{\mathfrak{A}}, \ldots, R_{n-1}^{\mathfrak{A}}, \emptyset) \in Q$$
$$\Leftrightarrow (A, R_1^{\mathfrak{A}}, \ldots, R_{n-1}^{\mathfrak{A}}, \emptyset) \models \phi \; \Leftrightarrow \; (A, R_1^{\mathfrak{A}}, \ldots, R_{n-1}^{\mathfrak{A}}) \models \phi'.$$

$\qquad\square$

Second, we use the fact that $\mathsf{FO}(\mathrm{ind})$ characterises all ESO-definable team properties. Note that $\mathrm{rel}(X)$ refers to a relation $\{s(x_1, \ldots, x_n) \mid s \in X\}$ where $x_1, \ldots, x_n$ is some enumeration of $\mathsf{Dom}(X)$.

Theorem 24 ([7,22]). *Let $\phi(\overline{x})$ be an $\mathsf{FO}(\mathrm{ind})$ ($\mathsf{FO}(\mathrm{dep})$, resp.) formula, and let R be an $|\overline{x}|$-ary relation. Then there is an (downwards closed with respect to R, resp.) ESO-sentence $\psi(R)$ such that for all teams $X \neq \emptyset$ where $\mathsf{Dom}(X) = \overline{x}$,*

$$\mathfrak{A} \models_X \phi(\overline{x}) \Leftrightarrow (\mathfrak{A}, R := \mathrm{rel}(X)) \models \psi(R)$$

The same statement holds also vice versa.

In fact, in the above theorem we may substitute all ESO-definable poly-atoms and classical disjunction for independence atoms.

Theorem 25. *Let $\mathcal{C}$ be a set of ESO-definable poly-atoms, and let ϕ be a formula in $\mathsf{PFO}(\mathcal{C})$. Then there is a formula $\phi^* \in \mathsf{FO}(\mathrm{ind})$ such that for any structure $\mathfrak{A}$ and any non-empty team X representing $\overline{X}$,*

$$\mathfrak{A} \models_{\overline{X}} \phi \Leftrightarrow \mathfrak{A} \models_X \phi^*.$$

Proof. We may assume that $\mathcal{C}$ is the set of all ESO-definable poly-atoms. Let ϕ be an arbitrary formula of $\mathsf{PFO}(\mathcal{C})$. Without loss of generality, it suffices to prove the above equivalence with respect to structures that have at least two elements. By Theorem 21 and Proposition 23, we find a formula $\phi' \in \mathsf{FO}(\mathcal{C}, \mathrm{inc}, \varovee)$ such that, for every structure $\mathfrak{A}$, with at least two elements, and team X that represents some non-empty polyteam $\overline{X}$ of ϕ, it holds that

$$\mathfrak{A} \models_{\overline{X}} \phi \Leftrightarrow \mathfrak{A}' \models_X \phi',$$

where $\mathfrak{A}'$ is an expansion of $\mathfrak{A}$ with two distinct constants 0 and 1. Define the formula $\phi'' := \exists xy \big(=(x) \wedge =(y) \wedge x \neq y \wedge \phi'(x/0, y/1) \big)$, where x and y are fresh distinct variables that do not occur in ϕ', and $\phi'(x/0, y/1)$ is obtained from ϕ' by replacing 0 and 1 by x and y, respectively. Clearly

$$\mathfrak{A}' \models_X \phi' \Leftrightarrow \mathfrak{A} \models_X \phi''.$$

Note that $\phi'' \in \mathsf{FO}(\mathcal{C}, \varovee)$, for dep and inc are ESO-definable poly-atoms. Also, when restricted to non-empty teams, $\mathsf{FO}(\mathcal{C}, \varovee) \leq \mathsf{FO}(\mathrm{ind})$, for $\mathsf{FO}(\mathcal{C}) \equiv \mathsf{FO}(\mathrm{ind})$ by Proposition 13, and $\mathsf{FO}(\mathrm{ind})$ is closed under classical disjunction because it inherits this property from ESO by Theorem 24. Thus we conclude that, by taking a formula ϕ^* of $\mathsf{FO}(\mathrm{ind})$ which is equivalent to ϕ'' with respect to non-empty teams, we obtain the required team representation of ϕ. $\square$

As an immediate corollary we obtain that poly-independence logic is representable in independence logic.

Corollary 26. *Let ϕ be a formula in $\mathsf{PFO}(\mathrm{pind})$. Then there is a formula $\phi^* \in \mathsf{FO}(\mathrm{ind})$ such that for any structure $\mathfrak{A}$ and any non-empty team X representing $\overline{X}$,*

$$\mathfrak{A} \models_{\overline{X}} \phi \Leftrightarrow \mathfrak{A} \models_X \phi^*.$$

5   Expressiveness

The expressiveness properties of dependence, independence, inclusion, and exclusion logic and their fragments enjoy already comprehensive classifications. Dependence logic and exclusion logic are equi-expressive and capture all downwards closed ESO properties of teams [7,22]. Independence logic, whose independence atoms violate downward closure, in turn captures all ESO team properties [7]. On the other hand, the expressivity of inclusion logic has been characterised by the so-called greatest fixed point logic [9]. In this section we turn attention to polyteams and consider the expressivity of the poly-dependence logics introduced in this paper. Section 5.1 deals with logics with only uni-dependencies whereas in Section 5.2 poly-dependencies are considered.

5.1 Uni-dependencies in polyteam semantics

First we turn attention to uni-atoms in polyteam semantics. We show that with uni-atoms no interaction between different relations is possible.

Theorem 27. *Let $\mathcal{C}$ be a set of uni-atoms. Each formula $\phi(\overline{x}^1, \ldots, \overline{x}^n) \in \mathsf{PFO}(\mathcal{C})$ can be associated with a sequence of formulae $\psi_1(\overline{x}^1), \ldots, \psi_n(\overline{x}^n) \in \mathsf{FO}(\mathcal{C})$ such that for all structures $\mathfrak{A}$ and all $\overline{X} = (X_1, \ldots, X_n)$, where X_i is a team with domain $\overline{x}^i$,*

$$\mathfrak{A} \models_{\overline{X}} \phi(\overline{x}^1, \ldots, \overline{x}^n) \Leftrightarrow \forall i = 1, \ldots, n : \mathfrak{A} \models_{X_i} \psi_i(\overline{x}^i).$$

Similarly, the statement holds vice versa.

Proof. The latter statement is clear as it suffices to set $\phi(\overline{x}^1, \ldots, \overline{x}^n) := \psi_1(\overline{x}^1) \wedge \ldots \wedge \psi_n(\overline{x}^n)$. For the other direction, we define recursively functions f_i that map formulae $\phi(\overline{x}^1, \ldots, \overline{x}^n) \in \mathsf{PFO}(\mathcal{C})$ to formulae $\psi_i(\overline{x}^i) \in \mathsf{FO}(\mathcal{C})$. By Proposition 15 we may assume that only disjunctions of type $\vee^i$, for some $i \in \mathbb{N}$, may occur in ϕ. The functions f_i are defined as follows:

- If $\phi(\overline{x}^j)$ is an atom, then $f_i(\phi) := \begin{cases} \phi & \text{if } i = j, \\ \top & \text{otherwise.} \end{cases}$

- $f_i(\psi \vee^j \theta) := \begin{cases} f_i(\psi) \vee f_i(\theta) & \text{if } i = j, \\ f_i(\psi) \wedge f_i(\theta) & \text{otherwise.} \end{cases}$

- $f_i(\psi \wedge \theta) := f_i(\psi) \wedge f_i(\theta)$.

- For $Q \in \{\exists, \forall\}$, set $f_i(Qx^j\psi) := \begin{cases} Qx f_i(\psi) & \text{if } i = j, \\ f_i(\psi) & \text{otherwise.} \end{cases}$

We set $\psi_i := f_i(\phi)$ and show the claim by induction on the structure of the formula. The cases for atoms and conjunctions are trivial. We show the case for $\vee^i$.

Let $\phi = \psi \vee^j \theta$ and assume that the claim holds for ψ and θ. Now

$$\mathfrak{A} \models_{\overline{X}} \phi \quad \text{iff} \quad \mathfrak{A} \models_{\overline{X}[Y_j/X_j]} \psi \text{ and } \mathfrak{A} \models_{\overline{X}[Z_j/X_j]} \theta,$$
$$\text{for some } Y_j, Z_j \subseteq X_j \text{ such that } Y_j \cup Z_j = X_j.$$

By the induction hypothesis, $\mathfrak{A} \models_{\overline{X}[Y_j/X_j]} \psi$ and $\mathfrak{A} \models_{\overline{X}[Z_j/X_j]} \theta$ iff $\mathfrak{A} \models_{Y_j} f_j(\psi), \mathfrak{A} \models_{Z_j} f_j(\theta)$, and $\mathfrak{A} \models_{X_i} f_i(\psi), \mathfrak{A} \models_{X_i} f_i(\theta)$ for each $i \neq j$. Thus we obtain that $\mathfrak{A} \models_{\overline{X}} \phi$ holds iff

$$\mathfrak{A} \models_{X_j} f_j(\psi) \vee f_j(\theta), \text{ and } \mathfrak{A} \models_{X_i} f_i(\psi) \text{ and } \mathfrak{A} \models_{X_i} f_i(\theta) \text{ for each } i \neq j.$$

The above can be rewritten as

$$\mathfrak{A} \models_{X_j} f_j(\psi) \vee f_j(\theta), \text{ and } \mathfrak{A} \models_{X_i} f_i(\psi) \wedge f_i(\theta) \text{ for each } i \neq j.$$

The claim now follows, since $f_j(\psi) \vee f_j(\theta) = f_j(\psi \vee^j \theta)$ and $f_i(\psi) \wedge f_i(\theta) = f_i(\psi \vee^j \theta)$, for $i \neq j$.

The cases for the quantifiers are similar. $\qquad\square$

Theorem 27 implies that poly-atoms which describe relations between two teams are beyond the scope of uni-logics. The following proposition illustrates this for $\mathsf{PFO}(\mathsf{dep})$.

Proposition 28. *The poly-constancy atom $=(x^1/x^2)$ cannot be expressed in $\mathsf{PFO}(\mathsf{dep})$.*

Proof. Assume that $=(x^1/x^2)$ can be defined by some $\phi(x^1, x^2) \in \mathsf{PFO}(\mathsf{dep})$. By Theorem 27 there are $\mathsf{FO}(\mathsf{dep})$-formulae $\psi_1(x^1)$ and $\psi_2(x^2)$ such that for all $\overline{X} = (X_1, X_2)$, where X_i is a team with domain x^i, it holds that

$$\mathfrak{A} \models_{\overline{X}} = (x^1/x^2) \Leftrightarrow \forall i = 1, 2 : \mathfrak{A} \models_{X_i} \psi_i(x^i). \tag{5}$$

Define teams $X_1 := \{x^1 \mapsto 0\}$, $X_2 := \{x^2 \mapsto 0\}$, $Y_1 := \{x^1 \mapsto 1\}$, and $Y_2 := \{x^2 \mapsto 1\}$. Now clearly $\mathfrak{A} \models_{(X_1, X_2)} = (x^1/x^2)$, and $\mathfrak{A} \models_{(Y_1, Y_2)} = (x^1/x^2)$. Hence by (5), we obtain first that $\mathfrak{A} \models_{X_1} \psi_i(x^1)$ and $\mathfrak{A} \models_{Y_2} \psi_i(x^2)$, and then that $\mathfrak{A} \models_{(X_1, Y_2)} = (x^1/x^2)$, which is a contradiction. □

It is now easy to see that Theorems 27 and 24 together imply that $\mathsf{PFO}(\mathsf{ind})$ ($\mathsf{PFO}(\mathsf{dep})$, resp.) captures all conjunctions of (downward closed, resp.) ESO properties of teams.

Theorem 29. *Let* $\phi(\overline{x}^1, \ldots, \overline{x}^n)$ *be a* $\mathsf{PFO}(\mathsf{ind})$ *(*$\mathsf{PFO}(\mathsf{dep})$*, resp.) formula where* $\overline{x}^i$ *is a sequence of variables from* $\mathrm{Var}(i)$. *Let* R_i *be an* $|\overline{x}^i|$*-ary relation symbol for* $i = 1, \ldots, n$. *Then there are (downwards closed with respect to* R_i*, resp.)* ESO*-sentences* $\psi_1(R_1), \ldots, \psi_n(R_n)$ *such that for all polyteams* $\overline{X} = (X_1, \ldots, X_n)$ *where* $\mathsf{Dom}(X_i) = \overline{x}^i$ *and* $X_i \neq \emptyset$

$$\mathfrak{A} \models_{\overline{X}} \phi(\overline{x}^1, \ldots, \overline{x}^n) \Leftrightarrow (\mathfrak{A}, R_1 := \mathrm{rel}(X_1), \ldots, R_n := \mathrm{rel}(X_n)) \models \psi_1(R_1) \wedge \ldots \wedge \psi_n(R_n).$$

The same statement holds also vice versa.

5.2 Poly-dependencies in polyteam semantics

Next we consider poly-dependencies in polyteam semantics. We begin by observing that many translations between different team logics carry over to polyteam logics.

Lemma 30. *The following equivalences hold:*

$$= (\overline{x}^1, \overline{y}^1/\overline{u}^2, \overline{v}^2) \equiv \overline{y}^1/\overline{y}^1 \perp_{\overline{x}^1, \overline{u}^2/\overline{x}^1} \overline{v}^2/\overline{y}^1, \tag{6}$$

$$= (\overline{x}^1, y^1/\overline{u}^2, v^2) \equiv \forall z^1 (y^1 = z^1 \vee^1 \overline{x}^1 z^1 \mid \overline{u}^2 v^2), \tag{7}$$

$$\overline{x}^1 \subseteq \overline{u}^2 \equiv \overline{x}^1/\overline{u}^2 \perp \emptyset/\emptyset, \tag{8}$$

$$\overline{x}^1 \subseteq \overline{u}^2 \equiv \forall \overline{v}^2 (\overline{x}^1 \mid \overline{v}^2 \vee^2 \overline{v}^2 \subseteq \overline{u}^2), \tag{9}$$

$$\overline{x}^1 \mid \overline{u}^2 \equiv \exists y^1 z^1 v^2 w^2 (= (\overline{x}^1, y^1 z^1/\overline{u}^2, v^2 w^2) \tag{10}$$
$$\wedge y^1 = z^1 \wedge v^2 \neq w^2),$$

$$\overline{x}^1 \mid \overline{u}^2 \equiv \exists \overline{y}^1 (\overline{u}^2 \subseteq \overline{y}^1 \wedge \overline{x}^1 \mid \overline{y}^1), \tag{11}$$

$$\overline{y}^2/\overline{y}^1 \perp_{\overline{x}^2, \overline{x}^3/\overline{x}^1} \overline{z}^3/\overline{z}^1 \equiv \forall \overline{p}^2 \overline{q}^2 \exists u^2 v^2 \forall \overline{p}^3 \overline{q}^3 \overline{r}^3 \exists u^3 v^3 \Big(\tag{12}$$
$$= (\overline{p}^2 \overline{q}^2, u^2 v^2/\overline{p}^3 \overline{q}^3, u^3 v^3)$$
$$\wedge (u^2 = v^2 \vee^1 (u^2 \neq v^2 \wedge \overline{x}^2 \overline{y}^2 \mid \overline{p}^2 \overline{q}^2))$$
$$\wedge (u^3 \neq v^3 \vee^2 \overline{x}^3 \overline{z}^3 \mid \overline{p}^3 \overline{r}^3 \vee^2 \overline{p}^3 \overline{q}^3 \overline{r}^3 \subseteq \overline{x}^1 \overline{y}^1 \overline{z}^1) \Big).$$

Proof. The equivalences (6)–(11) are straightforward and (12) is analogous to the corresponding translation in the team semantics setting (see [7]). □

The following theorem compares the expressive powers of different polyteam logics. Observe that the expressivity of the logics with two poly-dependency atoms remains the same even if either one of the atoms has the standard team semantics interpretation.

Theorem 31. *The following equivalences of logic hold:*

(1) $\mathsf{PFO}(\mathrm{pdep}) \equiv \mathsf{PFO}(\mathrm{pexc})$,
(2) $\mathsf{PFO}(\mathrm{pind}) \equiv \mathsf{PFO}(\mathrm{pexc}, \mathrm{inc}) \equiv \mathsf{PFO}(\mathrm{pinc}, \mathrm{exc}) \equiv \mathsf{PFO}(\mathrm{pdep}, \mathrm{inc})$
  $\equiv \mathsf{PFO}(\mathrm{pinc}, \mathrm{dep}) \equiv \mathsf{PFO}(\mathrm{pdep}, \mathrm{ind}) \equiv \mathsf{PFO}(\mathrm{pexc}, \mathrm{ind}) \equiv \mathsf{PFO}(\mathrm{pinc}, \mathrm{ind})$.

Proof. Item (1) follows by equivalences (7) and (10). Item (2) follows from the following list of relationships:

– $\mathsf{PFO}(\mathrm{pind}) \subseteq \mathsf{PFO}(\mathrm{pexc}, \mathrm{inc})$ by (7), (9), and (12).
– $\mathsf{PFO}(\mathrm{pexc}, \mathrm{inc}) \equiv \mathsf{PFO}(\mathrm{pinc}, \mathrm{exc})$ by (9) and (11).
– $\mathsf{PFO}(\mathrm{pexc}, \mathrm{inc}) \equiv \mathsf{PFO}(\mathrm{pdep}, \mathrm{inc})$ by (7) and (10).
– $\mathsf{PFO}(\mathrm{pinc}, \mathrm{exc}) \equiv \mathsf{PFO}(\mathrm{pinc}, \mathrm{dep})$, since exclusion (dependence, resp.) atoms can be described in $\mathsf{FO}(\mathrm{dep})$ ($\mathsf{FO}(\mathrm{exc})$, resp.) [7].
– $\mathsf{PFO}(\mathrm{pdep}, \mathrm{inc}) \subseteq \mathsf{PFO}(\mathrm{pdep}, \mathrm{ind})$, $\mathsf{PFO}(\mathrm{pexc}, \mathrm{inc}) \subseteq \mathsf{PFO}(\mathrm{pexc}, \mathrm{ind})$, and $\mathsf{PFO}(\mathrm{pinc}, \mathrm{dep}) \subseteq \mathsf{PFO}(\mathrm{pinc}, \mathrm{ind})$ since inclusion atoms can be described in $\mathsf{FO}(\mathrm{ind})$ [7] and dependence atoms by independence atoms [11].
– $\mathsf{PFO}(\mathrm{pdep}, \mathrm{ind}) \subseteq \mathsf{PFO}(\mathrm{ind})$, $\mathsf{PFO}(\mathrm{pexc}, \mathrm{ind}) \subseteq \mathsf{PFO}(\mathrm{ind})$, and $\mathsf{PFO}(\mathrm{pinc}, \mathrm{ind}) \subseteq \mathsf{PFO}(\mathrm{pind})$ by (6), (8), and (10).

$\square$

Next we show the analogue of Theorem 24 for polyteams.

Theorem 32. *Let* $\phi(R_1, \ldots, R_n)$ *be an* ESO*-sentence. There is a* $\mathsf{PFO}(\mathrm{pdep}, \mathrm{inc})$*-formula* $\phi^*(\overline{x}^1, \ldots, \overline{x}^n)$, *where* $|\overline{x}^i| = \mathrm{ar}(R_i)$, *such that for all structures* $\mathfrak{A}$ *and all polyteams* $\overline{X} = (X_1, \ldots, X_n)$ *with* $\mathsf{Dom}(X_i) = \overline{x}^i$ *and* $X_i \neq \emptyset$,

$$\mathfrak{A} \models_{\overline{X}} \phi^*(\overline{x}^1, \ldots, \overline{x}^n) \Leftrightarrow (\mathfrak{A}, R_1 := \mathrm{rel}(X_1), \ldots, R_n := \mathrm{rel}(X_n)) \models \phi(R_1, \ldots, R_n).$$

The statement holds also vice versa.

Proof. Considering first the direction from $\mathsf{PFO}(\mathrm{pdep}, \mathrm{inc})$ to ESO, let $\phi(\overline{x}^1, \ldots, \overline{x}^n)$ be a $\mathsf{PFO}(\mathrm{pdep}, \mathrm{inc})$-formula. Since poly-dependence and uni-inclusion atoms are ESO-definable, ϕ can be represented by some $\mathsf{FO}(\mathrm{ind})$-formula ϕ^* (Theorem 25), which in turn can be expressed as some ESO-formula $\psi(R)$ (Theorem 24). Let $R_1, \ldots, R_n$ be fresh relation symbols with respective arities $|\overline{x}^1|, \ldots, |\overline{x}^n|$. Let ψ' be obtained from ψ by replacing each atom $R(\overline{v}_1, \ldots, \overline{v}_n)$, where $\overline{v}_1, \ldots, \overline{v}_n$ are tuples of variables respective lengths $|\overline{x}^1|, \ldots, |\overline{x}^n|$, with the conjunction $R_1(\overline{v}_1) \wedge \ldots \wedge R_n(\overline{v}_n)$. Then we observe that for all models $\mathfrak{A}$ and polyteams $\overline{X} = (X_1, \ldots, X_n)$ represented by X,

$$\mathfrak{A} \models_{\overline{X}} \phi \iff \mathfrak{A} \models_X \phi^* \iff (\mathfrak{A}, \mathrm{rel}(X)) \models \psi \iff (\mathfrak{A}, \mathrm{rel}(X_1), \ldots, \mathrm{rel}(X_n)) \models \psi'.$$

Consider then the opposite direction. Analogously to [7], we can rewrite $\phi(R_1, \ldots, R_n)$ as

$$\exists \overline{f} \forall \overline{u} \left(\bigwedge_{i=1}^{n} (R_i(\overline{u}_i) \leftrightarrow f_{2i-1}(\overline{u}_i) = f_{2i}(\overline{u}_i)) \wedge \psi(\overline{u}, \overline{f}) \right)$$

where $\overline{f} = f_1, \ldots, f_{2n}, \ldots, f_m$ is a list of function variables, ψ is a quantifier-free formula in which no R_i appears, each $\overline{u}_i$ is a subsequence of $\overline{u}$, and each f_i occurs only as $f_i(\overline{u}_{j_i})$ for some fixed tuple $\overline{u}_{j_i}$ of variables. For instance, $j_i = i/2$ for even $i \leq 2n$.

Let $\overline{b}^i$ be sequences of variables of sort i such that $|\overline{b}^i| = |\overline{u}_i|$, and let $\overline{u}^1 \overline{y}^1$ be a sequence of variables of sort 1 such that $\overline{u}^1$ is a copy of $\overline{u}$ and $\overline{y}^1 = y_1^1, \ldots, y_m^1$. We define $\phi^*(\overline{x}^1, \ldots, \overline{x}^n)$ as the formula

$$\forall \overline{b}^1 \exists z_0^1 z_1^1 \ldots \forall \overline{b}^n \exists z_0^n z_1^n \forall \overline{u}^1 \exists \overline{y}^1 \left(\theta_0 \wedge \theta_1 \wedge \psi'(\overline{u}^1, \overline{y}^1) \right)$$

where

$$\theta_0 := \bigwedge_{i=1}^{n} = \left(\overline{b}^i, z_0^i\right) \wedge = \left(\overline{b}^i, z_1^i\right) \wedge ((\overline{b}^i \subseteq \overline{x}^i \wedge z_0^i = z_1^i) \vee^i (\overline{x}^i \mid \overline{b}^i \wedge z_0^i \neq z_1^i)),$$

$$\theta_1 := \bigwedge_{i=1}^{n} = \left(\overline{u}_i^1, y_{2i-1}^1/\overline{b}^i, z_0^i\right) \wedge = \left(\overline{u}_i^1, y_{2i}^1/\overline{b}^i, z_1^i\right) \wedge \bigwedge_{i=n+1}^{m} = \left(\overline{u}_{j_i}^1, y_i^1\right),$$

and $\psi'(\overline{u}^1, \overline{y}^1)$ is obtained from $\psi(\overline{u}, \overline{f})$ by replacing $\overline{u}$ pointwise with $\overline{u}^1$ and each $f_i(\overline{u}_{j_i})$ with y_i^1. Above, θ_0 amounts to the description of the characteristic functions f_{2i-1} and f_{2i}. We refer the reader to [7] to check that $\mathfrak{A} \models_{\overline{X}} \theta_0$ iff for all i the functions $s(\overline{b}^i) \mapsto s(z_0^i)$ and $s(\overline{b}^i) \mapsto s(z_1^i)$ determined by the assignments $s \in X_i$ agree on $s(\overline{b}^i)$ exactly when $s(\overline{b}^i) \in \mathrm{rel}(X_i)$. The poly-dependence atoms in θ_1 then transfer these functions over to the first team, and the dependence atoms in ψ_1 describe the remaining functions. As in [7], it can now be seen that ϕ^* correctly simulates ϕ. Since exclusion atoms can be expressed in dependence logic, the claim then follows. $\square$

By item (2) of Theorem 31 the result of Theorem 32 extends to a number of other logics as well. For instance, we obtain that poly-independence logic captures all ESO properties of polyteams. The proof of Theorem 32 can be now easily adapted to show that poly-exclusion and poly-dependence logic capture all downwards closed ESO properties of polyteams.

Theorem 33. *Let* $\phi(R_1, \ldots, R_n)$ *be an* ESO-*sentence that is downwards closed with respect to* R_i. *Then there is a* PFO(pdep)-*formula* $\phi^*(\overline{x}^1, \ldots, \overline{x}^n)$, *where* $|\overline{x}^i| = \mathrm{ar}(R_i)$, *such that for all structures* $\mathfrak{A}$ *and all polyteams* $\overline{X} = (X_1, \ldots, X_n)$ *with* $\mathsf{Dom}(X_i) = \overline{x}^i$ *and* $X_i \neq \emptyset$,

$$\mathfrak{A} \models_{\overline{X}} \phi^*(\overline{x}^1, \ldots, \overline{x}^n) \Leftrightarrow (\mathfrak{A}, R_1 := \mathrm{rel}(X_1), \ldots, R_n := \mathrm{rel}(X_n)) \models \phi(R_1, \ldots, R_n).$$

The statement holds also vice versa.

Proof. Consider first the direction from PFO(pdep) to ESO. By Theorem 31, PFO(pdep) is subsumed by PFO(pind), and thus the previous theorem yields a suitable ESO-sentence $\phi(R_1, \ldots, R_n)$. Since PFO(pdep) is downwards closed (Proposition 23), this sentence is also downwards closed with respect to R_i.

For the other direction, let $\phi(R_1, \ldots, R_n)$ be an ESO-sentence in which the relations R_i appear only negatively. As in the proof of Theorem 32 and by downward closure we may transform it to an equivalent form (see [22] for details)

$$\exists \overline{f} \forall \overline{u} \big(\bigwedge_{i=1}^{n} (\neg R_i(\overline{u}_i) \vee f_{2i-1}(\overline{u}_i) = f_{2i}(\overline{u}_i)) \wedge \psi(\overline{u}, \overline{f}) \big)$$

Now the translation $\phi(\overline{x}^1, \ldots, \overline{x}^n)$ is defined analogously to the proof of Theorem 32 except for θ_0 which is redefined as

$$\theta_0 := \bigwedge_{i=1}^{n} = \left(\overline{b}^i, z_0^i\right) \wedge = \left(\overline{b}^i, z_1^i\right) \wedge (\overline{x}^i \mid \overline{b}^i \vee^i z_0^i = z_1^i).$$

Finally the claim follows by eliminating the exclusion atoms from θ_0. $\square$

Next we turn to poly-inclusion logic. Over sentences, inclusion logic is known to be as expressive as positive greatest fixed point logic (PosGFP), the fragment of greatest fixed point logic in which all fixed point operators occur within the scope of an even number of negations. Moreover, all team properties definable in inclusion logic are also definable in PosGFP, but the converse does not hold due to union closure.

Theorem 34 ([9]). *Every* $\mathsf{FO}(\text{inc})$*-sentence is equivalent to some* PosGFP*-sentence, and vice versa. Moreover, for every* $\mathsf{FO}(\text{inc})$*-formula* $\phi(\overline{x})$ *there is a* PosGFP*-sentence* $\psi(R)$, *where* $|\overline{x}| = \text{ar}(R)$, *such that for all structures* $\mathfrak{A}$ *and all teams* X *with* $\mathsf{Dom}(X) = \overline{x}$,

$$\mathfrak{A} \models_X \phi(\overline{x}) \iff (\mathfrak{A}, R := \text{rel}(X)) \models \psi(R).$$

To generalise these results to polyteam semantics, we use the fact that all team connectives and quantifiers distribute over classical disjunctions.

Proposition 35 ([8]). *Let* ϕ *be an* $\mathsf{FO}(\mathcal{C}, \varovee)$*-formula, where* $\mathcal{C}$ *is any set of atoms. Then* ϕ *is equivalent to some formula of the form* $\psi_1 \varovee \ldots \varovee \psi_n$ *where* $\psi_1, \ldots, \psi_n$ *are* $\mathsf{FO}(\mathcal{C})$*-formulae.*

Theorem 36. *Every* $\mathsf{PFO}(\text{pinc})$*-sentence is equivalent to some* PosGFP*-sentence, and vice versa. Moreover, for every* $\mathsf{PFO}(\text{pinc})$*-formula* $\phi(\overline{x}^1, \ldots, \overline{x}^n)$ *there is a* PosGFP*-sentence* $\psi(R_1, \ldots, R_n)$, *where* $|\overline{x}^i| = \text{ar}(R_i)$, *such that for all structures* $\mathfrak{A}$ *and all polyteams* $\overline{X} = (X_1, \ldots, X_n)$ *with* $\mathsf{Dom}(X_i) = \overline{x}^i$,

$$\mathfrak{A} \models_{\overline{X}} \phi(\overline{x}^1, \ldots, \overline{x}^n) \iff (\mathfrak{A}, R_1 := \text{rel}(X_1), \ldots, R_n := \text{rel}(X_n)) \models \psi(R_1, \ldots, R_n).$$

Proof. Let $\phi(\overline{x}^1, \ldots, \overline{x}^n) \in \mathsf{PFO}(\text{pinc})$ be a formula, and let $\phi^*(\overline{x}^1, \ldots, \overline{x}^n) \in \mathsf{FO}(\text{inc}, \varovee)$ be its team representation, obtained by Corollary 22, in which two additional constants 0 and 1 occur. Without loss of generality, we may restrict our attention to structures with at least two elements. By Proposition 35 ϕ^* is equivalent to a disjunction $\psi_1 \varovee \ldots \varovee \psi_n$, where ψ_i are $\mathsf{FO}(\text{inc})$-formulae. By Theorem 34 each ψ_i is equivalent to some PosGFP-sentence $\Phi_i(R)$. Define

$$\Phi(R) := \exists yz \Big(y \neq z \wedge \big(\Phi_1(y/0, z/1) \vee \ldots \vee \Phi_n(y/0, z/1) \big) \Big),$$

where $\Phi_i(y/0, z/1)$ are obtained from Φ_i by substituting y and z respectively for 0 and 1. Let $R_1, \ldots, R_n$ be fresh relation symbols with respective arities $|\overline{x}^1|, \ldots, |\overline{x}^n|$. Let Φ' be the formula obtained from Φ by replacing each atom $R(\overline{y}_1, \ldots, \overline{y}_n)$, where $\overline{y}_1, \ldots, \overline{y}_n$ are tuples of variables with respective lengths $|\overline{x}^1|, \ldots, |\overline{x}^n|$, with the conjunction $R_1(\overline{y}_1) \wedge \ldots \wedge R_n(\overline{y}_n)$. The following equivalence holds for all structures $\mathfrak{A}$, with at least two elements, and strictly non-empty polyteams $\overline{X} = (X_1, \ldots, X_n)$. Let X denote the team representation of $\overline{X}$ obtained by taking the Cartesian product of the teams X_i, $1 \leq i \leq n$.

$$\mathfrak{A} \models_{\overline{X}} \phi \Leftrightarrow \mathfrak{A}^* \models_X \psi_1 \varovee \ldots \varovee \psi_n$$

$$\Leftrightarrow (\mathfrak{A}, R := \text{rel}(X)) \models \Phi$$

$$\Leftrightarrow \big(\mathfrak{A}, R_1 := \text{rel}(X_1), \ldots, R_n := \text{rel}(X_n) \big) \models \Phi',$$

where $\mathfrak{A}^*$ is an expansion of $\mathfrak{A}$ with two distinct constants 0 and 1. The converse direction for PosGFP-sentences follows by Theorem 34 and since $\mathsf{FO}(\text{inc})$ is a fragment of $\mathsf{PFO}(\text{pinc})$. $\square$

6 Conclusion

In this article we have laid the foundations of polyteam semantics in order to facilitate the fruitful exchange of ideas and results between team semantics and database theory. Our results show that many of the familiar properties and results from team semantics carry over to the polyteam setting. In particular, we identified a natural polyteam analogue of dependence atoms and gave a complete axiomatisation for the associated implication problem. We also showed that polyteam semantics can sometimes be reduced to team semantics,

although it can be questioned whether such an interpretation is reasonable in the first place. The examples of this paper demonstrate that polyteam semantics is a conceptually more natural framework for capturing properties of sets of relations. Specifications for multiple relations are easier to parse if different relations are explicitly distinguished in formulae. Also, if polyteam logics are interpreted as data constraint languages, as in Example 10, then the reduction from polyteam semantics to team semantics incurs an unnecessary computational overhead. First, a single team to represent the polyteam has to be constructed, e.g., by taking a Cartesian product of all coordinate teams; and second, this team has to be validated against a team logic formula which is much larger in size than the initial polyteam formula.

Our results also open up interesting avenues for further research. One question is to determine whether poly-dependence logic reduces to dependence logic. Our methods only work for poly-independence and poly-inclusion logic, and the proviso in the latter case was to include classical disjunction. As inclusion logic with classical disjunction is not union closed, it would also be interesting to study the team properties definable in this logic. Apart from poly-dependence atoms, we did not consider axioms for any other poly-atoms. Since the axioms of poly-inclusion atoms are already known from database theory, a natural next step would be to axiomatise marginal poly-independence atoms. It would also be interesting to develop axiomatic methods for more expressive fragments of polyteam logics (cf. [12,23,28]).

References

1. William W. Armstrong. Dependency Structures of Data Base Relationships. In *Proc. of IFIP World Computer Congress*, pages 580–583, 1974.
2. Marco A. Casanova, Ronald Fagin, and Christos H. Papadimitriou. Inclusion dependencies and their interaction with functional dependencies. *J. Comput. Syst. Sci.*, 28(1):29–59, 1984.
3. Arnaud Durand, Miika Hannula, Juha Kontinen, Arne Meier, and Jonni Virtema. Approximation and dependence via multiteam semantics. *Ann. Math. Artif. Intell.*, 83(3-4):297–320, 2018. URL: `https://doi.org/10.1007/s10472-017-9568-4`, `doi:10.1007/s10472-017-9568-4`.
4. Arnaud Durand, Miika Hannula, Juha Kontinen, Arne Meier, and Jonni Virtema. Probabilistic team semantics. In *Foundations of Information and Knowledge Systems - 10th International Symposium, FoIKS 2018, Budapest, Hungary, May 14-18, 2018, Proceedings*, pages 186–206, 2018. URL: `https://doi.org/10.1007/978-3-319-90050-6_11`, `doi:10.1007/978-3-319-90050-6\_11`.
5. 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: Theory and Applications*, pages 5–32. Springer International Publishing, Cham, 2016.
6. Ronald Fagin, Phokion G. Kolaitis, Renée J. Miller, and Lucian Popa. Data exchange: semantics and query answering. *Theoretical Computer Science*, 336(1):89 – 124, 2005.
7. Pietro Galliani. Inclusion and exclusion dependencies in team semantics: On some logics of imperfect information. *Annals of Pure and Applied Logic*, 163(1):68 – 84, 2012.
8. Pietro Galliani. On strongly first-order dependencies. In *Dependence Logic, Theory and Applications*, pages 53–71. 2016.
9. Pietro Galliani and Lauri Hella. Inclusion Logic and Fixed Point Logic. In Simona Ronchi Della Rocca, editor, *Computer Science Logic 2013 (CSL 2013)*, volume 23 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 281–295, Dagstuhl, Germany, 2013. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.
10. Dan Geiger, Azaria Paz, and Judea Pearl. Axioms and algorithms for inferences involving probabilistic independence. *Information and Computation*, 91(1):128–141, 1991.
11. Erich Grädel and Jouko A. Väänänen. Dependence and independence. *Studia Logica*, 101(2):399–410, 2013.
12. Miika Hannula. Axiomatizing first-order consequences in independence logic. *Ann. Pure Appl. Logic*, 166(1):61–91, 2015.
13. Miika Hannula. Reasoning about embedded dependencies using inclusion dependencies. In *Logic for Programming, Artificial Intelligence, and Reasoning - 20th International Conference, LPAR-20 2015, Suva, Fiji, November 24-28, 2015, Proceedings*, pages 16–30, 2015. URL: `https://doi.org/10.1007/978-3-662-48899-7_2`, `doi:10.1007/978-3-662-48899-7\_2`.
14. Miika Hannula, Åsa Hirvonen, Juha Kontinen, Vadim Kulikov, and Jonni Virtema. Facets of distribution identities in probabilistic team semantics. In *JELIA*, volume 11468 of *Lecture Notes in Computer Science*, pages 304–320. Springer, 2019.
15. Miika Hannula and Juha Kontinen. A finite axiomatization of conditional independence and inclusion dependencies. *Inf. Comput.*, 249:121–137, 2016.

16. Miika Hannula, Juha Kontinen, and Sebastian Link. On the finite and general implication problems of independence atoms and keys. *J. Comput. Syst. Sci.*, 82(5):856–877, 2016.
17. Christian Herrmann. On the undecidability of implications between embedded multivalued database dependencies. *Information and Computation*, 122(2):221 – 235, 1995.
18. Wilfrid Hodges. Compositional Semantics for a Language of Imperfect Information. *Journal of the Interest Group in Pure and Applied Logics*, 5 (4):539–563, 1997.
19. Paris C. Kanellakis. Elements of relational database theory. In *Handbook of Theoretical Computer Science, Volume B: Formal Models and Sematics (B)*, pages 1073–1156. 1990.
20. Juha Kontinen, Antti Kuusisto, and Jonni Virtema. Decidability of Predicate Logics with Team Semantics. In *Proceedings of MFCS 2016*, pages 60:1–60:14, 2016.
21. Juha Kontinen, Sebastian Link, and Jouko A. Väänänen. Independence in database relations. In *Proc. 20th WoLLIC*, volume 8071 of *LNCS*, pages 179–193. Springer, 2013.
22. Juha Kontinen and Jouko Väänänen. On definability in dependence logic. *Journal of Logic, Language and Information*, 3(18):317–332, 2009.
23. Juha Kontinen and Jouko A. Väänänen. Axiomatizing first-order consequences in dependence logic. *Ann. Pure Appl. Logic*, 164(11):1101–1117, 2013.
24. Antti Kuusisto. A double team semantics for generalized quantifiers. *Journal of Logic, Language and Information*, 24(2):149–191, 2015.
25. Yehoshua Sagiv and Scott F. Walecka. Subset dependencies and a completeness result for a subclass of embedded multivalued dependencies. *J. ACM*, 29(1):103–117, 1982.
26. Douglas Stott Parker Jr. and Kamran Parsaye-Ghomi. Inferences involving embedded multivalued dependencies and transitive dependencies. In *Proceedings of the 1980 ACM SIGMOD International Conference on Management of Data*, pages 52–57, 1980.
27. Jouko Väänänen. *Dependence Logic*. Cambridge University Press, 2007.
28. Fan Yang. Negation and partial axiomatizations of dependence and independence logic revisited. *Ann. Pure Appl. Logic*, 170(9):1128–1149, 2019. URL: `https://doi.org/10.1016/j.apal.2019.04.010`, `doi:10.1016/j.apal.2019.04.010`.