

Upwards Closed Dependencies in Team Semantics

Pietro Galliani*

Department of Mathematics and Statistics
Helsinki, Finland
pgallian@gmail.com

We prove that adding upwards closed first-order dependency atoms to first-order logic with team semantics does not increase its expressive power (with respect to sentences), and that the same remains true if we also add constancy atoms. As a consequence, the negations of functional dependence, conditional independence, inclusion and exclusion atoms can all be added to first-order logic without increasing its expressive power.

Furthermore, we define a class of bounded upwards closed dependencies and we prove that unbounded dependencies cannot be defined in terms of bounded ones.

1 Introduction

Team semantics is a generalization of Tarski’s semantics in which formulas are satisfied or not satisfied by sets of assignments, called *teams*, rather than by single assignments. It was originally developed by Hodges, in [14], as a compositional alternative to the imperfect-information *game theoretic semantics* for independence friendly logic [13, 18].

Over the past few years team semantics has been used to specify and study many other extensions of first-order logic. In particular, since a team describes a relation between the elements of its model team semantics offers a natural way to add to first-order logic atoms corresponding to database-theoretic *dependency notions*.

This line of thought led first to the development of *dependence logic* [19], and later to that of *independence logic* [12] and *inclusion and exclusion logics* [8].¹ By now there are many results in the literature concerning the properties of these logics, and in Section 2 we recall some of the principal ones.

One common characteristic of all these logics is that they are much stronger than first-order logic proper, even though they merely add *first-order definable* dependency conditions to its language. Indeed, the rules of team semantics straddle the line between first and second order, since they evaluate first-order connectives by means of second-order machinery: and, while in the case of first-order logic formulas team semantics can be reduced to Tarski’s semantics, if we add to our language atoms corresponding to further conditions the second-order nature of team semantics can take over.

The purpose of the present paper is to investigate the boundary between first and second order “from below”, so to say, taking first-order logic with team semantics and trying to find out how much we can add to it while preserving first-orderness. In Section 3 we define a fairly general family of classes of first-order definable dependency conditions and prove they can be safely added to first-order logic; then in Section 4 we expand this family, and in Section 5 we show that, as a consequence, the negations of all the main dependency atoms studied in team semantics do not “blow up” first-order logic into a higher

*Research supported by Grant 264917 of the Academy of Finland.

¹The literature contains many other extensions of first-order logic with team semantics, but we do not examine them in this work.

order one. Finally, in Section 6 we introduce a notion of *boundedness* for dependencies and use it to demonstrate some *non-definability results*.

2 Preliminaries

In this section we will recall some fundamental definitions and results concerning team semantics.

Definition 1 (Team) Let M be a first-order model and let $\text{Dom}(M)$ be the set of its elements.² Furthermore, let V be a finite set of variables. Then a team X over M with domain $\text{Dom}(X) = V$ is a set of assignments s from V to $\text{Dom}(M)$.

Given a team X and a tuple of variables \vec{v} contained in the domain of X , we write $X \upharpoonright \vec{v}$ for the team obtained by restricting all assignments of X to the variables of \vec{v} and $X(\vec{v})$ for the relation $\{s(\vec{v}) : s \in X\} \subseteq \text{Dom}(M)^{|\vec{v}|}$.

As it is common when working with team semantics, we will assume that all our expressions are in negation normal form.

Definition 2 (Team Semantics for First-Order Logic) Let $\phi(\vec{x})$ be a first-order formula in negation normal form with free variables in \vec{x} . Furthermore, let M be a first-order model whose signature contains the signature of ϕ and let X be a team over it whose domain contains \vec{x} . Then we say that X satisfies ϕ in M , and we write $M \models_X \phi$, if and only if this follows from these rules:³

TS-lit: For all first-order literals α , $M \models_X \alpha$ if and only if for all $s \in X$, $M \models_s \alpha$ according to the usual Tarski semantics;

TS- \vee : For all ψ and θ , $M \models_X \psi \vee \theta$ if and only if $X = Y \cup Z$ for two subteams Y and Z such that $M \models_Y \psi$ and $M \models_Z \theta$;

TS- \wedge : For all ψ and θ , $M \models_X \psi \wedge \theta$ if and only if $M \models_X \psi$ and $M \models_X \theta$;

TS- \exists : For all ψ and all variables v , $M \models_X \exists v \psi$ if and only if there exists a function

$$H : X \rightarrow \mathcal{P}(\text{Dom}(M)) \setminus \{\emptyset\}$$

such that $M \models_{X[H/v]} \psi$, where $X[H/v] = \{s[m/v] : s \in X, m \in H(s)\}$ and $\mathcal{P}(\text{Dom}(M))$ is the powerset of $\text{Dom}(M)$;

TS- \forall : For all ψ and all variables v , $M \models_X \forall v \psi$ if and only if $M \models_{X[M/v]} \psi$, where $X[M/v] = \{s[m/v] : s \in X, m \in M\}$.

Given a sentence (that is, a formula with no free variables) ϕ and a model M over its signature, we say that ϕ is true in M and we write $M \models \phi$ if and only if $M \models_{\{\emptyset\}} \phi$.⁴

The following is a useful and easily derived rule:

Lemma 3 Let $\vec{v} = v_1 \dots v_n$ be a tuple of n variables and let $\exists \vec{v} \psi$ be a shorthand for $\exists v_1 \dots \exists v_n \psi$. Then $M \models_X \exists \vec{v} \psi$ if and only if there exists a function $H : X \rightarrow \mathcal{P}(\text{Dom}(M)^n) \setminus \{\emptyset\}$ such that $M \models_{X[H/\vec{v}]} \psi$, where $X[H/\vec{v}] = \{s[\vec{m}/\vec{v}] : s \in X, \vec{m} \in H(s)\}$.

²We always assume that models have at least two elements in their domain.

³What we give here is the so-called *lax* version of team semantics. There also exists a *strict* version, with slightly different rules for disjunction and existential quantification; but as pointed out in [8], *locality* – in the sense of Theorem 8 here – fails in strict team semantics for some of the logics we are interested in. Therefore, in this work we will only deal with lax team semantics.

⁴Of course, one should not confuse the team $\{\emptyset\}$, which contains only the empty assignment, with the *empty team* \emptyset , which contains no assignments at all.

With respect to first-order formulas, team semantics can be reduced to Tarski's semantics. Indeed,

Proposition 4 ([14, 19]) *Let $\phi(\vec{x})$ be a first-order formula in negation normal form with free variables in \vec{x} . Furthermore, let M be a first-order model whose signature contains that of ϕ , and let X be a team over M whose domain contains \vec{x} . Then $M \models_X \phi$ if and only if, for all $s \in X$, $M \models_s \phi$ with respect to Tarski's semantics.*

In particular, a first-order sentence ϕ is true in a model M with respect to team semantics if and only if it is true in M with respect to Tarski's semantics.

Therefore, not all first-order definable properties of relations correspond to the satisfaction conditions of first-order formulas: for example, the non-emptiness of a relation R is definable by $\exists \vec{x} R\vec{x}$, but there is no first order ϕ such that $M \models_X \phi$ if and only if $X \neq \emptyset$. More in general, let $\phi^*(R)$ be a first-order sentence specifying a property of the k -ary relation R and let $\vec{x} = x_1 \dots x_k$ be a tuple of new variables: then, as it follows easily from the above proposition, there exists a first-order formula $\phi(\vec{x})$ such that

$$M \models_X \phi(\vec{x}) \Leftrightarrow M, X(\vec{x}) \models \phi^*(R)$$

if and only if $\phi^*(R)$ can be put in the form $\forall \vec{x}(R\vec{x} \rightarrow \theta(\vec{x}))$ for some θ in which R does not occur.⁵

It is hence possible to extend first-order logic (with team semantics) by introducing new atoms corresponding to further properties of relations. Database theory is a most natural choice as a source for such properties; and, in the rest of this section, we will recall the fundamental database-theoretic extensions of first-order logic with team semantics and some of their properties.

Dependence logic $\text{FO}(=(\cdot, \cdot))$, from [19], adds to first-order logic *functional dependence atoms* $=(\vec{x}, \vec{y})$ based on database-theoretic *functional dependencies* ([2]). Their rule in team semantics is

TS-fdep: $M \models_X=(\vec{x}, \vec{y})$ if and only if for all $s, s' \in X$, $s(\vec{x}) = s'(\vec{x}) \Rightarrow s(\vec{y}) = s'(\vec{y})$.

This atom, and dependence logic as a whole, is *downwards closed*: for all dependence logic formulas ϕ , models M and teams X , if $M \models_X \phi$ then $M \models_Y \phi$ for all $Y \subseteq X$. It is not however *union closed*: if $M \models_X \phi$ and $M \models_Y \phi$ then we cannot in general conclude that $M \models_{X \cup Y} \phi$.

Dependence logic is equivalent to existential second-order logic over sentences:

Theorem 5 ([19]) *Every dependence logic sentence ϕ is logically equivalent to some ESO sentence ϕ^* , and vice versa.*

Constancy logic $\text{FO}(=(\cdot))$ is the fragment of dependence logic which only allows functional dependence atoms of the form $=(\emptyset, \vec{x})$, which we will abbreviate as $=(\vec{x})$ and call *constancy atoms*. Clearly we have that

TS-const: $M \models_X=(\vec{x})$ if and only if for all $s, s' \in X$, $s(\vec{x}) = s'(\vec{x})$.

As proved in [8], every constancy logic sentence is equivalent to some first-order sentence: therefore, constancy logic is strictly weaker than dependence logic. Nonetheless, constancy logic is more expressive than first-order logic with respect to the second-order relations generated by the satisfaction conditions of formulas: indeed, it is an easy consequence of Proposition 4 that no first-order formula is logically equivalent to the constancy atom $=(x)$.

Exclusion logic $\text{FO}(|)$, from [8], adds to first-order logic *exclusion atoms* $\vec{x} | \vec{y}$, where \vec{x} and \vec{y} are tuples of variables of the same length. Just as functional dependence atoms correspond to functional database-theoretic dependencies, exclusion atoms correspond to *exclusion dependencies* [3]; and their satisfaction rule is

⁵That is, according to the terminology of [19], if and only if $\phi^*(R)$ is *flat*.

TS-excl: $M \models_X \vec{x} \mid \vec{y}$ if and only if $X(\vec{x}) \cap X(\vec{y}) = \emptyset$.

As proved in [8], exclusion logic is entirely equivalent to dependence logic: every exclusion logic formula is logically equivalent to some dependence logic formula, and vice versa.

Inclusion logic $\text{FO}(\subseteq)$, also from [8], adds instead to first-order logic *inclusion atoms* $\vec{x} \subseteq \vec{y}$ based on database-theoretic *inclusion dependencies* [6]. The corresponding rule is

TS-inc: $M \models_X \vec{x} \subseteq \vec{y}$ if and only if $X(\vec{x}) \subseteq X(\vec{y})$.

Inclusion logic is stronger than first-order logic, but weaker than existential second-order logic: indeed, as shown in [9], sentence-wise it is equivalent to positive greatest fixed point logic GFP^+ . Formula-wise, it is incomparable with constancy, dependence or exclusion logic, since its formulas are union closed but not downwards closed.

Independence logic $\text{FO}(\perp)$, from [12], adds to first-order logic *independence atoms* $\vec{x} \perp \vec{y}$ with the intended meaning of “the values of \vec{x} and \vec{y} are informationally independent”. More formally,

TS-ind: $M \models_X \vec{x} \perp \vec{y}$ if and only if $X(\vec{x}\vec{y}) = X(\vec{x}) \times X(\vec{y})$.

This notion of informational independence has a long history: see for example [11] for an analysis of this concept from a probabilistic perspective.

The *conditional* independence atoms $\vec{x} \perp_{\vec{z}} \vec{y}$, also from [12], relativize the independence of \vec{x} and \vec{y} to all *fixed* value of \vec{z} . Their semantics is

TS-c-ind: $M \models_X \vec{x} \perp_{\vec{z}} \vec{y}$ if and only if for all tuples $\vec{m} \in \text{Dom}(M)^{|\vec{z}|}$ and for $X_{\vec{z}=\vec{m}} = \{s \in X : s(\vec{z}) = \vec{m}\}$ it holds that $X_{\vec{z}=\vec{m}}(\vec{x}\vec{y}) = X_{\vec{z}=\vec{m}}(\vec{x}) \times X_{\vec{z}=\vec{m}}(\vec{y})$.

As pointed out in [4], the rule for $\vec{x} \perp_{\vec{z}} \vec{y}$ corresponds precisely to the database-theoretic *embedded multivalued dependency* [5] ($\vec{z} \twoheadrightarrow \vec{x} \mid \vec{y}$).

In [12] it was shown that every dependence logic formula is equivalent to some $\text{FO}(\perp_c)$ (conditional independence logic) formula, but not vice versa; and sentence-wise, both of these logics are equivalent to each other (and to ESO). Furthermore, in [8] it was proved that $\text{FO}(\perp_c)$ is equivalent to **inclusion/exclusion logic**⁶ $\text{FO}(\subseteq, \mid)$, even with respect to open formulas, and that this is, roughly speaking, the most general logic obtainable by adding first-order (or even existential second-order) definable dependency conditions to first-order logic.⁷ More recently, in [10], it was shown that $\text{FO}(\perp)$ and $\text{FO}(\perp_c)$ are also equivalent.

We conclude this section with Figure 1, which depicts the relations between the logics we discussed so far.

3 Upwards Closed Dependencies

In this work we will study the properties of the logics obtained by adding families of *dependency conditions* to the language of first-order logic. But what is a dependency condition, in a general sense? The following definition is based on the *generalized atoms* of [17]:

Definition 6 *Let $n \in \mathbb{N}$. A dependency of arity n is a class \mathbf{D} , closed under isomorphisms, of models over the signature $\{R\}$ where R is a n -ary relation symbol. If \vec{x} is a tuple of n variables (possibly with repetitions), M is a first-order model and X is a team over it whose domain contains all variables of \vec{x} then*

⁶That is, to first-order logic plus inclusion *and* exclusion atoms.

⁷To be more precise, for every ESO formula $\phi^*(R)$ there exists a $\text{FO}(\perp_c)$ formula $\phi(\vec{x})$ such that, for all suitable models M and nonempty teams X , $M \models_X \phi(\vec{x})$ if and only if $M, X(\vec{x}) \models \phi^*(R)$.

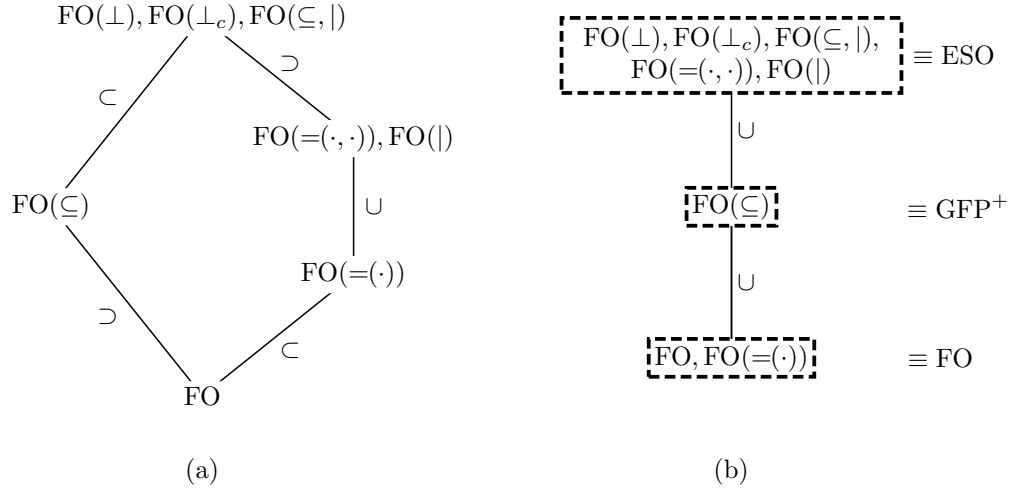


Figure 1: Relations between logics wrt formulas (a) and sentences (b).

TS-D: $M \models_X \mathbf{D}\vec{x}$ if and only if $(\text{Dom}(M), X(\vec{x})) \in \mathbf{D}$.

Definition 7 Let $\mathcal{D} = \{\mathbf{D}_1, \mathbf{D}_2, \dots\}$ be a family of dependencies. Then we write $FO(\mathcal{D})$ for the logic obtained by adding to the language of first-order logic all dependency atoms $\mathbf{D}\vec{x}$, where $\mathbf{D} \in \mathcal{D}$ and \vec{x} is a tuple of variables of the arity of \mathbf{D} .

It is not difficult to represent the logics of Section 2 in this notation. For example, dependence logic is $FO(=(\cdot, \cdot))$ for $=(\cdot, \cdot) = \{=(n, m) : n, m \in \mathbb{N}\}$, where $(\text{Dom}(M), R) \in =(n, m)$ if and only if

$$\vec{a}\vec{b}, \vec{a}\vec{c} \in R \Rightarrow \vec{b} = \vec{c}$$

for all tuples of elements $\vec{a} = a_1 \dots a_n, \vec{b} = b_1 \dots b_m, \vec{c} = c_1 \dots c_m \in \text{Dom}(M)$.

The following property can be easily verified, by induction on the formulas ϕ :⁸

Theorem 8 (Locality) Let \mathcal{D} be a family of dependencies and let $\phi(\vec{x})$ be a formula of $FO(\mathcal{D})$ with free variables in \vec{x} . Then for all models M and all teams X over it whose domain contains \vec{x} , $M \models_X \phi(\vec{x})$ if and only if $M \models_{X \upharpoonright \vec{x}} \phi(\vec{x})$.

In this work, we will be mainly interested in dependencies which correspond to first-order definable properties of relations:

Definition 9 A dependency notion \mathbf{D} is first-order definable if there exists a first-order sentence $\mathbf{D}^*(R)$ over the signature $\{R\}$, where R is a new relation symbol, such that

$$M \in \mathbf{D} \Leftrightarrow M \models \mathbf{D}^*(R)$$

for all models $M = (\text{Dom}(M), R)$.

⁸For the sake of reference, we mention Theorem 4.22 of [8] in which the same result is proved in detail for (conditional) independence logic. The only new case here is the one in which $\phi(\vec{x}) = \mathbf{D}\vec{y}$ for some $\mathbf{D} \in \mathcal{D}$ and \vec{y} is contained in \vec{x} ; and for it the result follows at once from condition **TS-D** and from the fact that $X(\vec{y}) = (X \upharpoonright \vec{x})(\vec{y})$.

It is not necessarily the case that if \mathbf{D} is first-order definable then $\text{FO}(\mathbf{D})$ and FO are equivalent with respect to sentences. For example $=(n, m)^*(R)$ is $\forall \vec{x}\vec{y}\vec{z}(R\vec{x}\vec{y} \wedge R\vec{x}\vec{z} \rightarrow \vec{y} = \vec{z})$, where \vec{x} has length n and \vec{y}, \vec{z} have length m ; but as we said in Section 2, dependence logic is stronger than first-order logic.

When is then the case that dependency conditions can be added safely to first-order logic, without increasing the expressive power? The following definition will provide us a partial answer:

Definition 10 A dependency notion \mathbf{D} is upwards closed if

$$(\text{Dom}(M), R) \in \mathbf{D}, R \subseteq S \Rightarrow (\text{Dom}(M), S) \in \mathbf{D}$$

for all models $(\text{Dom}(M), R)$ and all relations S over $\text{Dom}(M)$ of the same arity of R .

It is easy to see that upwards closed dependencies induce upwards closed satisfaction rules: if \mathbf{D} is upwards closed, $M \models_X \mathbf{D}\vec{x}$ and $X \subseteq Y$ then it is always the case that $M \models_Y \mathbf{D}\vec{x}$. However, differently from the case of downwards or union closure, upwards closure is not preserved by team semantics: if \mathbf{D} is upwards closed, $\phi \in \text{FO}(\mathbf{D})$ and $M \models_X \phi$ then it is not in general true that $M \models_Y \phi$ for all $Y \supseteq X$ (for example, let ϕ be a nontrivial first-order literal and recall Rule **TS-lit**).

Some examples of upwards closed dependencies follow:

Non-emptiness: $M \models_X \text{NE}$ if and only if $X \neq \emptyset$;

Intersection: $M \models_X \diamond(\vec{x} = \vec{y})$ if and only if there exists a $s \in X$ with $s(\vec{x}) = s(\vec{y})$;

Inconstancy: $M \models_X \neq(\vec{x})$ if and only if $|X(\vec{x})| > 1$;

n -bigness: For all $n \in \mathbb{N}$, $M \models_X |\vec{x}| \geq n$ if and only if $|X(\vec{x})| \geq n$;

Totality: $M \models_X \text{All}(\vec{x})$ if and only if $X(\vec{x}) = \text{Dom}(M)^{|\vec{x}|}$;

Non-dependence: $M \models_X \neq(\vec{x}, \vec{y})$ if and only if there exist $s, s' \in X$ with $s(\vec{x}) = s'(\vec{x})$ but $s(\vec{y}) \neq s'(\vec{y})$;⁹

Non-exclusion: $M \models_X \vec{x} \uparrow \vec{y}$ if and only if there exist $s, s' \in X$ with $s(\vec{x}) = s'(\vec{y})$;

Infinity: $M \models_X |\vec{x}| \geq \omega$ if and only if $X(\vec{x})$ is infinite;

κ -bigness: For all cardinals κ , $M \models_X |\vec{x}| \geq \kappa$ if and only if $|X(\vec{x})| \geq \kappa$.

All the above examples except infinity and κ -bigness are first-order definable. The NE atom is the adaptation to first-order team semantics of the non-emptiness atom introduced in [20] for the propositional version of dependence logic, and the totality atom All is due to Abramsky and Väänänen ([1]).

The main result of this section is the following:

Theorem 11 Let \mathcal{D} be a collection of upwards closed first-order definable dependency conditions. Then for every formula $\phi(\vec{x})$ of $\text{FO}(\mathcal{D})$ with free variables in \vec{x} there exists a first-order sentence $\phi^*(R)$, where R is a new $|\vec{x}|$ -ary relation symbol, such that

$$M \models_X \phi(\vec{x}) \Leftrightarrow M, X(\vec{x}) \models \phi^*(R)$$

for all models M over the signature of ϕ and all teams X .

In particular, every sentence of $\text{FO}(\mathcal{D})$ is equivalent to some first-order sentence.

Let us begin by adapting the notion of *flattening* of [19] to the case of an arbitrary logic $\text{FO}(\mathcal{D})$:

⁹The same symbol $\neq(\vec{x}, \vec{y})$ has been used in [7] to describe a different non-dependence notion, stating that for every $s \in X$ there exists a $s' \in X$ with $s(\vec{x}) = s'(\vec{x}), s(\vec{y}) \neq s'(\vec{y})$. In that thesis it was proved that the resulting “non-dependence logic” is equivalent to inclusion logic. As we will see, this is not the case for the non-dependence notion of this paper.

Definition 12 Let \mathcal{D} be any set of dependency conditions and let ϕ be a $FO(\mathcal{D})$ formula. Then its flattening ϕ^f is the first-order formula obtained by replacing any non-first-order atom with \top , where \top is the trivially true atom.

It is trivial to see, by induction on ϕ , that

Lemma 13 For all \mathcal{D} , all $\phi \in FO(\mathcal{D})$, all models M and all teams X over M , if $M \models_X \phi$ then $M \models_X \phi^f$.

As we said, even if \mathcal{D} contains only upwards closed dependency conditions it is not true that all formulas of $FO(\mathcal{D})$ are upwards closed. However, the following restricted variant of upwards closure is preserved:

Theorem 14 Let ϕ be a $FO(\mathcal{D})$ formula, where \mathcal{D} contains only upwards closed dependencies. Let M be a first-order model, and let X, Y be teams such that $X \subseteq Y$, $M \models_X \phi$, and $M \models_Y \phi^f$. Then $M \models_Y \phi$.

Proof:

The proof is by structural induction on ϕ .

1. If ϕ is a first-order literal, $\phi^f = \phi$ and there is nothing to prove;
2. If ϕ is of the form $\mathbf{D}\vec{x}$ for some $\mathbf{D} \in \mathcal{D}$, $M \models_X \phi$ and $X \subseteq Y$, then by upwards closure $M \models_Y \phi$;
3. Suppose that $M \models_X \phi_1 \vee \phi_2$ and $M \models_Y \phi_1^f \vee \phi_2^f$. Now $X = X_1 \cup X_2$ for two X_1, X_2 such that $M \models_{X_1} \phi_1$ and $M \models_{X_2} \phi_2$, and therefore by Lemma 13 $M \models_{X_1} \phi_1^f$ and $M \models_{X_2} \phi_2^f$. Furthermore, $Y = Y_1 \cup Y_2$ for two Y_1, Y_2 such that $M \models_{Y_1} \phi_1^f$ and $M \models_{Y_2} \phi_2^f$. Let $Z_1 = X_1 \cup Y_1$ and $Z_2 = X_2 \cup Y_2$; then $Z_1 \cup Z_2 = X \cup Y = Y$, and by Proposition 4 $M \models_{Z_1} \phi_1^f$ and $M \models_{Z_2} \phi_2^f$. But $M \models_{X_1} \phi_1$ and $X_1 \subseteq Z_1$, so by induction hypothesis $M \models_{Z_1} \phi_1$; and similarly, $M \models_{X_2} \phi_2$ and $X_2 \subseteq Z_2$, so $M \models_{Z_2} \phi_2$. Therefore $M \models_Y \phi_1 \vee \phi_2$, as required.
4. If $M \models_X \phi_1 \wedge \phi_2$ then $M \models_X \phi_1$ and $M \models_X \phi_2$. Then by induction hypothesis, since $M \models_Y \phi_1^f$ and $X \subseteq Y$, $M \models_Y \phi_1$; and similarly, since $M \models_Y \phi_2^f$ and $X \subseteq Y$, $M \models_Y \phi_2$, and therefore $M \models_Y \phi_1 \wedge \phi_2$.
5. If $M \models_X \exists v \phi$ then there is a function $H : X \rightarrow \mathcal{P}(\text{Dom}(M)) \setminus \{\emptyset\}$ such that $M \models_{X[H/v]} \phi$, and therefore (by Lemma 13) such that $M \models_{X[H/v]} \phi^f$. Similarly, if $M \models_Y \exists v \phi^f$ then for some K we have that $M \models_{Y[K/v]} \phi^f$. Now let $W : Y \rightarrow \mathcal{P}(\text{Dom}(M)) \setminus \{\emptyset\}$ be such that

$$W(s) = \begin{cases} H(s) \cup K(s) & \text{if } s \in X; \\ K(s) & \text{if } s \in Y \setminus X. \end{cases}$$

Then $Y[W/v] = X[H/v] \cup Y[K/v]$, and therefore by Proposition 4 $M \models_{Y[W/v]} \phi^f$. Then by induction hypothesis $M \models_{Y[W/v]} \phi$, since $X[H/v]$ satisfies ϕ and is contained in $Y[W/v]$; and therefore $M \models_Y \exists v \phi$, as required.

6. If $M \models_X \forall v \phi$ then $M \models_{X[M/v]} \phi$, and if $M \models_Y \forall v \phi^f$ then $M \models_{Y[M/v]} \phi^f$. Now $X[M/v] \subseteq Y[M/v]$, so by induction hypothesis $M \models_{Y[M/v]} \phi$, and therefore $M \models_Y \forall v \phi$.

□

Definition 15 If θ is a first-order formula and ϕ is a $FO(\mathcal{D})$ formula we define $(\phi \uparrow \theta)$ as $(\neg\theta) \vee (\theta \wedge \phi)$, where $\neg\theta$ is a shorthand for the first-order formula in negation normal form which is equivalent to the negation of θ .

The following lemma is obvious:

Lemma 16 For all first order θ and $\phi \in FO(\mathcal{D})$, $M \models_X (\phi \upharpoonright \theta)$ if and only if $M \models_Y \phi$ for $Y = \{s \in X : M \models_s \theta\}$.

One can observe that $(\phi \upharpoonright \theta)$ is logically equivalent to $\theta \leftrightarrow \phi$, where \leftrightarrow is the *maximal implication* of [16]:

TS-maximp: $M \models_X \theta \leftrightarrow \phi$ if and only if for all maximal $Y \subseteq X$ s.t. $M \models_Y \theta$, $M \models_Y \phi$.

We use the notation $(\phi \upharpoonright \theta)$, instead of $\theta \leftrightarrow \phi$, to make it explicit that θ is first order and that Lemma 16 holds.

The next step of our proof of Theorem 11 is to identify a fragment of our language whose satisfaction conditions do not involve quantification over second-order objects such as teams or functions. We do so by limiting the availability of disjunction and existential quantification:

Definition 17 A $FO(\mathcal{D})$ formula ϕ is clean if

1. All its disjunctive subformulas $\psi_1 \vee \psi_2$ are first order or of the form $\psi \upharpoonright \theta$ for some suitable choice of ψ and θ (where θ is first order);
2. All its existential subformulas $\exists v \psi$ are first order.

As the next proposition shows, clean formulas correspond to first-order definable properties of relations.

Proposition 18 Let \mathcal{D} be a class of first-order definable dependencies and let $\phi(\vec{x}) \in FO(\mathcal{D})$ be a clean formula with free variables in \vec{x} . Then there exists some first-order sentence $\phi^*(R)$, where R is a new $|\vec{x}|$ -ary relation, such that

$$M \models_X \phi(\vec{x}) \Leftrightarrow M, X(\vec{x}) \models \phi^*(R). \quad (1)$$

Proof:

By induction over ϕ .

1. If $\phi(\vec{x})$ is a first-order formula (not necessarily just a literal) then let $\phi^*(R) = \forall \vec{x}(R\vec{x} \rightarrow \phi(\vec{x}))$. By Proposition 4, (1) holds.
2. If $\phi(\vec{x})$ is a dependency atom $\mathbf{D}\vec{y}$, where $\mathbf{D} \in \mathcal{D}$ and \vec{y} is a tuple (possibly with repetitions) of variables occurring in \vec{x} , let $\phi^*(R)$ be obtained from $\mathbf{D}^*(S)$ by replacing every instance $S\vec{z}$ of S in it with $\exists \vec{x}(\vec{z} = \vec{y} \wedge R\vec{x})$. Indeed, $M \models_X \mathbf{D}\vec{y}$ if and only if $M, X(\vec{y}) \models \mathbf{D}^*(S)$, and $\vec{m} \in X(\vec{y})$ if and only if $M, X(\vec{x}) \models \exists \vec{x}(\vec{m} = \vec{y} \wedge R\vec{x})$.
3. If $\phi(\vec{x})$ is of the form $(\psi(\vec{x}) \upharpoonright \theta(\vec{x}))$, let $\phi^*(R)$ be obtained from $\psi^*(R)$ by replacing every instance $R\vec{z}$ of R with $R\vec{z} \wedge \theta(\vec{z})$. Indeed, by Lemma 16 $M \models_X (\psi(\vec{x}) \upharpoonright \theta(\vec{x}))$ if and only if $M \models_Y \psi(\vec{x})$ for $Y = \{s \in X : M \models_s \theta\}$, and $\vec{m} \in Y(\vec{x}) \Leftrightarrow \vec{m} \in X(\vec{x})$ and $M \models \theta(\vec{m})$.
4. If $\phi(\vec{x})$ is of the form $\psi(\vec{x}) \wedge \theta(\vec{x})$ simply let $\phi^*(R) = \psi^*(R) \wedge \theta^*(R)$.
5. If $\phi(\vec{x})$ is of the form $\forall v \psi(\vec{x}, v)$, where we assume without loss of generality that v is distinct from all $x \in \vec{x}$, and $\psi^*(S)$ corresponds to $\psi(\vec{x}, v)$ then let $\phi^*(R)$ be obtained from $\psi^*(S)$ by replacing every $S\vec{z}w$ with $R\vec{z}$. Indeed, $M \models_X \forall v \psi$ if and only if $M \models_{X[M/v]} \psi(\vec{x}, v)$ and $\vec{m}m' \in X[M/v](\vec{x}v)$ if and only if $\vec{m} \in X(\vec{x})$.

□

All that is now left to prove is the following:

Proposition 19 Let \mathcal{D} be a family of upwards closed dependencies. Then every $FO(\mathcal{D})$ formula is equivalent to some clean $FO(\mathcal{D})$ formula.

Proof:

It suffices to observe the following facts:

- If $\phi_1(\vec{x})$ and $\phi_2(\vec{x})$ are in $\text{FO}(\mathcal{D})$ then $\phi_1(\vec{x}) \vee \phi_2(\vec{x})$ is logically equivalent to

$$(\phi_1^f \vee \phi_2^f) \wedge (\phi_1 \uparrow \phi_1^f) \wedge (\phi_2 \uparrow \phi_2^f).$$

Indeed, suppose that $M \models_X \phi_1 \vee \phi_2$: then, by Lemma 13, $M \models_X \phi_1^f \vee \phi_2^f$. Furthermore, $X = Y \cup Z$ for two Y and Z such that $M \models_Y \phi_1$ and $M \models_Z \phi_2$. Now let $Y' = \{s \in X : M \models_s \phi_1^f\}$ and $Z' = \{s \in X : M \models_s \phi_2^f\}$: by Lemma 13 and Proposition 4 we have that $Y \subseteq Y'$ and that $Z \subseteq Z'$, and therefore by Theorem 14 $M \models_{Y'} \phi_1$ and $M \models_{Z'} \phi_2$. Thus by Lemma 16 $M \models_X (\phi_1 \uparrow \phi_1^f)$ and $M \models_X (\phi_2 \uparrow \phi_2^f)$, as required.

Conversely, suppose that $M \models_X (\phi_1^f \vee \phi_2^f) \wedge (\phi_1 \uparrow \phi_1^f) \wedge (\phi_2 \uparrow \phi_2^f)$. Then let $Y = \{s \in X : M \models_s \phi_1^f\}$ and $Z = \{s \in X : M \models_s \phi_2^f\}$. By Proposition 4 and since $M \models_X \phi_1^f \vee \phi_2^f$, $X = Y \cup Z$; and by Lemma 16, $M \models_Y \phi_1$ and $M \models_Z \phi_2$. So $M \models_X \phi_1 \vee \phi_2$, as required.

- If $\phi(\vec{x}, v) \in \text{FO}(\mathcal{D})$ then $\exists v \phi(\vec{x}, v)$ is logically equivalent to

$$(\exists v \phi^f(\vec{x}, v)) \wedge \forall v (\phi(\vec{x}, v) \uparrow \phi^f(\vec{x}, v)).$$

Indeed, suppose that $M \models_X \exists v \phi(\vec{x}, v)$. Then by Lemma 13 $M \models_X \exists v \phi^f(\vec{x}, v)$. Furthermore, for some $H : X \rightarrow \mathcal{P}(\text{Dom}(M)) \setminus \{\emptyset\}$ and for $Y = X[H/v]$ it holds that $M \models_Y \phi(\vec{x}, v)$. Now let $Z = \{h \in X[M/v] : M \models_h \phi^f(\vec{x}, v)\}$. By Proposition 4, $M \models_Z \phi^f(\vec{x}, v)$; and since $Y \subseteq Z$, by Theorem 14 $M \models_Z \phi(\vec{x}, v)$, and therefore by Lemma 16 $M \models_{X[M/v]} (\phi(\vec{x}, v) \uparrow \phi^f(\vec{x}, v))$, as required.

Conversely, suppose that $M \models_X (\exists v \phi^f(\vec{x}, v)) \wedge \forall v (\phi(\vec{x}, v) \uparrow \phi^f(\vec{x}, v))$. Then, for all $s \in X$, let $K(s) = \{m \in \text{Dom}(M) : M \models_{s[m/v]} \phi^f(\vec{x}, v)\}$. Since $M \models_X \exists v \phi^f(\vec{x}, v)$, $K(s)$ is nonempty for all $s \in X$, and by construction $X[K/v] = \{s \in X[M/v] : M \models_s \phi^f(\vec{x}, v)\}$. Now $M \models_{X[M/v]} (\phi(\vec{x}, v) \uparrow \phi^f(\vec{x}, v))$, so by Lemma 16 $M \models_{X[K/v]} \phi(\vec{x}, v)$ and in conclusion $M \models_X \exists v \phi(\vec{x}, v)$.

Applying inductively these two results to all subformulas of some $\phi \in \text{FO}(\mathcal{D})$ we can obtain some clean ϕ' to which ϕ is equivalent, and this concludes the proof.

□

Finally, the proof of Theorem 11 follows at once from Propositions 18 and 19.

Since, as we saw, the negations of functional and exclusion dependencies are upwards closed, we obtain at once the following corollary:

Corollary 20 *Any sentence of $\text{FO}(\neq, \cdot, \cdot, \uparrow)$ (that is, of first-order logic plus negated functional and exclusion dependencies) is equivalent to some first-order sentence.*

4 Adding Constancy Atoms

As we saw in the previous section, upwards closed dependencies can be added to first-order logic without increasing its expressive power (with respect to sentences); and as mentioned in Section 2, this is also true for the (non upwards-closed) constancy dependencies $=(\vec{x})$.

But what if our logic contains both upwards closed *and* constancy dependencies? As we will now see, the conclusion of Theorem 11 remains valid:

Theorem 21 *Let \mathcal{D} be a collection of upwards closed first-order definable dependency conditions. Then for every formula $\phi(\vec{x})$ of¹⁰ $FO(=(\cdot), \mathcal{D})$ with free variables in \vec{x} there exists a first-order sentence $\phi^*(R)$, where R is a new $|\vec{x}|$ -ary relation symbol, such that*

$$M \models_X \phi(\vec{x}) \Leftrightarrow M, X(\vec{x}) \models \phi^*(R).$$

In particular, every sentence of $FO(\mathcal{D})$ is equivalent to some first-order sentence.

The main ingredient of our proof will be the following lemma.

Lemma 22 *Let \mathcal{D} be any family of dependencies and let $\phi(\vec{x})$ be a $FO(=(\cdot), \mathcal{D})$ formula. Then $\phi(\vec{x})$ is equivalent to some formula of the form $\exists \vec{v}(=(\vec{v}) \wedge \psi(\vec{x}, \vec{v}))$, where $\psi \in FO(\mathcal{D})$ contains exactly the same instances of **D**-atoms (for all **D** $\in \mathcal{D}$) that ϕ does, and in the same number.*

The proof of this lemma is by induction on ϕ , and it is entirely analogous to the corresponding proof from [8].

Now we can prove Theorem 21.

Proof:

Let $\phi(\vec{x})$ be a $FO(=(\cdot), \mathcal{D})$ -formula. Then by Lemma 22 $\phi(\vec{x})$ is equivalent to some sentence of the form $\exists \vec{v}(=(\vec{v}) \wedge \psi(\vec{x}, \vec{v}))$, where $\psi(\vec{x}, \vec{v}) \in FO(\mathcal{D})$. But then by Theorem 11 there exists a first-order formula $\psi^*(S)$ such that $M \models_X \psi(\vec{x}, \vec{v})$ if and only if $M, X(\vec{x}\vec{v}) \models \psi^*(S)$. Now let $\theta(R, \vec{v})$ be obtained from $\psi^*(S)$ by replacing any $S\vec{y}\vec{z}$ with $R\vec{y} \wedge \vec{z} = \vec{v}$. Since $X[\vec{m}/\vec{v}](\vec{x}\vec{v}) = \{\vec{a}\vec{m} : \vec{a} \in X(\vec{x})\}$ it is easy to see that $M \models_X \exists \vec{v}(=(\vec{v}) \wedge \psi(\vec{x}, \vec{v}))$ if and only if $M, X(\vec{x}) \models \exists \vec{v}\theta(R, \vec{v})$, and this concludes the proof. \square

5 Possibility, Negated Inclusion and Negated Conditional Independence

By Corollary 20, the negations of exclusion and functional dependence atoms can be added to first-order logic without increasing its power. But what about the negations of inclusion and (conditional) independence? These are of course first-order definable, but they are not upwards closed: indeed, their semantic rules can be given as

TS- $\not\subseteq$: $M \models_X \vec{x} \not\subseteq \vec{y}$ if and only if there is a $s \in X$ such that for all $s' \in X$, $s(\vec{x}) \neq s'(\vec{y})$;

TS- $\not\subseteq_c$: $M \models_X \vec{x} \not\subseteq_c \vec{y}$ if and only if there are $s, s' \in X$ with $s(\vec{z}) = s'(\vec{z})$ and such that for all $s'' \in X$, $s''(\vec{x}\vec{z}) \neq s(\vec{x}\vec{z})$ or $s''(\vec{y}\vec{z}) \neq s(\vec{y}\vec{z})$.

However, we will now prove that, nonetheless, $FO(\neq(\cdot, \cdot), \not\subseteq, \not\subseteq_c)$ is equivalent to FO on the level of sentences. In order to do so, let us first define the following *possibility operator* and prove that it is uniformly definable in $FO(=(\cdot), \neq(\cdot))$:

Definition 23 *Let ϕ be any $FO(\mathcal{D})$ formula, for any choice of \mathcal{D} . Then*

TS- \diamond : $M \models_X \diamond\phi$ if there exists a $Y \subseteq X$, $Y \neq \emptyset$, such that $M \models_Y \phi$.

Lemma 24 *Let ϕ be any $FO(\mathcal{D})$ formula, for any \mathcal{D} . Then $\diamond\phi$ is logically equivalent to*

$$\exists u_0 u_1 \exists v(=(u_0) \wedge =(u_1) \wedge (v = u_0 \vee v = u_1) \wedge (\phi \upharpoonright v = u_1) \wedge \neq(v)). \quad (2)$$

¹⁰Here $=(\cdot)$ represents the class of all constancy dependencies of all arities. But it is easy to see that the one of arity 1 would suffice: indeed, if \vec{x} is $x_1 \dots x_n$ then $=(\vec{x})$ is logically equivalent to $=(x_1) \wedge \dots \wedge =(x_n)$.

Proof:

Suppose that there is a $Y \subseteq X$, $Y \neq \emptyset$, such that $M \models_Y \phi$. Then let $0, 1 \in \text{Dom}(M)$ be such that $0 \neq 1$, let $H : X[01/u_0u_1] \rightarrow \mathcal{P}(\text{Dom}(M)) \setminus \{\emptyset\}$ be such that

$$H(s[01/u_0u_1]) = \begin{cases} \{0, 1\} & \text{if } s \in Y; \\ \{0\} & \text{if } s \in X \setminus Y \end{cases}$$

and let $Z = X[01/u_0u_1][H/v]$. Clearly $M \models_Z (u_0) \wedge (u_1) \wedge (v = u_0 \vee v = u_1) \wedge (\phi \upharpoonright v = u_1)$, and it remains to show that $M \models_Z \neq(v)$. But by hypothesis Y is nonempty, and therefore there exists a $s \in Y \subseteq X$ such that $\{s[010/u_0u_1v], s[011/u_0u_1v]\} \subseteq Z$. So v is not constant in Z , as required, and X satisfies (2).

Conversely, suppose that X satisfies (2), let 0 and 1 be our choices for u_0 and u_1 , and let H be the choice function for v . Then let $Y = \{s \in X : 1 \in H(s[01/u_0u_1])\}$. By locality, Lemma 16 and the fact that $M \models_{X[01H/u_1u_2v]} (\phi \upharpoonright v = u_1)$ we have that $M \models_Y \phi$; and Y is nonempty, since $M \models_Z (v = u_0 \vee v = u_1) \wedge \neq(v)$.

□

It is now easy to see that the negations of inclusion and conditional independence are in $\text{FO}(=(\cdot), \neq(\cdot))$:

Proposition 25 For all \vec{x}, \vec{y} with $|\vec{x}| = |\vec{y}|$, $\vec{x} \not\subseteq \vec{y}$ is logically equivalent to

$$\exists \vec{z} (=(\vec{z}) \wedge \diamond(\vec{z} = \vec{x}) \wedge \vec{z} \neq \vec{y}).$$

Proposition 26 For all \vec{x}, \vec{y} and \vec{z} , $\vec{x} \not\perp_{\vec{z}} \vec{y}$ is logically equivalent to

$$\exists \vec{p}\vec{q}\vec{r} (=(\vec{p}\vec{q}\vec{r}) \wedge \diamond(\vec{p}\vec{r} = \vec{x}\vec{z}) \wedge \diamond(\vec{q}\vec{r} = \vec{y}\vec{z}) \wedge \vec{p}\vec{q}\vec{r} \neq \vec{x}\vec{y}\vec{z}).$$

Corollary 27 Every sentence of $\text{FO}(\neq(\cdot, \cdot), \not\subseteq, \upharpoonright, \not\perp_c)$ is equivalent to some sentence of $\text{FO}(=(\cdot), \neq(\cdot, \cdot), \upharpoonright)$, and hence to some first-order sentence.

6 Bounded Dependencies and Totality

Now that we know something about upwards closed dependencies, it would be useful to classify them in different categories and prove *non-definability* results between the corresponding extensions of first-order logic. As a first such classification, we introduce the following property:

Definition 28 (Boundedness) Let κ be a (finite or infinite) cardinal. A dependency condition \mathbf{D} is κ -bounded if whenever $M \models_X \mathbf{D}\vec{x}$ there exists a $Y \subseteq X$ with $|Y| \leq \kappa$ such that $M \models_Y \mathbf{D}\vec{x}$.

We say that \mathbf{D} is bounded if it is κ -bounded for some κ .¹¹

For example, non-emptiness and intersection are 1-bounded; inconstancy and the negations of functional dependence and exclusion are 2-bounded; and for all finite or infinite κ , κ -bigness is κ -bounded. However, totality is not bounded at all. Indeed, for any κ consider a model M of cardinality greater than κ and take the team $X = \{\emptyset\}[M/x]$. Then $M \models_X \text{All}(x)$, but if $Y \subseteq X$ has cardinality $\leq \kappa$ then $Y(x) \subsetneq \text{Dom}(M)$ and $M \not\models_Y \text{All}(x)$.

As we will now see, the property of boundedness is preserved by the connectives of our language.

Definition 29 (Height of a formula) Let \mathcal{D} be any family of bounded dependencies. Then for all formulas $\phi \in \text{FO}(\mathcal{D})$, the height $ht(\phi)$ of ϕ is defined as follows:

¹¹After a fashion, this notion of boundedness may be thought of as a dual of the notion of *coherence* of [15].

1. If ϕ is a first-order literal then $\text{ht}(\phi) = 0$;
2. If ϕ is a functional dependence atom $\mathbf{D}\vec{x}$ then $\text{ht}(\phi)$ is the least cardinal κ such that \mathbf{D} is κ -bounded;
3. If ϕ is of the form $\psi_1 \vee \psi_2$ or $\psi_1 \wedge \psi_2$ then $\text{ht}(\phi) = \text{ht}(\psi_1) + \text{ht}(\psi_2)$;
4. If ϕ is of the form $\exists v\psi$ or $\forall v\psi$. then $\text{ht}(\phi) = \text{ht}(\psi)$.

In other words, the height of a formula is the sum of the heights of all instances of dependency atoms occurring in it.

Theorem 30 *Let \mathcal{D} be a family of bounded upwards closed dependencies. Then for all formulas $\phi \in \text{FO}(\mathcal{D})$*

$$M \models_X \phi \Rightarrow \exists Y \subseteq X \text{ with } |Y| \leq \text{ht}(\phi) \text{ s.t. } M \models_Y \phi.$$

Proof:

The proof is by induction on ϕ .

1. If ϕ is a first-order literal then $\text{ht}(\phi) = 0$ and it is always the case that $M \models_{\emptyset} \phi$, as required.
2. If ϕ is an atom $\mathbf{D}\vec{x}$ then the statement follows at once from the definitions of boundedness and height.
3. If ϕ is a disjunction $\psi_1 \vee \psi_2$ then $\text{ht}(\phi) = \text{ht}(\psi_1) + \text{ht}(\psi_2)$. Suppose now that $M \models_X \psi_1 \vee \psi_2$: then $X = X_1 \cup X_2$ for two X_1 and X_2 such that $M \models_{X_1} \psi_1$ and $M \models_{X_2} \psi_2$. This implies that there exist $Y_1 \subseteq X_1$, $Y_2 \subseteq X_2$ such that $M \models_{Y_1} \psi_1$ and $M \models_{Y_2} \psi_2$, $|Y_1| \leq \text{ht}(\psi_1)$ and $|Y_2| \leq \text{ht}(\psi_2)$. But then $Y = Y_1 \cup Y_2$ satisfies $\psi_1 \vee \psi_2$ and has at most $\text{ht}(\psi_1) + \text{ht}(\psi_2)$ elements.
4. If ϕ is a conjunction $\psi_1 \wedge \psi_2$ then, again, $\text{ht}(\phi) = \text{ht}(\psi_1) + \text{ht}(\psi_2)$. Suppose that $M \models_X \psi_1 \wedge \psi_2$: then $M \models_X \psi_1$ and $M \models_X \psi_2$, and therefore by Lemma 13 $M \models_X \psi_1^f$ and $M \models_X \psi_2^f$; and, by induction hypothesis, there exist $Y_1, Y_2 \subseteq X$ with $|Y_1| \leq \text{ht}(\psi_1)$, $|Y_2| \leq \text{ht}(\psi_2)$, $M \models_{Y_1} \psi_1$ and $M \models_{Y_2} \psi_2$. Now let $Y = Y_1 \cup Y_2$: since $Y \subseteq X$, by Proposition 4 $M \models_Y \psi_1^f$ and $M \models_Y \psi_2^f$. But $Y_1, Y_2 \subseteq Y$, and therefore by Theorem 14 $M \models_Y \psi_1$ and $M \models_Y \psi_2$, and in conclusion $M \models_Y \psi_1 \wedge \psi_2$.
5. If ϕ is of the form $\exists v\psi$ then $\text{ht}(\phi) = \text{ht}(\psi)$. Suppose that $M \models_X \exists v\psi$: then for some H we have that $M \models_{X[H/v]} \psi$, and therefore by induction hypothesis there exists a $Z \subseteq X[H/v]$ with $|Z| \leq \text{ht}(\psi)$ such that $M \models_Z \psi$. For any $h \in Z$, let $\mathfrak{f}(h)$ be a $s \in X$ such that $h \in s[H/v] = \{s[m/v] : m \in H(s)\}$,¹² and let $Y = \{\mathfrak{f}(h) : h \in Z\}$. Now $Z \subseteq Y[H/v] \subseteq X[H/v]$. Since $M \models_{X[H/v]} \psi^f$ and $Y[H/v] \subseteq X[H/v]$, we have that $M \models_{Y[H/v]} \psi^f$; and since $M \models_Z \psi$, this implies that $M \models_{Y[H/v]} \psi$ and that $M \models_Y \exists v\psi$. Furthermore $|Y| = |Z| \leq \text{ht}(\psi)$, as required.
6. If ϕ is of the form $\forall v\psi$ then, again, $\text{ht}(\phi) = \text{ht}(\psi)$. Suppose that $M \models_{X[M/v]} \psi$: again, by induction hypothesis there is a $Z \subseteq X[M/v]$ with $|Z| \leq \text{ht}(\psi)$ and such that $M \models_Z \psi$. For any $h \in Y$, let $\mathfrak{g}(h)$ pick some $s \in X$ which agrees with h on all variables except v , and let $Y = \{\mathfrak{g}(h) : h \in Z\}$. Similarly to the previous case, $Z \subseteq Y[M/v] \subseteq X[M/v]$: therefore, since $M \models_{X[M/v]} \psi^f$ we have that $M \models_{Y[M/v]} \psi^f$, and since $M \models_Z \psi$ we have that $M \models_{Y[M/v]} \psi$. So in conclusion $M \models_Y \forall v\psi$, as required, and $|Y| = |Z| \leq n$.

□

Even though constancy atoms are not upwards closed, it is possible to extend this result to $\text{FO}(=(\cdot), \mathcal{D})$. Indeed, constancy atoms are trivially 0-bounded, since the empty team always satisfies them, and

¹²Since $Z \subseteq X[H/v]$, such a s always exists. Of course, there may be multiple ones; in that case, we pick one arbitrarily.

Corollary 31 *Let \mathcal{D} be a family of upwards closed bounded dependencies. Then for all $\phi \in FO(=(\cdot), \mathcal{D})$*

$$M \models_X \phi \Rightarrow \exists Y \subseteq X \text{ with } |Y| \leq \text{ht}(\phi) \text{ s.t. } M \models_Y \phi.$$

Proof:

Let $\phi \in FO(=(\cdot), \mathcal{D})$: then by Lemma 22 ϕ is equivalent to some formula of the form $\exists \vec{v}(=(\vec{v}) \wedge \psi)$, where ψ does not contain constancy atoms and $\text{ht}(\psi) = \text{ht}(\phi)$. Now suppose that $M \models_X \phi$: then, for some choice of elements $\vec{m} \in \text{Dom}(M)^{|\vec{v}|}$, $M \models_{X[\vec{m}/\vec{v}]} \psi$. Now by Theorem 30 there exists a $Z \subseteq X[\vec{m}/\vec{v}]$, with $|Z| \leq \text{ht}(\psi)$, such that $M \models_Z \psi$; and Z is necessarily of the form $Y[\vec{m}/\vec{v}]$ for some $Y \subseteq X$ with $|Y| = |Z| \leq \text{ht}(\psi)$. But then $M \models_Y \exists \vec{v}(=(\vec{v}) \wedge \psi)$, as required.

□

This result allows us to prove at once a number of nondefinability results concerning upwards closed dependencies. For example, it is now easy to see that

Corollary 32 *Let \mathcal{D} be a family of upwards closed bounded dependencies. Then the totality dependency $A\forall$ is not definable in $FO(=(\cdot), \mathcal{D})$. In particular, totality atoms cannot be defined by means of the negations of inclusion, exclusion, functional dependence and independence atoms.*

Corollary 33 *Let \mathcal{D} be a family of κ -bounded upwards closed dependencies and let $\kappa' > \kappa$ be infinite. Then κ' -bigness is not definable in $FO(=(\cdot), \mathcal{D})$.*

Corollary 34 *Let \mathbf{D} be a k -bounded upwards closed dependency, and let $n > k$. If $\phi(\vec{x})$ of $FO(=(\cdot), \mathbf{D})$ characterizes n -bigness, in the sense that for all M and X*

$$M \models_X \phi(\vec{x}) \Leftrightarrow |X(\vec{x})| \geq n,$$

then $\phi(\vec{x})$ contains at least $\lceil \frac{n}{k} \rceil$ instances of \mathbf{D} .

7 Conclusions and Further Work

In this work we discovered a surprising asymmetry between downwards closed and upwards closed first-order definable dependency conditions: whereas, as it was known since [19], the former can bring the expressive power of a logic with team semantics beyond the first order, the latter cannot do so by their own or even together with constancy atoms. As a consequence, the negations of the principal dependency notions studied so far in team semantics can all be added to first-order logic without increasing its expressive power.

Our original question was: how much can we get away with adding to the team semantics of first-order logic before ending up in a higher order logic? The answer, it is now apparent, is *quite a lot*. This demonstrates that team semantics is useful not only (as it has been employed so far) as a formalism for the study of very expressive extensions of first-order logic, but also as one for that of more treatable ones.

Much of course remains to be done. The notion of boundedness of Section 6 allowed us to find some non-definability results between our extensions; but the classification of these extensions is far from complete. In particular, it would be interesting to find necessary and sufficient conditions for $FO(\mathcal{D})$ to be equivalent to FO over sentences. The complexity-theoretic properties of these logics, or of fragments thereof, also deserve further investigation.

Another open issue concerns the development of sound and complete proof systems for our logics. Of course, one can check whether a theory T implies a formula ϕ simply by using Theorems 11 and 21 to translate everything in first-order logic and then use one of the many well-understood proof systems for it; but nonetheless, it could be very informative to find out directly which logical laws our formalisms obey.

Acknowledgments The author thanks the referees for a number of useful suggestions and corrections.

References

- [1] Samson Abramsky & Jouko Väänänen (2013): *Dependence logic, social choice and quantum physics*. In preparation.
- [2] William W. Armstrong (1974): *Dependency Structures of Data Base Relationships*. In: *Proc. of IFIP World Computer Congress*, pp. 580–583.
- [3] Marco A. Casanova & Vânia M. P. Vidal (1983): *Towards a sound view integration methodology*. In: *Proceedings of the 2nd ACM SIGACT-SIGMOD symposium on Principles of database systems, PODS '83*, ACM, New York, NY, USA, pp. 36–47, doi:10.1145/588058.588065.
- [4] Fredrik Engström (2012): *Generalized quantifiers in dependence logic*. *Journal of Logic, Language and Information* 21(3), pp. 299–324, doi:10.1007/s10849-012-9162-4.
- [5] Ronald Fagin (1977): *Multivalued dependencies and a new normal form for relational databases*. *ACM Transactions on Database Systems* 2, pp. 262–278, doi:10.1145/320557.320571.
- [6] Ronald Fagin (1981): *A normal form for relational databases that is based on domains and keys*. *ACM Transactions on Database Systems* 6, pp. 387–415, doi:10.1145/319587.319592.
- [7] Pietro Galliani (2012): *The Dynamics of Imperfect Information*. Ph.D. thesis, University of Amsterdam. Available at <http://dare.uva.nl/record/425951>.
- [8] Pietro Galliani (2012): *Inclusion and exclusion dependencies in team semantics: On some logics of imperfect information*. *Annals of Pure and Applied Logic* 163(1), pp. 68 – 84, doi:10.1016/j.apal.2011.08.005.
- [9] Pietro Galliani & Lauri Hella (2013): *Inclusion Logic and Fixed Point Logic*. ArXiv:1304.4267.
- [10] Pietro Galliani & Jouko Väänänen (2013): *On Dependence Logic*. ArXiv:1305.5948.
- [11] Dan Geiger, Azaria Paz & Judea Pearl (1991): *Axioms and algorithms for inferences involving probabilistic independence*. *Information and Computation* 91(1), pp. 128 – 141, doi:10.1016/0890-5401(91)90077-F.
- [12] Erich Grädel & Jouko Väänänen (2013): *Dependence and Independence*. *Studia Logica* 101(2), pp. 399–410, doi:10.1007/s11225-013-9479-2.
- [13] Jaakko Hintikka & Gabriel Sandu (1989): *Informational independence as a semantic phenomenon*. In J.E. Fenstad, I.T. Frolov & R. Hilpinen, editors: *Logic, methodology and philosophy of science*, Elsevier, pp. 571–589, doi:10.1016/S0049-237X(08)70066-1.
- [14] Wilfrid Hodges (1997): *Compositional Semantics for a Language of Imperfect Information*. *Journal of the Interest Group in Pure and Applied Logics* 5 (4), pp. 539–563, doi:10.1093/jigpal/5.4.539.
- [15] Jarmo Kontinen (2013): *Coherence and Computational Complexity of Quantifier-free Dependence Logic Formulas*. *Studia Logica* 101(2), pp. 267–291, doi:10.1007/s11225-013-9481-8.
- [16] Juha Kontinen & Ville Nurmi (2009): *Team Logic and Second-Order Logic*. In Hiroakira Ono, Makoto Kanazawa & Ruy de Queiroz, editors: *Logic, Language, Information and Computation, Lecture Notes in Computer Science* 5514, Springer Berlin / Heidelberg, pp. 230–241, doi:10.1007/978-3-642-02261-6_19.
- [17] Antti Kuusisto (2013): *Defining a Double Team Semantics for Generalized Quantifiers (Extended Version)*. Available at https://uta17-kk.lib.helsinki.fi/bitstream/handle/10024/68064/defining_double_team_2013.pdf?sequence=1. Manuscript.
- [18] Allen L. Mann, Gabriel Sandu & Merlijn Sevenster (2011): *Independence-Friendly Logic: A Game-Theoretic Approach*. Cambridge University Press, doi:10.1017/CB09780511981418.
- [19] Jouko Väänänen (2007): *Dependence Logic*. Cambridge University Press, doi:10.1017/CB09780511611193.
- [20] Jouko Väänänen & Fan Yang (2013): *Propositional dependence logic*. Manuscript.