Ann Math Artif Intell (2018) 83:297–320
https://doi.org/10.1007/s10472-017-9568-4

Approximation and dependence via multiteam semantics

Arnaud Durand[1] · Miika Hannula[2] · Juha Kontinen[2] · Arne Meier[3] · Jonni Virtema[3]

Published online: 18 January 2018

Abstract We define a variant of team semantics called *multiteam semantics* based on multisets and study the properties of various logics in this framework. In particular, we define natural probabilistic versions of inclusion and independence atoms and certain approximation operators motivated by approximate dependence atoms of Väänänen.

Keywords Dependence logic · Team semantics · Computational complexity

Mathematics Subject Classification (2010) 03B70 · 68Q17

✉ Jonni Virtema
jonni.virtema@uhasselt.be

Arnaud Durand
durand@math.univ-paris-diderot.fr

Miika Hannula
m.hannula@auckland.ac.nz

Juha Kontinen
juha.kontinen@helsinki.fi

Arne Meier
meier@thi.uni-hannover.de

[1] Institut de Mathématiques de Jussieu - Paris Rive Gauche, CNRS UMR 7586 - Université Paris Diderot, Paris, France

[2] Department of Mathematics and Statistics, University of Helsinki, Helsinki, Finland

[3] Institut für Theoretische Informatik, Leibniz Universität Hannover, Hanover, Germany

1 Introduction

Dependence logic was introduced by Väänänen in 2007 [34]. It extends first-order logic with dependence atomic formulas (dependence atoms) $=(\vec{x}, y)$ with the meaning that the value of the variable y is functionally determined by the values of the variables $\vec{x}$. The notion of dependence has real meaning only in plurals. Thus, in contrast to the usual Tarskian semantics, in dependence logic, the satisfaction of formulas is defined not via single assignments but via sets of assignments. Such sets are called *teams* and the semantics is called *team semantics*. In this article, we take a further step of replacing structures and teams by their multiset analogues. Multiteams have been considered in some earlier works [21, 22, 36] but so far no systematic study of the subject in the team semantics context has appeared. In the temporal logic setting (in the context of computation tree logic) multiteam semantics has been introduced and studied recently [30]. In this article we define the so-called *lax* and *strict* multiteam semantics and study properties of various logics under these semantics. Moreover, we show how the shift from sets to multisets naturally gives rise to probabilistic and approximate versions of dependence logic.

The idea of team semantics goes back to Hodges [20] whose aim was to define compositional semantics for *independence-friendly logic* [19]. The introduction of dependence logic and its many variants has manifested that team semantics is a very interesting and versatile semantical framework. In fact, team semantics has natural propositional, modal, and temporal variants. The study of *modal dependence logic* was initiated by Väänänen [35] in 2008. Shortly after, *extended modal dependence logic* was introduced by Ebbing et al. [5] and *modal independence logic* by Kontinen et al. [29]. In purely propositional context the study was initiated by Yang and Väänänen [40] and further studied, e.g., by Hannula et al. [17]. One of the most important developments in the area of team semantics was the introduction of *independence logic* by Grädel and Väänänen [12] in which dependence atoms of dependence logic are replaced by *independence atoms* $\vec{y} \perp_{\vec{x}} \vec{z}$. The natural meaning of the independence atom $\vec{y} \perp_{\vec{x}} \vec{z}$ is that, when the value of $\vec{x}$ is fixed, knowing the value of $\vec{z}$ does not tell us anything new about the value of $\vec{y}$. Soon after the introduction of independence logic Galliani [7] showed that independence atoms can be further analysed, and alternatively expressed, in terms of inclusion and exclusion atoms. The inclusion atom $\vec{x} \subseteq \vec{y}$ expresses that each value taken by $\vec{x}$ in a team X appears also as a value of $\vec{y}$ in X. The meaning of the exclusion atom $\vec{x}|\vec{y}$ is that $\vec{x}$ and $\vec{y}$ have no common values in X.

Independence, inclusion, and exclusion atoms have very interesting computational properties in the team semantics setting. For example, in lax semantics inclusion atoms give rise to a variant of dependence logic that corresponds to the complexity class P over finite ordered structures [9]. On the other hand, under strict team semantics inclusion logic captures the complexity class NP [8, 14]. In the context of two-variable logics, the complexity of the satisfiability and validity problems of several team based logics have been studied in [26, 27]. The complexity theoretic aspects of propositional, modal, and first-order logics with team semantics have been studied extensively during the past few years; see the survey of Durand et al. [4] and the references therein.

A team X over variables $x_1, \ldots, x_n$ can be viewed as a database table with $x_1, \ldots, x_n$ as its attributes. Under this interpretation, dependence, inclusion, exclusion, and independence atoms correspond exactly to functional, inclusion, exclusion, and embedded multivalued dependencies, respectively. These dependencies have been studied extensively in database theory. The close connection between team semantics and database theory has already led to fruitful interactions between these areas [15, 16, 28]. It is worth noting that multiset semantics (also known as bag semantics) is widely used in databases [1, 24, 31]. On the other hand,

independence atoms, embedded multivalued dependencies, and the notion of conditional independence $\vec{Y} \perp \vec{Z}|\vec{X}$ in statistics have very interesting connections, see, e.g., [13, 38]. In this article we establish that, in the multiteam semantics setting, independence atoms can be naturally interpreted exactly as conditional statistical independence. Probabilistic versions of dependence logic have been previously studied by Galliani and Mann [6, 10].

In practise dependencies such as functional dependence do not hold absolutely but with a small margin of error. In order to logically model such scenarios, Väänänen introduced approximate dependence atoms [36]. The corresponding approximate functional dependencies have been studied in the context of data mining [23]. In this article we define a general approximation operator which, in particular, can be used to express approximate dependence atoms. In the last section of the article, we study the computational aspects of logics extended by the approximation operator.

Previous work on multisets in team semantics The idea of generalising team semantics by the use of multisets has been discussed in several articles. Hyttinen et al. [22] study multiteams and their generalisations called *quantum teams*. Quantum teams are used to give semantics to a propositional logic called *quantum team logic*, that can be used for the logical analysis of phenomena in quantum physics. Moreover Hyttinen et al. [21] define a notion of a *measure team* and *measure team logic*. The latter is a logic for making inferences about probabilities of first-order formulas in measure teams. Furthermore Krebs et al. introduced team semantics with multisets for the temporal logic CTL [30]. Finally the fact that under multiteam semantics approximate dependence atoms have the locality property (compare to Proposition 37) is discussed by Väänänen [36].

Organisation This article is organised as follows. Section 2 briefly discusses the basic concepts and definitions. The generalisation of team semantics to multisets is presented in Section 3. Section 4 defines the approximation operators, and in Section 5 the complexity-theoretic aspects of logics with the approximation operators are studied.

2 Preliminaries

We assume familiarity with standard notions in computational complexity theory and logic. We will make use of the complexity classes NP and P. For an introduction to this topic, we refer to the excellent textbook of Papadimitriou [33].

2.1 Team Semantics

Vocabularies τ are finite 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 $ar(R_i)$-ary relation on A (i.e., $R_i^{\mathfrak{A}} \subseteq A^{ar(R_i)}$). We use $\mathfrak{A}$, $\mathfrak{B}$, etc. to denote τ-structures and A, B, etc. to denote the corresponding domains. In this article we restrict attention to finite structures.

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*. The set D is the *domain* of s, and the set A the *codomain* of s. 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* is a finite set of assignments with a common domain and codomain. Let X be a team, A a finite 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 $\{s(a/x) \mid s \in X, a \in F(s)\}$. Let $\mathfrak{A}$ be a τ-structure and X a team with codomain A, then we say that X is a team of $\mathfrak{A}$.

Let τ be a set of relation symbols. The syntax of first-order logic $\mathsf{FO}(\tau)$ is given by the following grammar, where $R \in \tau$, $\vec{x}$ is a tuple of variables, and x and y are variables. Note that in the definition the scope of negation is restricted to atomic formulas.

$$\varphi \quad ::= \quad x = y \mid x \neq y \mid R(\vec{x}) \mid \neg R(\vec{x}) \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid \exists x\varphi \mid \forall x\varphi.$$

Let $\vec{x}$, $\vec{y}$ be tuples of variables and φ a formula. We write $\mathrm{Var}(\varphi)$ for the set of variables that occur in φ, and $\mathrm{Var}(\vec{x})$ for the set of variables listed in $\vec{x}$. We also write $\vec{x}\vec{y}$ for the concatenation of $\vec{x}$ and $\vec{y}$, $\vec{x} \cap \vec{y}$ for any tuple listing the variables in $\mathrm{Var}(\vec{x}) \cap \mathrm{Var}(\vec{y})$, and $\vec{x} \setminus \vec{y}$ for any tuple listing the variables in $\mathrm{Var}(\vec{x}) \setminus \mathrm{Var}(\vec{y})$. For an assignment s, we write $s(\vec{x})$ to denote the sequence $\big(s(x_1), \ldots, s(x_n)\big)$.

Next, we define the lax and strict team semantics of first-order logic. It is worth noting that the disjunction has a non-classical interpretation. The classical disjunction of team semantics $\mathfrak{A} \models_X \varphi \mathbin{⩒} \psi \Leftrightarrow \mathfrak{A} \models_X \varphi$ or $\mathfrak{A} \models_X \psi$ does not correspond to the classical disjunction of first-order logic. E.g., $⩒$ does not satisfy the law of the excluded middle and consequently is sometimes referred to as the intuitionistic disjunction. In turn, the non-classical disjunction of team semantics does correspond to the classical disjunction of ordinary first-order logic; extensions of first-order logic in team semantics are conservative extensions of ordinary first-order logic.

Definition 1 (Lax team semantics) Let $\mathfrak{A}$ be a τ-structure and X a team of $\mathfrak{A}$. The satisfaction relation $\models_X$ for first-order logic is defined as follows:

$$\begin{aligned}
\mathfrak{A} \models_X x = y \quad &\Leftrightarrow \forall s \in X : s(x) = s(y)\\
\mathfrak{A} \models_X x \neq y \quad &\Leftrightarrow \forall s \in X : s(x) \neq s(y)\\
\mathfrak{A} \models_X R(\vec{x}) \quad &\Leftrightarrow \forall s \in X : s(\vec{x}) \in R^{\mathfrak{A}}\\
\mathfrak{A} \models_X \neg R(\vec{x}) \quad &\Leftrightarrow \forall s \in X : s(\vec{x}) \notin R^{\mathfrak{A}}\\
\mathfrak{A} \models_X (\psi \wedge \theta) \quad &\Leftrightarrow \mathfrak{A} \models_X \psi \text{ and } \mathfrak{A} \models_X \theta\\
\mathfrak{A} \models_X (\psi \vee \theta) \quad &\Leftrightarrow \mathfrak{A} \models_Y \psi \text{ and } \mathfrak{A} \models_Z \theta \text{ for some } Y, Z \subseteq X s.t. Y \cup Z = X\\
\mathfrak{A} \models_X \forall x\psi \quad &\Leftrightarrow \mathfrak{A} \models_{X[A/x]} \psi\\
\mathfrak{A} \models_X \exists x\psi \quad &\Leftrightarrow \mathfrak{A} \models_{X[F/x]} \psi \text{ holds for some } F\colon X \to \mathcal{P}(A) \setminus \{\emptyset\}.
\end{aligned}$$

The so-called *strict* team semantics is obtained from the previous definition by adding the following two requirements.

(i) Disjunction: $Y \cap Z = \emptyset$.
(ii) Existential quantification: $F(s)$ is singleton for all $s \in X$.

For a model $\mathfrak{A}$ and a *sentence* φ (i.e., a formula with no free variables), the satisfaction relation $\models$ is defined as:

$$\mathfrak{A} \models \varphi \text{ if } \mathfrak{A} \models_{\{\emptyset\}} \varphi,$$

where $\{\emptyset\}$ denotes the singleton team of empty assignment.

Team semantics enables extending first-order logic with various dependency notions. The following dependency atoms were introduced in [7, 12, 34].

Definition 2 (Dependency atoms) Let $\mathfrak{A}$ be a model and X a team of $\mathfrak{A}$. If $\vec{x}$, $\vec{y}$ are variable sequences, then $=(\vec{x}, \vec{y})$ is a *dependence atom* with the satisfaction relation:

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

If $\vec{x}$, $\vec{y}$ are variable sequences of the same length, then $\vec{x} \subseteq \vec{y}$ is an *inclusion atom* with the satisfaction relation:

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

If $\vec{x}$, $\vec{y}$, $\vec{z}$ are variable sequences, then $\vec{y} \perp_{\vec{x}} \vec{z}$ is a *conditional independence atom* with the satisfaction relation:

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

Note that in the previous definition it is allowed that some or all of the vectors of variables have length 0. For example, $\mathfrak{A} \models_X =(\emptyset, \vec{x})$ (written often as $\mathfrak{A} \models_X =(\vec{x})$ in the literature) holds if and only if $\forall s \in X : s(\vec{x}) = \vec{c}$ holds for some fixed tuple $\vec{c}$, and $\mathfrak{A} \models_X \vec{y} \perp_{\vec{x}} \vec{z}$ holds always if either of the vectors $\vec{y}$ or $\vec{z}$ is of length 0.

We write FO for first-order logic, and given a set of atoms $\mathcal{C}$, we write $\mathsf{FO}(\mathcal{C})$ (omitting the set parentheses of $\mathcal{C}$) for the logic obtained by adding the atoms of $\mathcal{C}$ to FO. Here we denote dependence atoms by $=(\cdot)$, inclusion atoms by $\subseteq$, and conditional independence atoms by $\perp_c$ (the atoms introduced later will be denoted similarly). For instance, $\mathsf{FO}(=(\cdot))$ denotes dependence logic.

Often in literature dependence atoms are defined such that $\vec{y}$ is a single variable, i.e., the widely used form is $=(\vec{x}, y)$. The definition above yields the strongest form of *functional dependence*. Moreover the atom $=(\vec{x}, \vec{y})$ can be equivalently rewritten as a conjunction of dependence atoms of type $=(\vec{x}, y)$.

3 Multiteam semantics

In this section we generalise team semantics with the concept of multisets. Multisets and multiteam semantics can be used, e.g., in applications to database theory to model reasoning with databases with duplicates. In practise, for a multitude of reasons, the existence of duplicates in databases is very common. Again as previously noted, we restrict attention to finite sets and finite multisets.

In Section 3.1, the basics of multiteam semantics are given. In Section 3.2, we introduce probabilistic versions of inclusion and independence atoms, and in Section 3.3 the fundamental properties locality, flatness, and union closure are discussed in the multiteam setting. In Sections 3.4 and 3.5, nonprobabilistic dependency notions in multiteam setting and probabilistic dependency notions in team semantics setting are studied, respectively.

3.1 Foundations

In the following definition, occurrences of "zero multiplicities" are allowed for notational convenience.

Definition 3 (Multiset) A *multiset* is a pair (A, m) where A is a set and $m\colon A \to \mathbb{N}$ is a (multiplicity) function. The function m determines the *multiplicities* of the elements in the

multiset (A, m). A multiset (X, m) is a *multiteam* if the underlying set X is a team. The domain (or the codomain) of the multiteam (X, m) is the domain (codomain) of the team X.

For each multiset (A, m), we define the *canonical set representative* $[(A, m)]_{\mathrm{cset}}$ of (A, m) as follows:

$$[(A, m)]_{\mathrm{cset}} := \{\, (a, i) \mid a \in A,\ 0 < i \leq m(a) \,\}.$$

We say that (A, m) is finite whenever $[(A, m)]_{\mathrm{cset}}$ is finite. We say that a multiset (A, m) is a submultiset of a multiset (B, n), and write $(A, m) \subseteq (B, n)$, if and only if $[(A, m)]_{\mathrm{cset}} \subseteq [(B, n)]_{\mathrm{cset}}$. Furthermore, we define that $(A, m) = (B, n)$ if and only if both $(A, m) \subseteq (B, n)$ and $(B, n) \subseteq (A, m)$ hold.

The *disjoint union* $(A, m) \uplus (B, n)$ of (A, m) and (B, n) is the multiset (C, k), where $C := A \cup B$ and $k : C \to \mathbb{N}$ is the function defined as follows:

$$k(s) := \begin{cases} m(s) + n(s) & \text{if } s \in A \text{ and } s \in B, \\ m(s) & \text{if } s \in A \text{ and } s \notin B, \\ n(s) & \text{if } s \notin A \text{ and } s \in B. \end{cases}$$

We write $|(A, m)|$ to denote the size of the multiset (A, m), i.e., $|(A, m)| := \sum_{a \in A} m(a)$. The set of non-empty submultisets of a multiset (A, m) is the set

$$\mathcal{P}^+\big((A, m)\big) := \{(C, l) \mid (C, l) \subseteq (A, m) \text{ s.t.} l(c) \geq 1 \text{ for each} c \in C\} \setminus \{(\emptyset, \emptyset)\}.$$

Let (X, m) be a multiteam, (A, n) be a finite multiset, and $F\colon [(X, m)]_{\mathrm{cset}} \to \mathcal{P}^+\big((A, n)\big)$ be a function. We denote by $(X, m)[(A, n)/x]$ the modified multiteam defined as

$$\biguplus_{s \in X} \biguplus_{a \in A} \Big\{ \big(s(a/x), m(s) \cdot n(a)\big) \Big\}.$$

By $(X, m)[F/x]$ we denote the multiteam defined as

$$\biguplus_{s \in X} \biguplus_{1 \leq i \leq m(s)} \Big\{ \big(s(b/x), l(b)\big) \mid (B, l) = F\big((s, i)\big), b \in B \Big\}.$$

Example 4 Figure 1 depicts a canonical set of a team (X, m) with domain $\{x, y\}$ and codomain $A = \{\alpha, \beta\}$. Observe that (X, m) is a multiteam where $X = \{s, t, u\}$ and $m\colon X \to \mathbb{N}$ is a multiplicity function such that $m(s) = 2$, $m(t) = 1$, and $m(u) = 1$. Note that s and t agree on y but disagree on x. Figure 1 also depicts the canonical set of $(X, m)[F/x]$ where F is defined as follows:

$$F\big((s, 1)\big) = (A, n),\ F\big((s, 2)\big) = (A, n),$$
$$F\big((t, 1)\big) = (A, n),\ F\big((u, 1)\big) = (\{\beta\}, n),$$

where n is a multiplicity function such that $n(\alpha) = 2$ and $n(\beta) = 1$. The assignments s_1 and s_2 arise from $(s, 1)$, $(s, 2)$, and $(t, 1)$; the assignment u_1 arises from $(u, 1)$. Furthermore, note that $[(X, m)[F/x]]_{\mathrm{cset}} \setminus \{(u_1, 1)\}$ is the canonical set of $(X \setminus \{u\}, m)[(A, n)/x]$.

A τ-multistructure is a tuple $\mathfrak{A} = \big((A, m), (R_i^{\mathfrak{A}})_{R_i \in \tau}\big)$ where (A, m) is a non-empty multiset and, for each $R_i \in \tau$, $R_i^{\mathfrak{A}}$ is an $\mathrm{ar}(R_i)$-ary relation over the set $\{a \in A \mid m(a) \geq 1\}$. A multiteam (X, m) over $\mathfrak{A}$ is a multiteam with codomain A.

Next we define lax multiteam semantics for first-order logic.

Fig. 1 A set $[(X,m)]_{\mathrm{cset}}$ and its associated $[(X,m)[F/x]]_{\mathrm{cset}}$ for function F of Example 4

$[(X,m)]_{\mathrm{cset}}$	y	x
$(s,1)$	α	α
$(s,2)$	α	α
$(t,1)$	α	β
$(u,1)$	β	α

$[(X,m)[F/x]]_{\mathrm{cset}}$	y	x
$(s_1,1)$	α	α
$(s_1,2)$	α	α
$(s_1,3)$	α	α
$(s_1,4)$	α	α
$(s_1,5)$	α	α
$(s_1,6)$	α	α
$(s_2,1)$	α	β
$(s_2,2)$	α	β
$(s_2,3)$	α	β
$(u_1,1)$	β	β

Definition 5 (Lax multiteam semantics) Let $\mathfrak{A}$ be a τ-multistructure, (A,n) the domain of $\mathfrak{A}$, and (X,m) a multiteam over $\mathfrak{A}$. The satisfaction relation $\models_{(X,m)}$ is defined as follows:

$$
\begin{aligned}
&\mathfrak{A} \models_{(X,m)} x = y && \Leftrightarrow \forall s \in X: \text{ if } m(s) \geq 1 \text{ then } s(x) = s(y)\\
&\mathfrak{A} \models_{(X,m)} x \neq y && \Leftrightarrow \forall s \in X: \text{ if } m(s) \geq 1 \text{ then } s(x) \neq s(y)\\
&\mathfrak{A} \models_{(X,m)} R(\vec{x}) && \Leftrightarrow \forall s \in X: \text{ if } m(s) \geq 1 \text{ then } s(\vec{x}) \in R^{\mathfrak{A}}\\
&\mathfrak{A} \models_{(X,m)} \neg R(\vec{x}) && \Leftrightarrow \forall s \in X: \text{ if } m(s) \geq 1 \text{ then } s(\vec{x}) \notin R^{\mathfrak{A}}\\
&\mathfrak{A} \models_{(X,m)} (\psi \wedge \theta) && \Leftrightarrow \mathfrak{A} \models_{(X,m)} \psi \text{ and } \mathfrak{A} \models_{(X,m)} \theta\\
&\mathfrak{A} \models_{(X,m)} (\psi \vee \theta) && \Leftrightarrow \mathfrak{A} \models_{(Y,k)} \psi \text{ and } \mathfrak{A} \models_{(Z,\ell)} \theta \text{ for some multisets}\\
&&& \quad (Y,k), (Z,\ell) \subseteq (X,m) s.t. (X,m) \subseteq (Y,k) \uplus (Z,\ell).\\
&\mathfrak{A} \models_{(X,m)} \forall x \psi && \Leftrightarrow \mathfrak{A} \models_{(X,m)[(A,n)/x]} \psi\\
&\mathfrak{A} \models_{(X,m)} \exists x \psi && \Leftrightarrow \mathfrak{A} \models_{(X,m)[F/x]} \psi \text{ holds for some function}\\
&&& \quad F: [(X,m)]_{\mathrm{cset}} \to \mathcal{P}^{+}\big((A,n)\big).
\end{aligned}
$$

The so-called *strict multiteam semantics* is obtained from the previous definition by adding the following two requirements.

(i) Disjunction: $(Y,k) \uplus (Z,l) = (X,m)$.
(ii) Existential quantification: for all $s \in X$ and $0 < i \leq m(s)$, $F\big((s,i)\big) = (B,n)$ for some singleton $B = \{b\}$ and $n(b) = 1$.

In most parts of this paper, the choice of semantics (strict or lax) for existential quantifiers and disjunctions does not have any effect. Thus, if not explicitly mentioned, all the results hereafter work for both strict and lax semantics. Note also that the multiset nature of domains of multistructures manifest itself only in the truth conditions of the quantifiers.

As demonstrated by the following proposition, the multiteam semantics and team semantics for first-order logic coincide when the multisets in multistructures and multiteams are essentially sets. Furthermore the semantical clauses of lax (strict, resp.) multiteam semantics collapse to the semantical clauses of lax (strict, resp.) team semantics. The proof of the proposition is self-evident.

Proposition 6 *Let $\mathfrak{A}$ be a multistructure with domain $(A,1)$, and $(X,1)$ a multiteam over $\mathfrak{A}$, where 1 denotes the constant 1 function, i.e., $1(a) = 1(s) = 1$ for all $a \in A$ and $s \in X$.*

Define $\mathfrak{B} := (A, (R^{\mathfrak{A}})_{R\in\tau})$. *Then for every* $\varphi \in \mathsf{FO}$ *it holds that*

$$\mathfrak{A} \models_{(X,1)} \varphi \text{ if and only if } \mathfrak{B} \models_X \varphi.$$

We point out that multiteam semantics is reminiscent of the *bag semantics* in relational databases. For instance, the disjoint union of multiteams and the restriction of a multiteam (introduced in Section 3.3) align with the union and projection operators of the *bag relational algebra* [32]. Similarly, the *query containment problem* in bag semantics is formulated using a notion of *bag containment* that is analogous to the containment relation of multiteams.

It is well known that bag semantics, although being relevant for database practise where duplicates are tolerated for computational reasons, renders the query containment problem difficult. Already for conjunctive queries this problem becomes, from being NP-complete in *set semantics*, Π_2^{P}-hard with decidability remaining an open problem when considering bag semantics instead [2].

Similar increase in complexity should be anticipated for multiteam semantics, although the vantage point here is somewhat different from that of bag semantics as the focus is not on queries over but on properties of multisets of relations.

3.2 Probabilistic dependency notions

Next we generalise inclusion and conditional independence atoms to multiteams by introducing their probabilistic versions. For a multiteam (X, m) of codomain A, a tuple of variables $\vec{x}$ from $\mathrm{Dom}(X)$, and $\vec{a} \in A^{|\vec{x}|}$, we denote by $(X, m)_{\vec{x}=\vec{a}}$ the multiteam (X, n) where n agrees with m on all assignments $s \in X$ with $s(\vec{x}) = \vec{a}$, and otherwise n maps s to 0.

Definition 7 Let $\mathfrak{A}$ be a multistructure with domain (A, n), and (X, m) a multiteam over $\mathfrak{A}$. If $\vec{x}, \vec{y}$ are variable sequences of the same length, then $\vec{x} \preceq \vec{y}$ is a *probabilistic inclusion atom* with the following semantics:

$$\mathfrak{A} \models_{(X,m)} \vec{x} \preceq \vec{y} \text{ if} |(X, m)_{\vec{x}=s(\vec{x})}| \leq |(X, m)_{\vec{y}=s(\vec{x})}| \text{ for all} s\colon \mathrm{Var}(\vec{x}) \to A.$$

It is worth noting that for finite multiteams $\mathfrak{A} \models_{(X,m)} \vec{x} \preceq \vec{y}$ implies that

$$|(X, m)_{\vec{x}=s(\vec{x})}| = |(X, m)_{\vec{y}=s(\vec{x})}| \text{ for all} s\colon \mathrm{Var}(\vec{x}) \to A.$$

Otherwise, the inequality below would be rendered strict:

$$|(X, m)| = \Sigma_{s\colon \mathrm{Var}(\vec{x})\to A} |(X, m)_{\vec{x}=s(\vec{x})}| \leq \Sigma_{s\colon \mathrm{Var}(\vec{x})\to A} |(X, m)_{\vec{y}=s(\vec{x})}| = |(X, m)|.$$

If $\vec{x}, \vec{y}, \vec{z}$ are variable sequences, then $\vec{y} \perp\!\!\!\perp_{\vec{x}} \vec{z}$ is a *probabilistic conditional independence atom* with the satisfaction relation defined as

$$\mathfrak{A} \models_{(X,m)} \vec{y} \perp\!\!\!\perp_{\vec{x}} \vec{z}$$

if for all $s\colon \mathrm{Var}(\vec{x}\vec{y}\vec{z}) \to A$ it holds that

$$|(X, m)_{\vec{x}\vec{y}=s(\vec{x}\vec{y})}| \cdot |(X, m)_{\vec{x}\vec{z}=s(\vec{x}\vec{z})}| = |(X, m)_{\vec{x}\vec{y}\vec{z}=s(\vec{x}\vec{y}\vec{z})}| \cdot |(X, m)_{\vec{x}=s(\vec{x})}|. \tag{1}$$

We call atoms of the form $\vec{x} \perp\!\!\!\perp_{\emptyset} \vec{y}$ *probabilistic marginal independence atoms*, written as the shorthand $\vec{x} \perp\!\!\!\perp \vec{y}$. Note that we obtain the following satisfaction relation for $\vec{x} \perp\!\!\!\perp \vec{y}$:

$$\mathfrak{A} \models_{(X,m)} \vec{x} \perp\!\!\!\perp \vec{y} \text{ if for all } \; s\colon \mathrm{Var}(\vec{x}\vec{y}) \to A, \qquad (2)$$
$$\frac{|(X,m)_{\vec{x}=s(\vec{x})}| \cdot |(X,m)_{\vec{y}=s(\vec{y})}|}{|(X,m)|} = |(X,m)_{\vec{x}\vec{y}=s(\vec{x}\vec{y})}|.$$

Multiteams (X, m) induce a natural probability distribution p over the assignments of X. Namely, we define $p\colon X \to [0,1]$ such that

$$p(s) = \frac{m(s)}{\sum_{s \in X} m(s)}. \qquad (3)$$

The probability that a tuple of (random) variables $\vec{x}$ takes value $\vec{a}$, written $\Pr(\vec{x} = \vec{a})$, is then

$$\sum_{\substack{s \in X, \\ s(\vec{x}) = \vec{a}}} p(s).$$

It is now easy to see that $\mathfrak{A} \models_{(X,m)} \vec{y} \perp\!\!\!\perp_{\vec{x}} \vec{z}$ if and only if, for all $\vec{a}\vec{b}\vec{c}$,

$$\Pr(\vec{y} = \vec{b}, \vec{z} = \vec{c} | \vec{x} = \vec{a}) = \Pr(\vec{y} = \vec{b} | \vec{x} = \vec{a}) \Pr(\vec{z} = \vec{c} | \vec{x} = \vec{a}),$$

that is, the probability of $\vec{y} = \vec{b}$ is independent of the probability of $\vec{z} = \vec{c}$, given $\vec{x} = \vec{a}$. Analogously, a probabilistic inclusion atom $\vec{x} \leq \vec{y}$ indicates that $\Pr(\vec{x} = \vec{a}) = \Pr(\vec{y} = \vec{a})$ for all values $\vec{a}$, and a probabilistic independence atom of the form $\vec{x} \perp\!\!\!\perp \vec{x}$ that $\Pr(\vec{x} = \vec{a}) = 1$ for some value $\vec{a}$. Note that such atoms have been studied in the literature under the name of *constancy atoms* [7].

Example 8 Strict and lax multiteam semantics give rise to different interpretations of multiteams. Consider the formula $\phi := \exists x \psi$ where $\psi := y \leq x \wedge x \perp\!\!\!\perp y$ and the question whether a given multiteam (X, m) satisfies ϕ.

In the strict semantics framework, it is natural to perceive (X, m) as a table in which each $s \in X$ occurs $m(s)$ many times and the values for x are missing. For instance, (X, m) may store incomplete discrete information about events of an experiment. Then (X, m) satisfies ϕ if and only if the missing values of x can be filled in such a way that ψ holds. In the strict semantics setting, however, satisfaction of a formula is sensitive to the actual multiplicities of assignments, not only their proportions in the multiteam. Figure 2 represents two multiteams (X, m) and (X, n) where $X = \{s_1, s_2\}$ with $s_i\colon y \mapsto i$, $m(s_i) = i$, and $n(s_i) = 3 \cdot i$. Then (X, m) falsifies ϕ, since $y \leq x$ enforces two occurrences of $x = 1$ and one occurrence of $x = 2$, in which case $x \perp\!\!\!\perp y$ cannot hold. On the other hand, (X, n) satisfies ϕ, since $(X, n)[F/x]$, depicted in Fig. 2, satisfies ψ.

In the lax semantics framework, it is more natural to identify (X, m) with its induced probability distribution p obtained from (3) (see Fig. 3), for satisfaction by (X, m) is more often invariant of the multiplicities $m(s)$ depending only on the ratios $p(s)$. In our example case (X, m) and (X, n) share the same distribution and both satisfy ϕ. For both multiteams, satisfaction of ϕ is verified by taking a constant function G that maps everything to $(\{1, 2\}, \ell)$, where ℓ is a multiplicity function that associates 1 and 2 with multiplicities 1 and 2, respectively.

Fig. 2 Representations of multiteams (X, m), (X, n), and $(X, n)[F/x]$ with "missing information"

(X,m)	y	x
s_1	1	
s_2	2	
s_2	2	

(X,n)	y	x
s_1	1	
s_1	1	
s_1	1	
s_2	2	
s_2	2	
s_2	2	
s_2	2	
s_2	2	
s_2	2	

$(X,n)[F/x]$	y	x
s_1'	1	1
s_1''	1	2
s_1'''	1	2
s_2'	2	1
s_2''	2	1
s_2'''	2	2
s_2''''	2	2
s_2'''''	2	2
s_2''''''	2	2

3.3 Basic properties

The set of *free variables* of a formula $\varphi \in \mathsf{FO}(\mathcal{C})$, denoted by $\mathrm{Fr}(\varphi)$, is defined in the obvious manner as in first-order logic. In particular, we define

$$\mathrm{Fr}(\vec{x} \subseteq \vec{y}) := \mathrm{Fr}(\vec{x} \leq \vec{y}) := \mathrm{Fr}\big(=(\vec{x}, \vec{y})\big) := \{\vec{x}, \vec{y}\}$$
$$\mathrm{Fr}\big(\vec{y} \perp\!\!\!\perp_{\vec{x}} \vec{z}\big) := \mathrm{Fr}\big(\vec{y} \perp_{\vec{x}} \vec{z}\big) := \{\vec{x}, \vec{y}, \vec{z}\}.$$

For $V \subseteq \mathrm{Dom}(X)$, we define $(X, m) \upharpoonright V := (X \upharpoonright V,\ n)$, where

$$n(s) := \sum_{\substack{s' \in X, \\ s' \upharpoonright V = s}} m(s').$$

We briefly recollect from the literature of team semantics the definitions and results of the basic properties locality, union closure, and flatness. For an overview on the structural properties in team semantics and multiteam semantics setting, see Figs. 4 and 5, respectively.

Definition 9 Let $\mathcal{L}$ be a logic under team semantics, φ be a formula, X, Y be teams, and V be a set of variables such that $\mathrm{Fr}(\varphi) \subseteq V \subseteq \mathrm{Dom}(X)$.

Then $\mathcal{L}$ is

- *local* if $\mathfrak{A} \models_X \varphi$ if and only if $\mathfrak{A} \models_{X \upharpoonright V} \varphi$,
- *union closed* if $\mathfrak{A} \models_X \varphi$ and $\mathfrak{A} \models_Y \varphi$ implies $\mathfrak{A} \models_{X \cup Y} \varphi$, and
- *flat* if $\mathfrak{A} \models_X \varphi$ if and only if $\mathfrak{A} \models_{\{s\}} \varphi$ for all $s \in X$.

(X,m) and (X,n)	y	$p(s_i)$
s_1	1	1/3
s_2	2	2/3

$(X,m)[G/x]$ and $(X,n)[G/x]$	y	x	$p(s_i)$
s_1'	1	1	1/9
s_1''	1	2	2/9
s_2'	2	1	2/9
s_2''	2	2	4/9

Fig. 3 Representations of multiteams (X, m), (X, n), $(X, m)[G/x]$, and $(X, n)[G/x]$ using the induced probability distribution from (3)

property	logic	lax	strict
locality	$\mathsf{FO}(=(\cdot))$	✓	✓
	$\mathsf{FO}(\subseteq, \perp_c)$, $\mathsf{FO}(\subseteq)$, $\mathsf{FO}(\perp_c)$	✓	×
flatness	FO	✓	✓
	$\mathsf{FO}(=(\cdot))$, $\mathsf{FO}(\subseteq)$, $\mathsf{FO}(\perp_c)$	×	×
union closure	FO	✓	✓
	$\mathsf{FO}(\subseteq)$	✓	×
	$\mathsf{FO}(=(\cdot))$, $\mathsf{FO}(\perp_c)$	×	×

Fig. 4 Structural properties overview in team setting

Proposition 10 *Under lax team semantics*

- $\mathsf{FO}(\subseteq, =(\cdot), \perp_c)$ *is local* [7, 34],
- $\mathsf{FO}(\subseteq)$ *is union closed* [7], *and*
- FO *is flat* [34].

Under strict semantics

- $\mathsf{FO}(=(\cdot))$ *is local* [34],
- FO *is union closed, and*
- FO *is flat* [7].

Note above that union closure follows from flatness. It is now easy to show that the logics having the above properties are maximal with respect to the used atoms $\subseteq$, $=(\cdot)$, $\perp_c$. The only nontrivial case is to show that under strict semantics $\mathsf{FO}(\subseteq)$ is not union closed. The analogous result was recently established for propositional inclusion logic in [18]. The following example, a modification of [18, Example 4], shows that under strict semantics $\mathsf{FO}(\subseteq)$ is not union closed.

Example 11 Let P be a unary proposition symbol and consider a model $\mathfrak{A}$ of vocabulary $\{P\}$ and domain $\{a, b\}$ such that $P^{\mathfrak{A}} = \{a\}$. Assignments s_1, s_2, and s_3 are depicted in Fig. 6. Define $\varphi := (P(x) \wedge x \subseteq z) \vee (P(y) \wedge y \subseteq z)$. Note that under strict semantics $\mathfrak{A} \models_{\{s_1,s_2\}} \varphi$ and $\mathfrak{A} \models_{\{s_2,s_3\}} \varphi$ but $\mathfrak{A} \not\models_{\{s_1,s_2,s_3\}} \varphi$.

Observe that under strict *team* semantics $\mathsf{FO}(\subseteq)$ does not satisfy locality [7], whereas we will see that (Proposition 13), by moving to *multiteam* semantics, locality can be regained

property	logic	lax	strict
locality	$\mathsf{FO}(\leq, \perp\!\!\!\perp_c, =(\cdot), \subseteq, \perp_c)$	✓	✓
flatness	FO	✓	✓
	$\mathsf{FO}(\mathcal{C})$, $\emptyset \neq \mathcal{C} \subseteq \{\leq, \perp\!\!\!\perp_c, =(\cdot), \subseteq, \perp_c\}$	×	×
weak flatness	$\mathsf{FO}(=(\cdot))$	✓	✓
	$\mathsf{FO}(=(\cdot), \subseteq, \perp_c)$, $\mathsf{FO}(\subseteq)$, $\mathsf{FO}(\perp_c)$	✓	×
	$\mathsf{FO}(\perp\!\!\!\perp_c)$, $\mathsf{FO}(\leq)$	×	×
union closure	$\mathsf{FO}(\leq, \subseteq)$	✓	✓
	$\mathsf{FO}(\perp\!\!\!\perp_c)$, $\mathsf{FO}(=(\cdot))$, $\mathsf{FO}\perp_c)$	×	×

Fig. 5 Structural properties overview in multiteam setting

Fig. 6 Illustration of assignments s_1, s_2, and s_3 for Example 11

	x	y	z
s_1	a	b	b
s_2	a	a	a
s_3	b	a	b

for both semantics. Consequently, the shift from teams to multiteams fixes a problematic property of strict team semantics.

Definition 12 (Locality) Let $\mathcal{L}$ be some logic, $\mathfrak{A}$ be a multistructure, (X, m) be a multiteam, and V be a set of variables such that $\mathrm{Fr}(\varphi) \subseteq V \subseteq \mathrm{Dom}(X)$. We say $\mathcal{L}$ is *local* (in the multiteam setting) if for all $\varphi \in \mathcal{L}$ the equivalence $\mathfrak{A} \models_{(X,m)} \varphi \Leftrightarrow \mathfrak{A} \models_{(X,m)\upharpoonright V} \varphi$ holds.

The following result holds by easy structural induction.

Proposition 13 *Under lax and strict multiteam semantics,* $\mathsf{FO}(\leq, \perp\!\!\!\perp_c, =(\cdot), \subseteq, \perp_c)$ *is local.*

Example 14 Let us illustrate why locality fails for strict team semantics whereas it holds for its multiteam variant. Consider a disjunction $\phi := x \subseteq y \vee y \subseteq z$. The team X from Fig. 7 clearly satisfies ϕ according to the strict team semantics. However, the formula is not true over $X \upharpoonright \{x, y, z\}$ which merges two assignments from X into one. In multiteam semantics, strict or lax, such collapses do not happen, since the definition of restriction in multiteam setting preserves multiplicities. The restriction of a multiteam thus bears resemblance to the marginalisation of a probability distribution where probabilities are preserved analogously.

The flatness property translates also to the multiteam setting. Recall that over teams this property indicates that for checking the satisfaction of a formula it suffices to consult only the singleton subsets of teams.

Definition 15 (Flatness) We say that a formula φ is *flat* (in multiteam setting) if for all multistructures $\mathfrak{A}$ and for all multiteams (X, m) it holds that

$$\mathfrak{A} \models_{(X,m)} \varphi \quad \Leftrightarrow \quad \forall s \in X : \mathfrak{A} \models_{(\{s\},1)} \varphi,$$

where 1 is the constant fuction that maps everything to 1. A logic is called *flat* if every formula of this logic is flat.

It is easy to prove that, analogous to the team semantics setting, the logic $\mathsf{FO}(\mathcal{C}), \mathcal{C} \subseteq \{\leq, \perp\!\!\!\perp_c, =(\cdot), \subseteq, \perp_c\}$, is flat (in the multiteam setting) only when $\mathcal{C} = \emptyset$. It turns out that by

Fig. 7 A team X and its restriction to variables x, y, z for Example 14

X			
x	y	z	v
1	0	3	0
0	1	2	1
0	1	2	2
3	2	1	3

$X \upharpoonright \{x, y, z\}$		
x	y	z
1	0	3
0	1	2
3	2	1

weakening the concept of flatness an interesting property of multiteams emerges. We call a formula weakly flat if it is insensitive to multiplicatives other than 0 and 1.

Definition 16 (Weak flatness) We say that a formula φ is *weakly flat* if for all multistructures $\mathfrak{A}$ and for all multiteams (X, m) it holds that

$$\mathfrak{A} \models_{(X,m)} \varphi \quad \Leftrightarrow \quad \mathfrak{A} \models_{(X,n)} \varphi,$$

where n agrees with m on all s with $m(s) = 0$, and otherwise maps all s to 1. The multiteam (X, n) is then called the *weak flattening* of (X, m). A logic is called *weakly flat* if every formula of this logic is weakly flat.

Intuitively, the flatness property precludes the possibility of expressing any dependencies (standard or probabilistic), whilst the weak flatness precludes only the possiblity of expressing probabilistic dependencies. An example of this is the following proposition which can be proved by structural induction (see also Fig. 5).

Proposition 17 *Under lax multiteam semantics,* $\mathsf{FO}(=(\cdot), \subseteq, \perp_c)$ *is weakly flat.*
Under strict multiteam semantics, $\mathsf{FO}(=(\cdot))$ *is weakly flat.*

On the other hand, under strict multiteam semantics, the logics $\mathsf{FO}(\perp_c)$ and $\mathsf{FO}(\subseteq)$ are not weakly flat as illustrated in Examples 18 and 19. Likewise, probabilistic dependencies do not satisfy weak flatness as shown in Example 20. Thus the logics $\mathsf{FO}(\perp\!\!\!\perp_c)$ and $\mathsf{FO}(\leq)$ are weakly flat in neither strict nor lax multiteam semantics. Also consequently, neither the atom $\perp\!\!\!\perp_c$ nor $\leq$ can be expressed in $\mathsf{FO}(=(\cdot), \subseteq, \perp_c)$.

Example 18 The multiteam (X, m), illustrated in Fig. 8, satisfies $(x \subseteq z) \vee (y \subseteq z)$ in strict semantics but its weak flattening (X, n) does not.

Example 19 The multiteam (Y, m), illustrated in Fig. 9, satisfies $(x \perp x') \vee (y \perp y')$ in strict semantics but its weak flattening (Y, n) does not.

Example 20 The multiteam (Z, m), illustrated in Fig. 10, does not satisfy $x \perp\!\!\!\perp y$ but its weak flattening (Z, n) does. Likewise (Z, m) does not satisfy $x \leq y$ but its weak flattening (Z, n) does.

Definition 21 (Union closure) A formula φ is called *union closed* (in the multiteam setting) if for all multistructures $\mathfrak{A}$ and all multiteams (X, m), (Y, n): if $\mathfrak{A} \models_{(X,m)} \varphi$ and $\mathfrak{A} \models_{(Y,n)} \varphi$, then $\mathfrak{A} \models_{(Z,h)} \varphi$, where $(Z, h) = (X, m) \uplus (Y, n)$. A logic is called union closed if all its formulas are union closed.

(X, m)	x	y	z	$m(s_i)$
s_1	0	0	1	2
s_2	1	2	0	1
s_3	2	1	0	1

(X, n)	x	y	z	$n(s_i)$
s_1	0	0	1	1
s_2	1	2	0	1
s_3	2	1	0	1

Fig. 8 Assignments for teams in Example 18

(Y, m)	x	x'	y	y'	$m(s_i)$
s_1	1	1	1	1	2
s_2	1	0	2	2	1
s_3	0	1	3	3	1
s_4	0	0	4	4	1
s_5	2	2	1	0	1
s_6	3	3	0	1	1
s_7	4	4	0	0	1

(Y, n)	x	x'	y	y'	$n(s_i)$
s_1	1	1	1	1	1
s_2	1	0	2	2	1
s_3	0	1	3	3	1
s_4	0	0	4	4	1
s_5	2	2	1	0	1
s_6	3	3	0	1	1
s_7	4	4	0	0	1

Fig. 9 Assignments for teams in Example 19

It is easy to show, by induction on the structure of formulas, that $\mathsf{FO}(\leq, \subseteq)$ satisfies union closure.

Proposition 22 *Under lax and strict multiteam semantics,* $\mathsf{FO}(\leq, \subseteq)$ *is union closed.*

It is trivial to establish that none of the atoms $\perp\!\!\!\perp_c$, $=(\cdot)$, $\perp_c$ is in general union closed as singleton multiteams always satisfy these atoms.

3.4 Database and probabilistic dependencies

One can also study the usual dependency notions of database theory in the multiteam semantics setting.

Definition 23 Let $\mathfrak{A}$ be a multistructure with domain (A, n), (X, m) a multiteam over $\mathfrak{A}$, and φ of the form $=(\vec{x}, \vec{y})$, $\vec{x} \subseteq \vec{y}$, or $\vec{y} \perp_{\vec{x}} \vec{z}$. Then the satisfaction relation $\models_{(X,m)}$ is defined as follows:

$$\mathfrak{A} \models_{(X,m)} \varphi \;\Leftrightarrow\; \mathfrak{A}^+ \models_{X^+} \varphi,$$

where X^+ is the team $\{s \in X \mid m(s) \geq 1\}$ and $\mathfrak{A}^+$ is the first-order structure with domain $\{s \in A \mid n(s) \geq 1\}$ obtained from $\mathfrak{A}$ in the obvious manner.

First we notice that the known translation of dependence atoms to independence atoms (see Grädel et al. [12]) works also in the probabilistic case.

Proposition 24 *Let* $\mathfrak{A}$ *be a multistructure,* (X, m) *a multiteam over* $\mathfrak{A}$*, and* $\vec{x}, \vec{y}$ *tuples of variables. Then* $\mathfrak{A} \models_{(X,m)} \vec{y} \perp\!\!\!\perp_{\vec{x}} \vec{y} \;\Leftrightarrow\; \mathfrak{A} \models_{(X,m)} =(\vec{x}, \vec{y})$.

Fig. 10 Multiteams (Z, m) and (Z, n) for Example 20

(Z, m)	x	y	$m(s_i)$
s_0	0	0	1
s_1	0	1	2
s_2	1	0	1
s_3	1	1	1

(Z, n)	x	y	$n(s_i)$
s_0	0	0	1
s_1	0	1	1
s_2	1	0	1
s_3	1	1	1

Proof From the truth definition we obtain that

$$\mathfrak{A} \models_{(X,m)} \vec{y} \perp\!\!\!\perp_{\vec{x}} \vec{y} \;\Leftrightarrow\; \text{for all } s\colon \mathrm{Var}(\vec{x}\vec{y}) \to A \text{ with } (X,m)_{\vec{x}\vec{y}=s(\vec{x}\vec{y})} \neq \emptyset, \quad |(X,m)_{\vec{x}\vec{y}=s(\vec{x}\vec{y})}| = |(X,m)_{\vec{x}=s(\vec{x})}|. \tag{4}$$

The result then follows since $\mathfrak{A} \models_{(X,m)} {=}(\vec{x},\vec{y})$ if and only if the right-hand side of (4) holds. For this, first note that the right-hand side of (4) fails if and only if there is a function $s\colon \mathrm{Var}(\vec{x}\vec{y}) \to A$ with $(X,m)_{\vec{x}\vec{y}=s(\vec{x}\vec{y})} \neq \emptyset$ such that $|(X,m)_{\vec{x}\vec{y}=s(\vec{x}\vec{y})}| < |(X,m)_{\vec{x}=s(\vec{x})}|$. This holds if and only if there exists two distinct functions $s, s'\colon \mathrm{Var}(\vec{x}\vec{y}) \to A$ with $s(\vec{x}) = s'(\vec{x})$ and $(X,m)_{\vec{x}\vec{y}=s(\vec{x}\vec{y})} \neq \emptyset \neq (X,m)_{\vec{x}\vec{y}=s'(\vec{x}\vec{y})}$. □

Note that the restriction of Proposition 24 to marginal independence states that

$$\mathfrak{A} \models_{(X,m)} \vec{x} \perp\!\!\!\perp \vec{x} \quad\Leftrightarrow\quad \mathfrak{A} \models_{(X,m)} {=}(\vec{x}).$$

It is left open whether one can define inclusion or conditional independence atoms in $\mathsf{FO}(\perp\!\!\!\perp_c, \leq)$. However, it can be shown that over constant multiplicity functions conditional independence atoms φ coincide with their probabilistic counterparts whenever $\mathrm{Var}(\varphi) = \mathrm{Dom}(X)$. To this end, let us first prove the following simple lemma. This lemma entails that any probabilistic independence atom can be expressed as a conjunction of $\vec{y} \perp\!\!\!\perp_{\vec{x}} \vec{z}$ and $\vec{v} \perp\!\!\!\perp_{\vec{x}} \vec{v}$ where $\vec{x}, \vec{y}, \vec{z}, \vec{v}$ are pairwise disjoint sequences of variables.

Note that by ${}^{\vec{x}}A$ we denote the team of all assignments $\mathrm{Var}(\vec{x}) \to A$.

Lemma 25 *Let $\mathfrak{A}$ be a multistructure and (X,m) a multiteam over $\mathfrak{A}$. Then*

(i) $\mathfrak{A} \models_{(X,m)} \vec{y} \perp\!\!\!\perp_{\vec{x}} \vec{z} \;\Leftrightarrow\; \mathfrak{A} \models_{(X,m)} \big(\vec{y} \setminus \vec{x} \perp\!\!\!\perp_{\vec{x}} \vec{z} \setminus \vec{x}\big)$,
(ii) $\mathfrak{A} \models_{(X,m)} \vec{y} \perp\!\!\!\perp_{\vec{x}} \vec{z} \;\Leftrightarrow\; \mathfrak{A} \models_{(X,m)} \big(\vec{y} \setminus \vec{z} \perp\!\!\!\perp_{\vec{x}} \vec{z} \setminus \vec{y}\big) \wedge \big(\vec{y} \cap \vec{z} \perp\!\!\!\perp_{\vec{x}} \vec{y} \cap \vec{z}\big)$.

Proof **Item (i)** The truth definition in (1) is symmetric, and hence it suffices to show that $\mathfrak{A} \models_{(X,m)} \vec{y}x \perp\!\!\!\perp_{\vec{x}} \vec{z} \Leftrightarrow \mathfrak{A} \models_{(X,m)} \vec{y} \perp\!\!\!\perp_{\vec{x}} \vec{z}$ whenever x is listed in $\vec{x}$. This follows since ${}^{\vec{x}\vec{y}x\vec{z}}A = {}^{\vec{x}\vec{y}\vec{z}}A$, and the equation in (1) remains the same after removing x.

Item (ii) First show that $\mathfrak{A} \models_{(X,m)} \vec{y} \perp\!\!\!\perp_{\vec{x}} \vec{z}$ implies $\mathfrak{A} \models_{(X,m)} \big(\vec{y} \cap \vec{z} \perp\!\!\!\perp_{\vec{x}} \vec{y} \cap \vec{z}\big)$. For this, it suffices to show that $\mathfrak{A} \models_{(X,m)} \vec{y}u \perp\!\!\!\perp_{\vec{x}} \vec{z}$ implies $\mathfrak{A} \models_{(X,m)} \vec{y} \perp\!\!\!\perp_{\vec{x}} \vec{z}$, for u not listed in $\vec{x}\vec{y}\vec{z}$. This follows since for all $s \in {}^{\vec{x}\vec{y}\vec{z}}A$,

$$\begin{aligned}
&|(X,m)_{\vec{x}\vec{z}=s(\vec{x}\vec{z})}| \cdot |(X,m)_{\vec{x}\vec{y}=s(\vec{x}\vec{y})}| \\
&\quad= |(X,m)_{\vec{x}\vec{z}=s(\vec{x}\vec{z})}| \cdot \Sigma_{a\in A} |(X,m)_{\vec{x}\vec{y}u=s(\vec{x}\vec{y})a}| \\
&\quad= \Sigma_{a\in A} \big(|(X,m)_{\vec{x}\vec{z}=s(\vec{x}\vec{z})}| \cdot |(X,m)_{\vec{x}\vec{y}u=s(\vec{x}\vec{y})a}|\big) \\
&\quad= \Sigma_{a\in A} \big(|(X,m)_{\vec{x}=s(\vec{x})}| \cdot |(X,m)_{\vec{x}\vec{y}\vec{z}u=s(\vec{x}\vec{y}\vec{z})a}|\big) \\
&\quad= |(X,m)_{\vec{x}=s(\vec{x})}| \cdot \Sigma_{a\in A} |(X,m)_{\vec{x}\vec{y}\vec{z}u=s(\vec{x}\vec{y}\vec{z})a}| \\
&\quad= |(X,m)_{\vec{x}=s(\vec{x})}| \cdot |(X,m)_{\vec{x}\vec{y}\vec{z}=s(\vec{x}\vec{y}\vec{z})}|,
\end{aligned}$$

where in the third equation we apply the assumption that $\mathfrak{A} \models_{(X,m)} \vec{y}a \perp\!\!\!\perp_{\vec{x}} \vec{z}$.

For the claim it now suffices to show that $\mathfrak{A} \models_{(X,m)} \vec{y} \perp\!\!\!\perp_{\vec{x}} \vec{z} \;\Leftrightarrow\; \mathfrak{A} \models_{(X,m)} \big(\vec{y} \setminus \vec{z} \perp\!\!\!\perp_{\vec{x}} \vec{z} \setminus \vec{y}\big)$ whenever $\mathfrak{A} \models_{(X,m)} \big(\vec{y} \cap \vec{z} \perp\!\!\!\perp_{\vec{x}} \vec{y} \cap \vec{z}\big)$. This follows directly from the truth definition since by (4) for all $s \in {}^{\vec{x}\vec{v}}A$ with $(X,m)_{\vec{x}\vec{v}=s(\vec{x}\vec{v})} \neq \emptyset$:

$$|(X,m)_{\vec{x}\vec{v}=s(\vec{x}\vec{v})}| = |(X,m)_{\vec{x}=s(\vec{x})}|,$$

for $\vec{v} := \vec{x} \cap \vec{y}$. □

If $\vec{x}, \vec{y}, \vec{z}$ are pairwise disjoint, then $\vec{y} \perp\!\!\!\perp_{\vec{x}} \vec{z}$ corresponds to the generalised embedded multivalued dependency $\vec{x} \multimap\!\!\to \vec{y} \mid \vec{z}$ that is defined over extended relational data models (i.e., relational data models equipped with a multiplicity function) using semantics that coincide with that of Definition 7 [37, 38]. It was shown by Wong [37] that the generalised multivalued dependency $\vec{x} \multimap\!\!\to \vec{y}$ holds in an extended relational data model if and only if the underlying relational model satisfies the multivalued dependency $\vec{x} \twoheadrightarrow \vec{y}$. This is stated in the following theorem reformulated into the framework of this article.

Theorem 26 [37] *Let $\mathfrak{A}$ be a multistructure, X a team over $\mathfrak{A}$, and $\vec{y} \perp\!\!\!\perp_{\vec{x}} \vec{z}$ a probabilistic conditional independence atom such that* $\mathrm{Var}(\vec{y} \perp\!\!\!\perp_{\vec{x}} \vec{z}) = \mathrm{Dom}(X)$ *and $\vec{x}, \vec{y}, \vec{z}$ are pairwise disjoint. Let 1 denote the constant function that maps all assignments of X to 1. Then* $\mathfrak{A} \models_{(X,1)} \vec{y} \perp\!\!\!\perp_{\vec{x}} \vec{z} \Leftrightarrow \mathfrak{A} \models_{(X,1)} \vec{y} \perp_{\vec{x}} \vec{z}$.

Using Lemma 25 the restriction that $\vec{x}, \vec{y}, \vec{z}$ are disjoint can be now removed.

Proposition 27 *Let $\mathfrak{A}$ be a multistructure, X a team over $\mathfrak{A}$, and $\vec{y} \perp\!\!\!\perp_{\vec{x}} \vec{z}$ a probabilistic conditional independence atom such that* $\mathrm{Var}(\vec{y} \perp\!\!\!\perp_{\vec{x}} \vec{z}) = \mathrm{Dom}(X)$. *Then* $\mathfrak{A} \models_{(X,1)} \vec{y} \perp\!\!\!\perp_{\vec{x}} \vec{z} \Leftrightarrow \mathfrak{A} \models_{(X,1)} \vec{y} \perp_{\vec{x}} \vec{z}$.

Proof First note that by Proposition 24 and Lemma 25, $\vec{y} \perp\!\!\!\perp_{\vec{x}} \vec{z}$ is equivalent in multiteam semantics to $\big(\vec{y} \setminus \vec{x}\vec{z} \perp\!\!\!\perp_{\vec{x}} \vec{z} \setminus \vec{x}\vec{y}\big) \wedge {=}(\vec{x}, \vec{y} \cap \vec{z})$. Moreover, it is known that in team semantics $\vec{y} \perp_{\vec{x}} \vec{z}$ is equivalent to $(\vec{y} \setminus \vec{x}\vec{z} \perp_{\vec{x}} \vec{z} \setminus \vec{x}\vec{y}) \wedge {=}(\vec{x}, \vec{y} \cap \vec{z})$ [12]. Hence the claim follows by Theorem 26. □

Note that $\vec{y} \perp\!\!\!\perp_{\vec{x}} \vec{z}$ implies $\vec{y} \perp_{\vec{x}} \vec{z}$ also over arbitrary multiplicity functions, since non-emptiness of $(X, m)_{\vec{x}\vec{y}=s(\vec{x}\vec{y})}$ and $(X, m)_{\vec{x}\vec{z}=s(\vec{x}\vec{z})}$ implies non-emptiness of $(X, m)_{\vec{x}\vec{y}\vec{z}=s(\vec{x}\vec{y}\vec{z})}$ by the truth definition in (1). The converse however does not hold; the multiteam (Y, m) depicted in Fig. 11 satisfies $x \perp y$ but violates $x \perp\!\!\!\perp y$.

3.5 Probabilistic notions in team semantics

In this section we examine probabilistic independence and inclusion logic in the team semantics setting. We restrict attention to lax semantics. Note that all the models considered in this section are usual first-order structures.

Satisfaction of probabilistic atoms in team semantics setting is defined by adding a constant multiplicity function.

Definition 28 Let $\mathfrak{A}$ be a model, X be a team over $\mathfrak{A}$, and φ be a probabilistic atom of the form $\vec{y} \perp\!\!\!\perp_{\vec{x}} \vec{z}$ or $\vec{x} \leq \vec{y}$. Then the satisfaction relation $\models_X$ is defined as follows:

$$\mathfrak{A} \models_X \varphi \Leftrightarrow \mathfrak{A} \models_{(X,1)} \varphi,$$

where 1 is the constant function that maps all assignments of X to 1.

Fig. 11 A multiteam (Y, m) satisfying $x \perp y$ but violating $x \perp\!\!\!\perp y$

(Y, m)			
	x	y	$m(s_i)$
s_0	0	0	2
s_1	0	1	1
s_2	1	0	1
s_3	1	1	1

The next theorem shows that, since probabilistic inclusion and independence atoms are expressible (in the team semantics setting) in $\mathsf{FO}(\perp_c)$ relative to teams of fixed domain, their addition does not increase the expressive power of $\mathsf{FO}(\perp_c)$.

Theorem 29 *Let $\varphi \in \mathsf{FO}(\leq, \perp\!\!\!\perp_c, =(\cdot), \subseteq, \perp_c)$ be a sentence. Then there exists a sentence $\varphi' \in \mathsf{FO}(\perp_c)$ such that for all models $\mathfrak{A}$ the equivalence $\mathfrak{A} \models \varphi \Leftrightarrow \mathfrak{A} \models \varphi'$ holds.*

Proof First note that inclusion and dependence atoms can be expressed in $\mathsf{FO}(\perp_c)$ [7, 12]. Also it is easy to see that one can construct existential second-order logic sentences that capture probabilistic inclusion and conditional independence atoms over teams of fixed domain. Namely, for all φ of the form $\vec{y} \perp\!\!\!\perp_{\vec{x}} \vec{z}$ or $\vec{x} \leq \vec{y}$ and all $V \supseteq \mathrm{Fr}(\varphi)$, there exists an ESO sentence $\varphi^*(R)$, where R is a k-ary relation symbol for $k = |\mathrm{Var}(\varphi)|$, such that for all $\mathfrak{A}$ and X with $\mathrm{Dom}(X) = V$,

$$\mathfrak{A} \models_X \varphi \Leftrightarrow (\mathfrak{A}, \mathrm{Rel}(X)) \models \varphi^*(R),$$

where $\mathrm{Rel}(X) = \{(s(x_1), \ldots, s(x_k)) \mid s \in X\}$. All ESO-definable properties of teams translate into $\mathsf{FO}(\perp_c)$ [7], and hence the formula φ' can be constructed from φ by replacing each probabilistic atom with a correct $\mathsf{FO}(\perp_c)$-translation. □

Note that probabilistic inclusion atoms are not closed under (set) unions in team semantics, and hence they cannot be expressed in $\mathsf{FO}(\subseteq)$ as shown in the following example.

Example 30 Let $\mathfrak{A}$ be a first-order structure with domain $\{0, 1, 2\}$, and the following three assignments given $s := \{(x, 0), (y, 1), (z, 0)\}$, $s' := \{(x, 1), (y, 0), (z, 1)\}$, and $s'' := \{(x, 0), (y, 1), (z, 2)\}$. Define $X := \{s, s'\}$ and $Y := \{s', s''\}$. Now $\mathfrak{A} \models_X x \leq y$, $\mathfrak{A} \models_Y x \leq y$, but $\mathfrak{A} \not\models_{X \cup Y} x \leq y$.

4 Approximate operators

Next, we define an existential and a universal *approximate* operator which allows one to state truth of formulas not with respect to the *full* team but with respect to a *ratio* of the team. The main motivator for this approach is the important application in database theory to be able to model the truth of properties in databases that may contain some faulty data. Moreover, in practise, duplicates occur frequently in databases for a multitude of reasons. Thus the study of database dependencies, such as inclusion dependencies and foreign key constraints, in combination with approximate operators is an important topic as it explains inherent properties of a given dataset. In this section we consider multiteam semantics.

Definition 31 Let $\mathfrak{A}$ be a multistructure, and (X, m) a multiteam over $\mathfrak{A}$, and $p \in [0, 1]$ a rational number.

$$\mathfrak{A} \models_{(X,m)} \langle p \rangle \varphi \Leftrightarrow \exists (Y, n) \subseteq (X, m), |(Y, n)| \geq p \cdot |(X, m)| : \mathfrak{A} \models_{(Y,n)} \varphi,$$
$$\mathfrak{A} \models_{(X,m)} [p] \varphi \Leftrightarrow \forall (Y, n) \subseteq (X, m), |(Y, n)| \geq p \cdot |(X, m)| : \mathfrak{A} \models_{(Y,n)} \varphi.$$

The notion of an approximate dependence atom $=_p(\cdot)$, introduced by Väänänen [36], can be now seen as a special instantiation of the existential approximate operator: $=_{1-p}(\vec{x}, y)$ is equivalent to the formula $\langle p \rangle =(\vec{x}, y)$. The difference to Väänänen's proposal is the scope of approximation which is now extended to arbitrary formulae.

In the following we observe that distributivity does not hold in general with respect to $\langle p \rangle$.

Proposition 32 *It is not true that* $\langle p \rangle(\varphi \vee \psi) \equiv \langle p \rangle\varphi \vee \langle p \rangle\psi$.

Proof Let $\mathfrak{A}$ be the multistructure over the empty vocabulary with the domain $(\{0, 1, 2\}, 1)$, where 1 is the constant 1 multiplicity function. Then $\mathfrak{A} \models_{(X,1)} \langle \frac{2}{3} \rangle(x = y \vee x = z)$ but $\mathfrak{A} \not\models_{(X,1)} \langle \frac{2}{3} \rangle(x = y) \vee \langle \frac{2}{3} \rangle(x = z)$, where $(X, 1)$ is the multiteam depicted in the Fig. 12. □

The next simple observation states the distributivity of $[p]$ with respect to conjunction $\wedge$, as well as the merger of two $\langle p \rangle$-operators and two $[q]$-operators, respectively.

Observation 33 *The following equivalences hold:*

1. $[p](\varphi \wedge \psi) \equiv [p]\varphi \wedge [p]\psi$,
2. $\langle p \rangle(\langle q \rangle\varphi) = \langle p \cdot q \rangle\varphi$,
3. $[p]([q]\varphi) = [p \cdot q]\varphi$.

The next two examples show that both downward closure and union closure are violated by the approximate operator.

Example 34 Let $\mathfrak{A}$ be the multistructure over the empty vocabulary with domain $(\{0, 1, 2\}, 1)$, where 1 is the constant 1 multiplicity function. Then $\mathfrak{A} \models_{(X,1)} \langle \frac{1}{3} \rangle(x = y)$ but $\mathfrak{A} \not\models_{(Y,1)} \langle \frac{1}{3} \rangle(x = y)$, where $(Y, 1) \subseteq (X, 1)$ are the multiteams depicted in the Fig. 12.

Example 35 Let $\mathfrak{A}$ be the multistructure over the empty vocabulary with domain $(\{0, 1\}, 1)$, where 1 is the constant 1 multiplicity function. The multiteams (Z, m), (Z, n), (Z, k), (Z, ℓ) are depicted in the Fig. 12. Now $\mathfrak{A} \models_{(Z,k)} [\frac{2}{3}](x \leq y)$ and $\mathfrak{A} \models_{(Z,\ell)} [\frac{2}{3}](x \leq y)$. However $\mathfrak{A} \not\models_{(Z,n)} x \leq y$ and thus $\mathfrak{A} \not\models_{(Z,m)} [\frac{2}{3}](x \leq y)$, even though $(Z, k) \uplus (Z, l) = (Z, m)$.

Proposition 36 *Let* $\mathcal{L}$ *be a logic and* $\varphi \in \mathcal{L}$ *a formula. Then* $\langle p \rangle$ *preserves union closure (whereas* $[p]$ *does not), i.e.,* $\langle p \rangle\varphi$ *is union closed whenever* φ *is.*

$(X, 1)$

	x	y	z	$1(s_i)$
s_1	0	0	1	1
s_2	0	1	0	1
s_3	0	1	2	1

$(Y, 1)$

	x	y	z	$1(s_i)$
s_2	0	1	0	1
s_3	0	1	2	1

$(Z, \cdot)$

	x	y	$m(s_i)$	$n(s_i)$	$k(s_i)$	$\ell(s_i)$
s_1	0	1	1	0	1	0
s_2	1	0	1	1	1	0
s_3	0	0	1	1	0	1

Fig. 12 Assignments for multiteams in Examples 34 and 35

Proof Let $\mathfrak{A}$ be a multistructure and X, Y be multiteams of $\mathfrak{A}$. Assume that $\mathfrak{A} \models_X \langle p \rangle \varphi$ and $\mathfrak{A} \models_Y \langle p \rangle \varphi$ where φ is closed under unions. Then there are multiteams $X' \subseteq X$ and $Y' \subseteq Y$ such that $|X'| \geq p|X|$, $|Y'| \geq p|Y|$, and both $\mathfrak{A} \models_{X'} \varphi$ and $\mathfrak{A} \models_{Y'} \varphi$. Hence, $\mathfrak{A} \models_{X' \uplus Y'} \varphi$ by union closure, and since $|X' \uplus Y'| = |X'| + |Y'| \geq p|X| + p|Y| = p(|X| + |Y|) = p|X \uplus Y|$, it follows that $\mathfrak{A} \models_{X \uplus Y} \langle p \rangle \varphi$. □

Yet locality holds for this logic as witnessed by the following proposition. The proof is by induction.

Proposition 37 (Locality) *Let $\mathfrak{A}$ be a multistructure, (X, m) a multiteam, and V be a set of variables such that* $\mathrm{Fr}(\varphi) \subseteq V \subseteq \mathrm{Dom}(X)$. *Then, for all $\varphi \in \mathsf{FO}(\langle p \rangle, [p], \leq, \perp\!\!\!\perp_c, =(\cdot), \subseteq, \perp_c)$, the equivalence $\mathfrak{A} \models_{(X,m)} \varphi \Leftrightarrow \mathfrak{A} \models_{(X,m)\upharpoonright V} \varphi$ holds.*

5 On the complexity of approximate dependence logic

In the following we study the computational complexity of model checking in dependence logic enriched with the operator $\langle p \rangle$. The results hold under both team and multiteam semantics. To simplify notation we work with team semantics in this section. Analogously to [3], our results can be seen as a first step towards a systematic classification of the syntactic fragments of approximate dependence logic for which data complexity of model-checking is tractable/intractable.

We first define the model checking problem in the context of team semantics. We consider only Boolean queries, that is we define the model checking problem for a logic $\mathcal{L}$ as follows: given a model $\mathfrak{A}$, a team X of $\mathfrak{A}$, and a formula φ of $\mathcal{L}$, decide whether $\mathfrak{A} \models_X \varphi$ holds. There are three parameters to this problem: the model $\mathfrak{A}$, the team X, and the formula φ. Depending on which of these parameters are fixed, a different variant of the model checking problem arises. Here we consider two of these variants: the variant with a fixed formula (this is called *data complexity*), and a variant in which nothing is fixed (this is called *combined complexity*).

The following two theorems reveal that already very simple formulas of approximate dependence logic witness the NP-completeness of the data complexity of the logic.

Theorem 38 *Model checking for $\langle \frac{1}{3} \rangle (=(x, y) \wedge =(u, v))$ is NP-complete.*

Proof For the lower bound we give a polynomial many-one reduction from 3SAT inspired by a similar proof of Jarmo Kontinen [25]. Start with a formula $\varphi = \bigwedge_{i=1}^{m} \bigvee_{j=1}^{3} \ell_{i,j}$ where $\ell_{i,j}$ is the jth literal in the ith clause, i.e., either a variable x (said of parity 0) or its negation $\neg x$ (of parity 1). In the following we will construct a tuple (X, ψ) from φ such that $\varphi \in$ 3SAT if and only if $\mathfrak{A} \models_X \psi$. Here $\mathfrak{A}$ is the model over the empty vocabulary such that the domain of $\mathfrak{A}$ is the codomain of X. First we define the team X to be the set

$$X = \{(i, j, x, p) \mid \text{in the } i\text{th clause the } j\text{th literal is the variable } x \text{ with parity } p\},$$

where (i, j, x, p) denotes the assignment mapping the variables *clause*, *literal*, *variable*, *parity* to i, j, x, p, respectively. Technically the team X can be seen as an encoding of the given formula. For instance the formula $\varphi = (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3)$ would yield the team

$$X = \{(1, 1, x_1, 0), (1, 2, x_2, 1), (1, 3, x_3, 0), (2, 1, x_1, 1)(2, 2, x_2, 1), (2, 3, x_3, 1)\}.$$

The formula ψ is defined as

$$\langle \frac{1}{3} \rangle \big(=(\text{clause}, \text{literal}) \wedge =(\text{variable}, \text{parity}) \big).$$

Then intuitively speaking ψ states that one has to decide for each clause a satisfying literal and do this consistently, i.e., the corresponding assignment has to be consistent. At first, one selects exactly one-third of the elements in X such that for each clause a literal is chosen (i.e., *clause* will determine the value of *literal*). Then the *parity* of each *variable* is consistently chosen (i.e., *variable* will determine the value of *parity*). We will next formally prove that $\varphi \in \mathsf{3SAT}$ if and only if $\mathfrak{A} \models_X \psi$.

We first show that $\varphi \in \mathsf{3SAT} \Rightarrow \mathfrak{A} \models_X \psi$. Thus assume that $\varphi \in \mathsf{3SAT}$. Let θ be an assignment such that $\theta \models \varphi$. For each $1 \leq k \leq m$, let $i_k \in \{1, 2, 3\}$ be a number such that the literal ℓ_{k,i_k} in the kth clause of ψ is satisfied by θ, i.e., $\theta \models \ell_{k,i_k}$. Let $I := \{i_1, \ldots, i_k\}$. In the following we will show that $\mathfrak{A} \models_X \psi$ holds. Define

$X' := \{(k, j, v, p) \in X \mid j = i_k\}$. Clearly $|X'| = \frac{1}{3}|X|$.

Moreover it is easy to cheque that for any two $(j, i, v, p), (j', i', v', p') \in X'$

(a) $j = j'$ implies $i = i'$ (the clause determines the literal) and
(b) $v = v'$ implies $p = p'$ (the variable determines the parity).

Hence from (a) is it follows that $\mathfrak{A} \models_{X'} =(\text{clause}, \text{literal})$ and from (b) it follows that $\mathfrak{A} \models_{X'} =(\text{variable}, \text{parity})$. Since $|X'| = \frac{1}{3}|X|$, we obtain $\mathfrak{A} \models_X \psi$.

Now turn to the direction $\mathfrak{A} \models_X \psi \Rightarrow \varphi \in \mathsf{3SAT}$ and assume that $\mathfrak{A} \models_X \psi$. Thus there exists a team $X' \subseteq X$ such that $|X'| \geq \frac{1}{3}|X|$ is true and also it holds that $\mathfrak{A} \models_{X'} =(\text{clause}, \text{literal}) \wedge =(\text{variable}, \text{parity})$. Since $\mathfrak{A} \models_{X'} =(\text{clause}, \text{literal})$ we have that

$$(j, i, v, p), (j', i', v', p') \in X' \text{ and } j = j' \text{ imply } i = i'. \tag{5}$$

Analogously, since $\mathfrak{A} \models_{X'} =(\text{variable}, \text{parity})$ we have that

$$(j, i, v, p), (j', i', v', p') \in X' \text{ and } v = v' \text{ imply } p = p'. \tag{6}$$

From (1) we can deduce that $|X'| \leq \frac{1}{3}|X|$. Since $|X'| \geq \frac{1}{3}|X|$, we obtain that $|X'| = \frac{1}{3}|X|$. This together with (2) ensures that

$$\text{for each clause } j \text{ of } \psi \text{ there exits some } i, v, p \text{ such that } (j, i, v, p) \in X'. \tag{7}$$

It is now easy to construct from X' an assignment θ such that $\theta \models \varphi$.

Define

$$\theta(v) := \begin{cases} 1 & \text{if } (j, i, v, 0) \in X' \text{ for some } j, i \in \mathbb{N}, \\ 0 & \text{if } (j, i, v, 1) \in X' \text{ for some } j, i \in \mathbb{N}. \end{cases}$$

From (2) it follows that θ is well-defined, whereas (1) and (3) ensure that every clause of φ is satisfied by θ. Hence we have $\varphi \in \mathsf{3SAT}$.

For the NP upper bound, first observe that we can simply guess a subset X' of X such that $|X'| \geq \frac{1}{3}|X|$. Then we just have to cheque whether $\mathfrak{A} \models_{X'} =(\text{clause}, \text{literal}) \wedge =(\text{variable}, \text{parity})$ holds. This can be clearly done in polynomial time. □

The next theorem shows that NP-hard properties can be defined using very simple formulas even if the operator $\langle p \rangle$ is restricted to appear only in front of dependence atoms. It is worth noting that the data complexity of formulas addressed in Theorem 39 without the operator $\langle p \rangle$ is in NL by the results of [25].

Theorem 39 *Model checking for* $=(x, y) \vee (\langle \frac{7}{10} \rangle =(x, y) \wedge =(u, v))$ *is NP-complete.*

Proof The upper bound is established analogously to the proof of Theorem 38.

Now we turn to the lower bound. Here we will reduce from 3SAT through Max-2Sat, a well-known NP-hard optimisation problem whose decision variant is NP-complete. The problem asks given a 2CNF-formula φ and a number $k \in \mathbb{N}$, if at least k of the clauses of φ can be simultaneously satisfied [11]. Garey et al. describe a reduction f from 3SAT to the decision variant of Max-2Sat such that $\varphi \in$ 3SAT if and only if at least $\frac{7}{10}$ of the clauses of $f(\varphi)$ can be satisfied.

We will exploit this known reduction in the following way. The team X is constructed in the same way as in the proof of Theorem 38. The formula then is

$$=(\text{clause, literal}) \vee (=(\text{clause, literal}) \wedge \langle \frac{7}{10} \rangle =(\text{variable, parity})).$$

Let us briefly sketch the proof as it is quite similar to the one of Theorem 38. The first $\vee$ just "removes" the not needed half of the literals in the clauses. Then $=$(clause, literal) takes care of that in each clause exactly one literal is chosen whereas $\langle \frac{7}{10} \rangle =$(variable, parity) allows us to get down to the fraction of clauses which have to be satisfied, hence have to obey the dependence atom stating that the remaining variables have to be consistently chosen, i.e., variable determines parity. □

Currently the $\langle p \rangle$ operator is defined with respect to some value of $p \in [0, 1]$. We saw that it depicts the behaviour of a *ratio*. Yet we want to shortly discuss a different approach for this setting. Instead, we define $\langle p \rangle$ for values of $p \in \mathbb{N}$ hence p is now a natural number with the following meaning. A team X satisfies a formula $\langle p \rangle \varphi$ if there exists a team $Y \subseteq X$ of size $\geq p$ such that $Y \models \varphi$—similarly for $[p]$ the meaning would be that every team $Y \subseteq X$ of size $\geq p$ satisfies φ.

Sticking to this approach would allow one to state a similar result as for Theorem 38 and Theorem 39 but now for *combined complexity* as follows. Here one would just explicitly state the number of rows to be removed from the team, i.e., setting p to m in the constructed formula in the proof of Theorem 38. Regarding Theorem 39 in this setting the formula $f(\varphi)$ increases the number of clauses by factor 10 and therefore requires to set p to $\frac{7}{10} \cdot 10 \cdot m = 7 \cdot m$ where m is the number of clauses of the given 3CNF formula φ.

Remark 40 The lower bounds of Theorems 38 and 39 transfer directly to the multiteam setting. For the upper bounds, it might matter how the multiplicity functions of multiteams are encoded. However, if the multiplicities are encoded in any reasonable way, such as in unary or binary, then also the upper bounds of Theorems 38 and 39 transfer to the multiteam setting. In the unary case they transfer immediately; instead of guessing subsets of the team, we may directly guess subsets of the canonical set representative. In the binary case, for example, a subset of a multiteam (A, m) can be guessed by guessing, for each $a \in A$, a binary number smaller than $m(a)$. The NP upper bound then follows, since addition and comparison of binary numbers can be done in polynomial time.

6 Conclusion and future issues

To the best of the authors' knowledge, this article is the first serious approach to defining team semantics with respect to multisets for first-order dependence logic. We also initiate the study of probabilistic analogues of independence and inclusion logic. Additionally

the paper provides a first step into the study of a general approximation operator in the team semantics framework. We show several foundational properties of these newly defined formalisms and present some first computational complexity results for dependence logic with the approximation operator. We show that the introduction of the approximate operator enables us to encode NP-hard properties into the model checking problem (data complexity) of this logic even with only two dependence atoms, a single approximate operator, and a single conjunction. It is an interesting open question to study the computational properties of the analogously defined approximate inclusion logic.

It is often argued that, in team semantics setting, the strict semantics for disjunction is not natural for the failure of locality. In strict multiteam semantics, however, locality is regained. In team semantics setting, the lax disjunction $\varphi \vee \varphi$ is equivalent with φ if and only if φ union closed (with respect to set unions). The disjunction of lax multiteam semantics has a similar, but weaker, property: for multisets (X, m) and (Y, n), we call the set

$$\{(X \cup Y, t) \mid \forall s \in X \cup Y : n(s), m(s) \leq t(s) \leq n(s) + m(s)\}$$

the set of *weak unions* of (X, m) and (Y, n). In the multiteam setting, the lax disjunction $\varphi \vee \varphi$ is equivalent with φ if and only if φ is union closed with respect to weak unions. However, in multiteam setting, unions of multiteams are most naturally defined by using disjoint unions. In fact, in multiteam semantics, the strict disjunction $\varphi \vee \varphi$ is equivalent with φ if and only if φ is union closed with respect to disjoint unions. This suggests that the combination of strict disjunctions and lax existential quantifiers deserves to be studied as a possible candidate for a base of probabilistic logics in multiteam setting.

Heretofore a broad field around intuitionistic logic has developed. Intuitionistic logic can be seen as classical propositional logic without the law of excluded middle. One of the main concepts here is the *intuitionistic implication* $\rightarrow$. In the setting of team semantics it is defined as follows. Let $\mathfrak{A}$ be a structure and X be a team. Then $\mathfrak{A} \models_X \varphi \rightarrow \psi$ is true if and only if for all subsets $X' \subseteq X$ it holds that $\mathfrak{A} \models_{X'} \varphi$ implies $\mathfrak{A} \models_{X'} \psi$. The intuitionistic implication has been studied in the context of dependence logic, see, e.g., the work of Yang [39]. An approximate variant of this operator in our setting will yield a nice resemblance to the $[p]$ operator. The slight and quite natural adjustment of intuitionistic implication to our setting is then: $\mathfrak{A} \models_X \varphi \rightarrow_p \psi$ if and only if for all subsets $X' \subseteq X$ with $|X'| \geq p \cdot |X|$ (and $p \in [0, 1] \cap \mathbb{Q}$) it holds that $\mathfrak{A} \models_{X'} \varphi$ implies that $\mathfrak{A} \models_{X'} \psi$. The operator $[p]$ can now be expressed with the help of the intuitionistic *approximate* implication. One can easily verify that $[p]\varphi$ is equivalent to $\top \rightarrow_p \varphi$.

In this article we have considered approximation in the context of multiteam semantics when restricted to the finite. However our definitions can be generalised in a straightforward manner to deal with arbitrary cardinalities.

Acknowledgments The third author was supported by grants 292767, 308099, and 308712 of the Academy of Finland and 57348395 (DAAD). The fourth author was supported by the DFG grant ME 4279/1-1. The last author was supported by the Foundations' Post Doc Pool via Jenny and Antti Wihuri Foundation.

References

1. Böttcher, S., Link, S., Zhang, L.: Pulling conjunctive query equivalence out of the bag. In: Proceedings 23rd ACM CIKM, pp. 41–50. ACM (2014)
2. Chaudhuri, S., Vardi, M.Y.: Optimization of real conjunctive queries. In: Proceedings of the Twelfth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, PODS '93, pp. 59–70, New York, NY, USA (1993). ACM

3. Durand, A., Kontinen, J., de Rugy-Altherre, N., Väänänen, J.: Tractability frontier of data complexity in team semantics. Proc. 6th GandALF, EPTCS **193**, 73–85 (2015)
4. Durand, A., Kontinen, J., Vollmer, H.: Expressivity and complexity of dependence logic. In: Abramsky, S., Kontinen, J., Väänänen, J., Vollmer, H. (eds.) Dependence Logic: Theory and Applications, pp. 5–32. Springer International Publishing (2016)
5. Ebbing, J., Hella, L., Meier, A., Müller, J.-S., Virtema, J., Vollmer, H.: Extended modal dependence logic. In: WoLLIC, volume 8071 of Lecture Notes in Computer Science, pp. 126–137. Springer Berlin Heidelberg (2013)
6. Galliani, P.: Probabilistic dependence logic. Manuscript (2008)
7. Galliani, P.: Inclusion and exclusion dependencies in team semantics - on some logics of imperfect information. Ann. Pure Appl. Logic **163**(1), 68–84 (2012)
8. Galliani, P., Hannula, M., Kontinen, J.: Hierarchies in independence logic. In: Ronchi, S., Rocca, D. (eds.) Computer Science Logic 2013 (CSL 2013), CSL 2013, September 2-5, 2013, Torino, Italy, volume 23 of LIPIcs, pp. 263–280 (2013)
9. Galliani, P., Hella, L.: Inclusion logic and fixed point logic. In: Proc. CSL, pp. 281–295 (2013)
10. Galliani, P., Mann, A.L.: Lottery semantics: A compositional semantics for probabilistic first-order logic with imperfect information. Studia Logica **101**(2), 293–322 (2013)
11. Garey, M.R., Johnson, D.S., Stockmeyer, L.J.: Some simplified np-complete graph problems. Theor. Comput. Sci. **1**, 237–267 (1976)
12. Grädel, E., Väänänen, J.A.: Dependence and independence. Studia Logica **101**(2), 399–410 (2013)
13. Gyssens, M., Niepert, M., Gucht, D.V.: On the completeness of the semigraphoid axioms for deriving arbitrary from saturated conditional independence statements. Inf. Process. Lett. **114**(11), 628–633 (2014)
14. Hannula, M., Kontinen, J.: Hierarchies in independence and inclusion logic with strict semantics. J. Log. Comput. **25**(3), 879–897 (2015)
15. Hannula, M., Kontinen, J.: A finite axiomatization of conditional independence and inclusion dependencies. Inf. Comput. **249**, 121–137 (2016)
16. Hannula, M., Kontinen, J., Link, S.: On the finite and general implication problems of independence atoms and keys. J. Comput. Syst. Sci. **82**(5), 856–877 (2016)
17. Hannula, M., Kontinen, J., Virtema, J., Vollmer, H.: Complexity of propositional independence and inclusion logic. In: Proceedings 40th MFCS, volume 9234 of LNCS, pp. 269–280. Springer (2015)
18. Hella, L., Kuusisto, A., Meier, A., Virtema, J.: Model checking and validity in propositional and modal inclusion logics. In: 42nd International Symposium on Mathematical Foundations of Computer Science (MFCS 2017), volume 83 of Leibniz International Proceedings in Informatics (LIPIcs), Dagstuhl, Germany (2017). Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik
19. Hintikka, J., Sandu, G.: Informational independence as a semantical phenomenon. In: Logic, Methodology and Philosophy of Science, volume 8, pp. 571–589. Elsevier, Amsterdam (1989)
20. Hodges, W.: Compositional semantics for a language of imperfect information. Log. J. IGPL **5**(4), 539–563 (1997). electronic
21. Hyttinen, T., Paolini, G., Väänänen, J.: Quantum team logic and bell's inequalities. Rew. Symb. Logic **8**(4), 722–742 (2015)
22. Hyttinen, T., Paolini, G., Väänänen, J.: A logic for arguing about probabilities in measure teams. Arch. Math. Log. **56**(5), 475–489 (2017)
23. Kivinen, J., Mannila, H.: Approximate inference of functional dependencies from relations. Theor. Comput. Sci. **149**(1), 129–149 (1995)
24. Kolaitis, P.G.: The query containment problem: Set semantics vs. bag semantics. In: Proceedings 7th AMW, volume 1087 of CEUR Workshop Proceedings (2013)
25. Kontinen, J.: Coherence and computational complexity of quantifier-free dependence logic formulas. Studia Logica **101**(2), 267–291 (2013)
26. Kontinen, J., Kuusisto, A., Lohmann, P., Virtema, J.: Complexity of two-variable dependence logic and if-logic. Inf. Comput. **239**, 237–253 (2014)
27. Kontinen, J., Kuusisto, A., Virtema, J.: Decidability of Predicate Logics with Team Semantics. In: Faliszewski, P., Muscholl, A., Niedermeier, R. (eds.) 41st International Symposium on Mathematical Foundations of Computer Science (MFCS 2016), volume 58 of Leibniz International Proceedings in Informatics (LIPIcs), pp. 60:1–60:14, Dagstuhl, Germany (2016). Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik
28. Kontinen, J., Link, S., Väänänen, J.A.: Independence in database relations. In: Proceedings 20th WoLLIC, volume 8071 of LNCS, pp. 179–193. Springer (2013)
29. Kontinen, J., Müller, J.-S., Schnoor, H., Vollmer, H.: A Van Benthem Theorem for Modal Team Semantics. In: Proceedings 24th CSL, volume 41 of LIPIcs, pp. 277–291, Dagstuhl, Germany (2015). Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik

30. Krebs, A., Meier, A., Virtema, J.: A team based variant of CTL. In: Proceedings TIME, p. 2015 (2015)
31. Lamperti, G., Melchiori, M., Zanella, M.: On multisets in database systems. In: Proceedings WMP, pp. 147–216, London, UK, UK. Springer-Verlag, Berlin (2001)
32. Liu, L., Özsu, T.M. (eds.): Encyclopedia of Database Systems. Springer, US (2009)
33. Papadimitriou, C.H.: Computational complexity. Addison-Wesley, Boston (1994)
34. Vääänänen, J.: Dependence Logic - A New Approach to Independence Friendly Logic, volume 70 of London Mathematical Society student texts. Cambridge University Press, Cambridge (2007)
35. Väänänen, J.: Modal dependence logic. In: van Rooij, R., Apt, K. (eds.) New Perspectives on Games and Interaction, volume 5 of Texts in Logic and Games, pp. 237–254. Amsterdam University Press (2008)
36. Väänänen, J.: The logic of approximate dependence. In: Başkent, C., Moss, L.S., Ramanujam, R. (eds.) Rohit Parikh on Logic, Language and Society, pp. 227–234. Springer International Publishing, Cham (2017)
37. Michael Wong, S.K.: An extended relational data model for probabilistic reasoning. J. Intell. Inf. Syst. **9**(2), 181–202 (1997)
38. Wong, S.K.M., Butz, C.J., Wu, D.: On the implication problem for probabilistic conditional independency. IEEE Trans. Syst. Man Cybern. Syst. Hum. **30**(6), 785–805 (2000)
39. Yang, F.: On Extensions and Variants of Dependence Logic – A study of intuitionistic connectives in the team semantics setting. PhD thesis, Department of Mathematics and Statistics. University of Helsinki, Helsinki (2014)
40. Yang, F., Vääänänen, J.: Propositional logics of dependence. Ann. Pure Appl. Logic **167**(7), 557–589 (2016)