Counting in Team Semantics

Erich Grädel[1] and Stefan Hegselmann[2]

1 Mathematical Foundations of Computer Science, RWTH Aachen University, Aachen, Germany
graedel@logic.rwth-aachen.de

2 Mathematical Foundations of Computer Science, RWTH Aachen University, Aachen, Germany
hegselmann@logic.rwth-aachen.de

Abstract

We explore several counting constructs for logics with team semantics. Counting is an important task in numerous applications, but with a somewhat delicate relationship to logic. Team semantics on the other side is the mathematical basis of modern logics of dependence and independence, in which formulae are evaluated not for a single assignment of values to variables, but for a set of such assignments. It is therefore interesting to ask what kind of counting constructs are adequate in this context, and how such constructs influence the expressive power, and the model-theoretic and algorithmic properties of logics with team semantics. Due to the second-order features of team semantics there is a rich variety of potential counting constructs. Here we study variations of two main ideas: *forking atoms* and *counting quantifiers*.

Forking counts how many different values for a tuple $\overline{w}$ occur in assignments with coinciding values for $\overline{v}$. We call this the *forking degree* of $\overline{v}$ with respect to $\overline{w}$. Forking is powerful enough to capture many of the previously studied atomic dependency properties. In particular we exhibit logics with forking atoms that have, respectively, precisely the power of dependence logic and independence logic.

Our second approach uses counting quantifiers $\exists^{\geq\mu}$ of a similar kind as used in logics with Tarski semantics. The difference is that these quantifiers are now applied to teams of assignments that may give different values to μ. We show that, on finite structures, there is an intimate connection between inclusion logic with counting quantifiers and FPC, fixed-point logic with counting, which is a logic of fundamental importance for descriptive complexity theory. For sentences, the two logics have the same expressive power. Our analysis is based on a new variant of model-checking games, called threshold safety games, on a trap condition for such games, and on game interpretations.

1998 ACM Subject Classification F.4.1 Mathematical Logic

Keywords and phrases logics with counting, team semantics, fixed-point logic with counting

Digital Object Identifier 10.4230/LIPIcs.CSL.2016.35

1 Introduction

> Ravens, so we read, can only count up to seven. They can't tell the difference between two numbers greater than or equal to eight. First-order logic is much the same as ravens, except that the cutoff point is rather higher: it's ω instead of 8. Wilfrid Hodges

Logic and counting. Counting the number of elements satisfying a certain property is a basic task of fundamental importance that arises in many applications. While this is computationally easy (if the underlying property is decidable in a simple way) it is problematic

25th EACSL Annual Conference on Computer Science Logic (CSL 2016).
Editors: Jean-Marc Talbot and Laurent Regnier; Article No. 35; pp. 35:1–35:18
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

for many classical logical systems. The quotation by Hodges [9] only shows the tip of the iceberg. If we do not look at first-order theories but at single sentences, then the cutoff point (depending on vocabulary and quantifier-rank) may be much lower, and the inability to count persists for logics that are much stronger than first-order logic, such as for instance fixed-point logics. In particular, the computationally trivial query of determining whether a given finite structure has an even or odd number of elements is not definable by any formula from first-order logic (FO), fixed-point logic (LFP), or even from $L^{\omega}_{\infty\omega}$, the infinitary logic with a bounded number of variables.

In finite model theory, a lot of attention has therefore been given to logics that incorporate counting in some way or another, for instance by counting quantifiers or counting terms, or by generalized quantifiers for cardinality comparison such as Rescher or Härtig quantifiers. The simplest way to add counting is in terms of quantifiers of form $\exists^{\geq i} x\varphi$ for fixed $i \in \mathbb{N}$, saying that there exist at least i distinct values for x satisfying φ. While such counting quantifiers do not add anything to the expressive power of full first-order logic or fixed-point logic, they are relevant for the study of logics with a bounded number of variables, such as C^k or $C^k_{\infty\omega}$. A more powerful variant of counting is obtained with counting terms of form $\#\varphi(x)$ or counting quantifiers $\exists^{\geq\mu}$ or $\exists^{\leq\mu}$ where μ is a variable, and the values assumed by μ or by counting terms are natural numbers (that are kept separate from the elements of the structure). Thus, logics with this kind of counting are evaluated on two-sorted structures $\mathfrak{A}^* = \mathfrak{A} \cup (\omega, <, +, \cdot, 0, e)$, i.e., finite structures expanded by a disjoint ordered numeric sort (here the natural numbers with arithmetic but there are other possible choices). The most important counting logic, at least in finite model theory, is fixed-point logic with counting (FPC). It has first been proposed, somewhat informally, by Immerman, later a more formal definition based on two-sorted structures, counting terms, and inflationary fixed-points of relations ranging over both sorts has been adopted. Meanwhile FPC has become the logic of reference in the search for a logic for PTIME. Although it has been known since the 1990s, by a fundamental construction due to Cai, Fürer, and Immerman, that FPC fails to express all polynomial-time queries, it comes rather close to being a logic for polynomial time. It is strong enough to express most of the fundamental algorithmic techniques leading to polynomial-time procedures and it captures PTIME on many interesting classes of finite structures, including trees, planar graphs, structures of bounded tree width, and actually all classes of graphs with an excluded minor. For a survey on FPC, and for references, see [1].

Logics with team semantics. In this paper, we study counting constructs for team semantics. The idea of team semantics goes back to a paper by Hodges [10] where he provided a model-theoretic, compositional, semantics for the independence-friendly logic IF, as an alternative to the semantics based on games of imperfect information or on Skolem functions. In team semantics a formula $\varphi(x_1, \dots x_k)$ is evaluated, on a given structure $\mathfrak{A}$, not for a single assignment $s : \{x_1, \dots, x_k\} \to A$ but for a *set* of such assignments, and, following [15], a set of assignments with a common (finite) domain of variables is called a team. Personally we find that the invention of team semantics by Wilfrid Hodges is really a major innovation in logic. Combined with Väänänen's proposal [15] to treat dependencies as atomic statements, and not as annotations of quantifiers, it has lead to a genuinely new area in logic, with an interdisciplinary motivation of providing logical systems for reasoning about the fundamental notions of dependence and independence that permeate many scientific disciplines. Methods from several areas of computer science, including finite model theory, database theory, and the algorithmic analysis of games have turned out as highly relevant for this area.

Notice that statements about dependence or independence, such as *"z is functionally dependent on x and y"* or *"x and y are independent"* do not make much sense for a single assignment to the variables, but require larger amounts of data, as given by a table or relation, or by a team of assignments. Team semantics is therefore the natural mathematical basis for the modern logics of dependence and independence in which dependency or independency statements are basic atomic building blocks, similar to equality statements. The best studied logic with team semantics is dependence logic, which extends first-order logic by dependency atoms of form $=(\overline{x}, \overline{y})$, saying that the variables $\overline{y}$ are functionally dependent on (i.e. completely determined by) the variables $\overline{x}$, but there are many other atomic dependence properties that give rise to interesting logics based on team semantics. In [7] we have discussed the notion of independence (which is a much more delicate but also more powerful notion than dependence) and introduced independence logics, and Galliani [4] and Engström [3] have studied several logics with team properties based on notions originating in database dependency theory. Of particular interest for us is *inclusion logic* $\mathrm{FO}(\subseteq)$ which extends first-order logic by atomic inclusion dependencies $(\overline{x} \subseteq \overline{y})$, which are true in a team X if every value for $\overline{x}$ in X also occurs as a value for $\overline{y}$ in X. There is also a dual notion, exclusion logic, based on exclusion statements $(\overline{x} \mid \overline{y})$, saying that $\overline{x}$ and $\overline{y}$ have disjoint sets of values in the team X. Exclusion logic has turned out to be equivalent to dependence logic [4].

Expressive power of logics with team semantics. If we study the expressive power of such logics, for instance by comparison to classical logics, we have to keep in mind the different nature of team semantics and Tarski semantics. For a formula with team semantics, we write $\mathfrak{A} \models_X \varphi$ to denote that φ is true in the structure $\mathfrak{A}$ for the team X, and for classical Tarski semantics we write $\mathfrak{A} \models_s \varphi$ to denote that φ is true in $\mathfrak{A}$ for the assignment s. A direct comparison is possible in the case of sentences. For any sentence ψ from a logic with team semantics, we write $\mathfrak{A} \models \psi$ if $\mathfrak{A} \models_{\{\emptyset\}} \psi$, i.e. if ψ is true for the team $X = \{\emptyset\}$ that consists just of the empty assignment[1]. For formulae with free variables the translation from a logic with team semantics into one with Tarski semantics requires that we represent the team in some way. The standard way to do this is by identifying a team X of assignments $s : \{x_1, \dots x_k\} \to A$ with the relation $\{s(\overline{x}) \in A^k : s \in X\} \subseteq A^k$ which, by slight abuse of notation, we also denote by X. One then translates formulae $\varphi(x_1, \dots, x_k)$ of vocabulary τ into *sentences* φ^* of the expanded vocabulary $\tau \cup \{X\}$ such that for every structure $\mathfrak{A}$ and every team X we have that

$$\mathfrak{A} \models_X \varphi(x_1, \dots, x_k) \iff (\mathfrak{A}, X) \models \varphi^*.$$

In all logics with team semantics that extend first-order logic (or a fragment thereof) by atomic dependency statements that are themselves first-order definable, and which do not make use of additional connectives beyond $\wedge, \vee$ and atomic negation, such a translation will always produce sentences in (a fragment of) existential second-order logic, denoted Σ^1_1. Understanding the expressive power of a logic L with team semantics thus means to identify the fragment of Σ^1_1 to which L is equivalent in the sense just described. The following is known in this context:

1. Dependence logic is equivalent to the fragment of all Σ^1_1-sentences $\psi(X)$ in which the predicate X describing the team appears only negatively [12].
2. Independence logic and inclusion-exclusion logic are equivalent with full Σ^1_1 (and thus can describe all NP-properties of teams) [4].

[1] Notice that we cannot replace this by the empty team $X = \emptyset$. The common logics with teams semantics have the empty team property which means that the empty team satisfies all formulae.

3. The extension of FO by inclusion and exclusion atoms of single variables only (not tuples of variables) is equivalent to monadic Σ_1^1 [14].
4. First-order logic without any dependence atoms has the so-called *flatness property*: $\mathfrak{A} \models_X \varphi \iff \mathfrak{A} \models_{\{s\}} \varphi$ for all $s \in X$. It thus corresponds to a very small fragment of Σ_1^1, namely FO-sentences of form $\forall \overline{x}(X\overline{x} \to \varphi(\overline{x}))$ where $\varphi(\overline{x})$ does not contain X.

For our study the most interesting result of this kind concerns the relationship of inclusion logic FO($\subseteq$) with posGFP, the fragment of LFP that uses only (non-negated) greatest fixed points. Since a greatest fixed-point formula $[\mathbf{gfp}\, R\overline{x} \,.\, \psi(R, \overline{x})](\overline{y})$ readily translates into $(\exists R)((\forall \overline{x}(R\overline{x} \to \psi(R, \overline{x})) \land R\overline{y}))$, posGFP can be viewed as a fragment of Σ_1^1. Galliani and Hella [5] established translations between inclusion logic and posGFP that extend the list of equivalences between logics with team semantics and fragments of Σ_1^1 by

5. Inclusion logic is equivalent to the set of sentences of form $\forall \overline{x}(X\overline{x} \to \psi(X, \overline{x}))$, where $\psi(X, \overline{x})$ is a formula in posGFP in which X occurs only positively. In particular, on any finite structure, the maximal team satisfying a formula $\varphi(\overline{x})$ of inclusion logic coincides with the greatest fixed point of (the operator defined by) the posGFP-formula $\psi(X, \overline{x})$.

A different proof for this result, based on safety games and game interpretations, has been presented in [6]. Notice that for sentences, inclusion logic and posGFP have the same expressive power. It is known that, on finite structures, the full logic LFP collapses to its posGFP-fragment [11]. Hence every property of finite structures that is LFP-definable is also definable in inclusion logic, and vice versa. It follows by the Immerman-Vardi-Theorem that, on *ordered* finite structures, inclusion logic captures polynomial time.

Counting constructs for team semantics. The relevance of logics with counting for finite model theory, and the known connections between logics with team semantics (such as inclusion logic) and logics that are important in finite model theory (such as LFP), raise the question of counting constructs for logics with team semantics, and how these enhance the expressive power of such logics. The second-order nature of team semantics in fact leads to a rich variety of potential counting constructs for logics with team semantics, because there are several different objects that one may wish to count.

We illustrate this variability with the well-known *majority quantifier* M. In Tarski semantics a formula $My\varphi(\overline{x}, y)$ expresses, on a finite structure $\mathfrak{A}$ and an assignment $s : \overline{x} \mapsto \overline{a}$, that $\mathfrak{A} \models \varphi(\overline{a}, b)$ for at least half of the possible values b for y. In particular, $MyExy$ is true for a graph $G = (V, E)$ and an assignment $s : x \mapsto v$ if v is adjacent to at least half of the nodes in G. There are several possibilities to define the team semantics of $MyExy$. A team X of assignments to x describes a subset $U := X[x] \subseteq V$. We may define that $G \models_X MyExy$ if *every node in* U is adjacent to at least half of the nodes in V. We shall see that this corresponds to defining majority via *counting quantifiers.* But we could also define that $G \models_X MyExy$ if U *as a set* is adjacent to at least half of the nodes in G, or, that at least half of the nodes in G are adjacent to *all* elements of U. These two possibilities are related to a different kind of counting that we call *forking.*

In all three cases, this amounts to a general definition of the team semantics of the majority quantifier, saying that $\mathfrak{A} \models_X My\varphi(\overline{v}, y)$ if, and only if, $\mathfrak{A} \models_{X[y \mapsto F]} \varphi$ for some appropriate function $F : X \to \mathcal{P}(A)$, but the requirements for F are different. In the first case we demand that $|F(s)| \geq |A|/2$ for all $s \in X$, in the second case that $|\bigcup_{s \in X} F(s)| \geq |A|/2$, and in the third case that $F(s) = B$ for all $s \in X$ and some fixed set $B \subseteq A$ with $|B| \geq |A|/2$.

In the sequel, we shall focus on variations of two main ideas: *forking* and *counting quantifiers.* Forking counts, for variables $\overline{v}$ and $\overline{w}$, how many different values for $\overline{w}$ occur in assignments with coinciding values for $\overline{v}$. We call this the *forking degree* of $\overline{v}$ with respect to $\overline{w}$. Notice that a forking degree of one is equivalent to functional dependence, so forking can be seen as a generalization of dependence. We shall introduce a counting mechanism for the forking degree by means of *forking atoms* over teams. We examine closure properties and expressive power of different extensions of first-order logic with forking atoms.

Our second approach uses counting quantifiers $\exists^{\geq\mu}$ of a similar kind as used in, say, fixed-point logic with counting. The difference is that these quantifiers are now applied to teams of assignments that may give different values to μ. A counting quantifier $\exists^{\geq\mu}x$ thus requires a witness that extends a given team X by adding to each $s \in X$ a set of at least $s(\mu)$ many distinct values for x. A formal definition will be given in Sect. 4. It is not hard to see that in logics that have the full power of Σ^1_1 (or come close to it) this form of counting is definable, even without a second numeric sort. Thus counting quantifiers are interesting mainly for logics with team semantics that are strictly weaker than Σ^1_1, in particular for inclusion logic. We shall analyse the power of logics with counting by an appropriate variant of games that we call threshold safety games. Our main result will indeed give an equivalence, in the sense described above, between inclusion logic with counting, $\text{FO}(\subseteq, \exists^{\geq\mu})$, and fixed-point logic with counting, FPC.

2 Preliminaries

Two-sorted structures. With every finite relational structure $\mathfrak{A}$, we associate the two-sorted structure $\mathfrak{A}^* = \mathfrak{A} \cup (\omega, <, +, \cdot, 0, e)$, where A (the universe of $\mathfrak{A}$) and ω (the set of natural numbers) are assumed to be disjoint. Let $A^* := A \cup \omega$. We call $\mathfrak{A}$ the point sort, and $(\omega, <, +, \cdot, 0, e)$ the numeric sort. The numeric constant e stands for $|A|$, the cardinality of the point sort. All variables are typed: we use Latin letters $x, y, z, \dots$ for variables over the point sort, and Greek letters $\mu, \nu, \lambda, \dots$ for variables over the numeric sort. Whenever the sort of a variable is irrelevant and not clear from context, we shall use the variable symbols v or w. For some applications, it is relevant to generalize such structures to expansions $(\mathfrak{A}^*, F)$ with a set F of functions $f : A^k \to \omega$.

Quantifiers over the numeric sort must be bounded, of the form $(\forall\mu < t)$ or $(\exists\mu < t)$, where t is a closed numeric term (but for ease of notation we shall suppress this in cases where the bounds are irrelevant or clear from context). Such bounds could be avoided altogether, if one would use a finite numeric sort, such as $(\{0, \dots, |A|\}, <, 0, e)$ instead of the natural numbers. Both approaches have their advantages, but here we use an infinite numeric sort to make the counting of tuples of elements simpler. We write $\omega_{<t}$ for the set of natural numbers smaller than (the value of) t.

Team semantics. From the start, we adapt the notion of teams from [15] to two-sorted structures $\mathfrak{A}^*$. For a set $\mathcal{V}$ of (typed) variables, an *assignment* into $\mathfrak{A}^*$ is a map $s : \mathcal{V} \to A^*$ that respects the types of the variables. Given such an assignment s, a tuple $\overline{v} = (v_1, \dots, v_k)$ of distinct variables, and elements $c_1, \dots, c_k \in A^*$ (with types corresponding to those of $\overline{v}$) we write $s[\overline{v} \mapsto \overline{c}]$ for the assignment with domain $\mathcal{V} \cup \{v_1, \dots, v_k\}$ that updates s by mapping v_i to c_i, for $i = 1, \dots, k$.

▶ **Definition 1.** A *team* is a set of assignments with the same domain into a structure $\mathfrak{A}^*$. For a tuple of variables $\bar{v} = (v_1, ..., v_n)$, let $s(\bar{v}) := (s(v_1), ..., s(v_n))$ and let $X[\bar{v}] := \{s(\bar{v}) : s \in X\}$ denote the set of values for $\overline{v}$ in X. For a team X, a k-tuple $\overline{v}$, and a function $F : X \to \mathcal{P}((A^*)^k)$, we write $X[\overline{v} \mapsto F]$ for the set of all assignments $s[\overline{v} \mapsto \overline{c}]$ with $s \in X$ and $\overline{c} \in F(s)$.

Team semantics, for a logic L, defines whether a formula $\psi \in L$ is satisfied by a team X in a structure $\mathfrak{A}^*$, written $\mathfrak{A}^* \models_X \psi$. We always assume formulae to be in negation normal form and require that X is correctly typed, with a domain that contains all free variables of ψ. Following the approach by Väänänen [15], modern logics of dependence and independence are based on atomic dependency properties of teams. The most important ones, for this paper, are

Dependence: A dependence atom has the form $=\!(\overline{v}, \overline{w})$. It is true in a team X if, and only if, $s(\overline{w}) = s'(\overline{w})$ for all assignments $s, s' \in X$ with $s(\overline{v}) = s'(\overline{v})$.

Constancy: A special case of a dependence atom is the constancy atom $=\!(v)$, saying that v assumes only one value in X.

Independence: A (pure) independence atom[2] has the form $\overline{v} \bot \overline{w}$. It holds in a team X if, and only if, for all $s, s' \in X$ there is a third assignment $s'' \in X$ such that $s''(\overline{v}) = s(\overline{v})$ and $s''(\overline{w}) = s'(\overline{w})$.

Inclusion: An inclusion atom has the form $(\overline{v} \subseteq \overline{w})$, for tuples $\overline{v}, \overline{w}$ of the same length and type. It is true in a team X if, and only if, $X[\overline{v}] \subseteq X[\overline{w}]$, i.e. if every value that occurs for $\overline{v}$ in X also occurs as a value for $\overline{w}$.

In the logics with team semantics that we consider, atomic dependency properties are used only positively. Thus, negation is applied only to first-order atoms. A first-order literal $\alpha(\overline{v})$ is defined to be true for a team X in the structure $\mathfrak{A}^*$ if it is true, in the sense of classical Tarski semantics, for all individual assignments $s \in X$, i.e.

$$\mathfrak{A}^* \models_X \alpha(\overline{v}) \iff \mathfrak{A}^* \models_s \alpha(\overline{v}) \text{ for all } s \in X.$$

The common logics of dependence and independence extend first-order literals and atomic dependency properties by the usual first-order connectives and quantifiers $\vee, \wedge, \exists, \forall$ to obtain full-fledged logics for reasoning about dependency properties. The semantic rules for these, in the context of two-sorted structures, are the following:

- $\mathfrak{A}^* \models_X (\varphi \wedge \vartheta)$ if, and only if, $\mathfrak{A}^* \models_X \varphi$ and $\mathfrak{A}^* \models_X \vartheta$.
- $\mathfrak{A}^* \models_X (\varphi \vee \vartheta)$ if, and only if, there exist teams Y, Z with $X = Y \cup Z$ such that $\mathfrak{A}^* \models_Y \varphi$ and $\mathfrak{A}^* \models_Z \vartheta$.
- $\mathfrak{A}^* \models_X \exists y \varphi$ if, and only if, there is a map $F : X \to (\mathcal{P}(A) \setminus \{\emptyset\})$ such that $\mathfrak{A}^* \models_{X[y \mapsto F]} \varphi$.
- $\mathfrak{A}^* \models_X \forall y \varphi$ if, and only if, $\mathfrak{A}^* \models_{X[y \mapsto A]} \varphi$ where $X[y \mapsto A] := \{s[y \mapsto a] : s \in X, a \in A\}$.
- $\mathfrak{A}^* \models_X (\exists \mu < t)\varphi$ if, and only if, there is a map $F : X \to (\mathcal{P}(\omega_{<t}) \setminus \{\emptyset\})$ such that $\mathfrak{A} \models_{X[\mu \mapsto F]} \varphi$.
- $\mathfrak{A}^* \models_X (\forall \mu < t)\varphi$ if, and only if, $\mathfrak{A} \models_{X[\mu \mapsto \omega_{<t}]} \varphi$ where $X[\mu \mapsto \omega_{<t}] := \{s[\mu \mapsto m] : s \in X, m < t\}$.

First-order logic extended with dependence atoms, constancy atoms, inclusion atoms, and independence atoms, are, respectively called dependence logic FO(dep), constancy logic FO(const), independence logic FO(indep), and inclusion logic FO($\subseteq$).

The locality principle. In our definitions we shall always make sure that the *locality principle* holds, saying that the meaning of a formula can only depend on the variables actually occurring in it. More precisely, if $Y = X \upharpoonright \text{free}(\psi)$ is the restriction of the team X to the free variables of ψ then $\mathfrak{A} \models_X \psi$ if, and only if, $\mathfrak{A} \models_Y \psi$. As obvious as it seems, the

[2] There is also a more general variant of independence atoms, but it is known that these are expressible by means of pure independence atoms.

locality principle is not trivial, and it it easy to violate it by choosing 'wrong' definitions. Consider for instance the seemingly more natural semantics (called strict semantics) for existential quantifiers, that defines a formula $\exists y\varphi$ to hold for a team X if, and only if, there exists a function $F : X \to A$ such that φ holds for the team of all assignments $s[y \mapsto F(s)]$ with $s \in X$. While such a choice of a single witness, rather than a non-empty set of witnesses, for an existentially quantified variable would make no difference for downwards closed logics such as pure first-order logic or dependence logic, it would lead to a violation of the locality principle when combined with, say, inclusion atoms. Indeed, even the simple formula $\exists x(y \subseteq x \wedge z \subseteq x)$, which trivially holds for all teams under the semantics given above, may depend under strict semantics on the presence of additional variables in the domain of the team.

More importantly for us, the locality principle may be violated by certain forms of counting, such as counting quantifiers of form $\exists^{\leq\mu}x$. Thus the locality principle provides a kind of sanity check for the definition of logical operators in team semantics, and we shall make sure that all our proposals pass this check.

Closure properties. One way to study the properties and power of logics with team semantics is based on the operations on teams that preserve the truth of their formulae. For instance, an important property of dependence logic is *downwards closure*: whenever a formula of FO(dep) is true for a team X, then it is also true for all subteams $Y \subseteq X$. Inclusion logic, on the other side, is not downwards closed, but closed under arbitrary unions of teams: If a formula of FO($\subseteq$) is true for each team in a finite or infinite collection $\{X_i : i \in I\}$, then the formula also holds for its union $\bigcup_{i\in I} X_i$. First-order logic (without dependency atoms) is both downwards closed and closed under union of teams, and thus has the *flatness property*: $\mathfrak{A} \models_X \psi$ if, and only if, $\mathfrak{A} \models_{\{s\}} \psi$ for all $s \in X$. Finally, independence logic is neither closed under unions of teams, nor downwards closed. For the counting constructs that we propose, we shall investigate whether or not they preserve such closure properties.

We will now introduce two methods for counting in team semantics that proved most fruitful during our work. After giving a formal definition, we will ensure that they fulfill the locality property and state their closure properties. Then, we will examine their expressive power in relation to other logics with team semantics, and fragments of Σ^1_1.

3 Forking

The *forking degree*, for two variable tuples $\overline{v}$ and $\overline{w}$, is defined as the number of different values assigned to $\overline{w}$ by all assignments with coinciding values for $\overline{v}$. A forking degree of one is equivalent to functional dependence so we can consider forking as a generalization of dependence. Note that this definition results in a separate forking degree for each group of assignments with the same value of $\overline{v}$. We introduce a counting mechanism based on *forking atoms* which, just as counting quantifiers, relate the counting result to a numeric variable.

▶ **Definition 2.** Forking atoms have the form $\overline{v}\lhd^{\leq\mu}\overline{w}$ and $\overline{v}\lhd^{\geq\mu}\overline{w}$, with the following semantics:

$$\mathfrak{A} \models_X \overline{v}\lhd^{\leq\mu}\overline{w} \iff \text{for all } s \in X, |\{s'(\overline{w}) : s' \in X,\ s'(\overline{v}) = s(\overline{v})\}| \leq s(\mu).$$

The definition for $\overline{v}\lhd^{\geq\mu}\overline{w}$ is analogous. When we extend a logic with both kinds of forking atoms, we can use the atom $\overline{v}\lhd^{=\mu}\overline{w}$ instead. Indeed, $\overline{v}\lhd^{=\mu}\overline{w} \equiv \overline{v}\lhd^{\leq\mu}\overline{w} \wedge \overline{v}\lhd^{\geq\mu}\overline{w}$ and, conversely, together with the ordering relation over numerals, we can simulate the two variants with $\overline{v}\lhd^{=\mu}\overline{w}$. We denote the extensions of FO by forking atoms of these forms by $\mathrm{FO}(\mathrm{fork}^{\leq})$, $\mathrm{FO}(\mathrm{fork}^{\geq})$, and $\mathrm{FO}(\mathrm{fork}^{=})$. Since forking atoms only depend on values of explicitly mentioned variables, they trivally have the locality property, and this is preserved by all first-order operations.

▶ **Proposition 3.** *All extensions of* FO *by forking atoms satisfy the locality property.*

Whenever functional dependence is accessible in a logic, it is possible to define a bijection between a tuple $\overline{w}$ and a fresh numeric variable, so that we can reduce the forking atoms to a simpler form, with a single variable w on the right side rather than a tuple. We write $\mathrm{FO}(\mathrm{fork}_1^{\leq})$ for this restriction of $\mathrm{FO}(\mathrm{fork}^{\leq})$, and similarly for $\geq$ and $=$. Particularly, since a forking degree of one is equivalent to functional dependence, we can easily express dependence with the forking atom $\overline{v}\lhd^{\leq\mu}w$ and hence $\mathrm{FO}(\mathrm{fork}^{\leq}) \equiv \mathrm{FO}(\mathrm{fork}_1^{\leq})$.

▶ **Example 4.** Regularity of graphs is definable in $\mathrm{FO}(\mathrm{fork}_1^{=})$, and even in $\mathrm{FO}(\mathrm{fork}_1^{\geq})$, by $\exists\mu\exists\nu(\mu+\nu = e \wedge \forall x\forall y((Exy \wedge x\lhd^{\geq\mu}y) \vee (\neg Exy \wedge x\lhd^{\geq\nu}y)))$. Recall the e is a numeric constant for the cardinality of the point sort.

Analogously to restricting dependence to constancy logic, there also is the special case of constant forking (abbreviated cfork), where the tuple $\overline{v}$ is of arity zero. The *constant forking atoms* $\lhd^{\leq\mu}\overline{w}$ and $\lhd^{\geq\mu}\overline{w}$ compare the number of different values of $\overline{w}$ with the value of μ.

▶ **Example 5** (Majority via constant forking)**.** One of the three possibilities, described in the introduction, for defining the team semantics of the majority quantifier was that $\mathfrak{A} \models_X My\varphi(\overline{v}, y)$ if $\mathfrak{A} \models_{X[y\mapsto F]} \varphi$ for some function $F : X \to \mathcal{P}(A)$ with $|\bigcup_{s\in X} F(s)| \geq |A|/2$. This can equivalently be defined in $\mathrm{FO}(\mathrm{cfork}^{\geq})$ by the formula

$$\overline{v} = \overline{v} \vee (\exists\mu(\mu+\mu \geq e \wedge \exists y(\varphi(\overline{v}, y) \wedge \lhd^{\geq\mu}y)).$$

To explain this, notice that the function $F : X \to \mathcal{P}(A)$ defines a split $X = Y \cup Z$ where $Y = \{s \in X : F(s) = \emptyset\}$ and $Z = \{s \in X : F(s) \neq \emptyset\}$, so that $X[y \mapsto F] = Z[y \mapsto F]$. Hence $\mathfrak{A} \models_{X[y\mapsto F]} \varphi$ means that $\mathfrak{A} \models_{Z[y\mapsto F]} \varphi$, and hence $\mathfrak{A} \models_Z \exists y\varphi$. Further $|\bigcup_{s\in X} F(s)| = |\bigcup_{s\in Z} F(s)|$. The displayed formula is a disjunction, and thus also imposes a split $X = Y \cup Z$, without any further restriction on Y (hence the disjunct $\overline{v} = \overline{v}$), and with the appropriate condition on Z. Notice the forking atom $\lhd^{\geq\mu}y$ implies in particular that Z is not empty.

One can also easily express in $\mathrm{FO}(\mathrm{fork}_1^{=})$ that two finite equivalence relations are isomorphic by saying that they have the same number of equivalence classes of any given size. Thus forking adds quite some expressiveness to first-order logic. It is easy to see, on the other side, that forking does not take us out of existential second-order logic (and NP). To examine the expressive power of forking more precisely it is useful to study the closure properties of forking logics.

▶ **Proposition 6.** *Forking atoms $\overline{v}\lhd^{\leq\mu}\overline{w}$ are downwards closed, but not under unions of teams, whereas atoms $\overline{v}\lhd^{\geq\mu}\overline{w}$ are closed under unions of teams but not downwards. Hence atoms $\overline{v}\lhd^{=\mu}\overline{w}$ are neither downwards closed nor closed under unions.*

As a direct consequence of the closure properties we obtain insights into the relationship between logics with forking and logics based on dependence atoms. Indeed, since dependence logic FO(dep) corresponds exactly to the downwards closed fragment of existential second-order logic [12] it must also contain $\mathrm{FO}(\mathrm{fork}^{\leq})$. By a more direct argument, we can, for two-sorted structures, directly translate a forking atom $\overline{v}\lhd^{\leq\mu}\overline{w}$ into $\exists\lambda(\lambda < \mu \wedge =\!(\overline{v}\lambda, \overline{w}))$. The idea of this formula is to extend the variables $\overline{v}$ with at most μ additional degrees of freedom and then demand a functional dependency between the extended tuple $\overline{v}\lambda$ and $\overline{w}$.

▶ **Theorem 7.** FO(dep) *is equivalent to* $\mathrm{FO}(\mathrm{fork}^{\leq})$*, and incomparable to* $\mathrm{FO}(\mathrm{fork}^{\geq})$*. Finally,* $\mathrm{FO}(\mathrm{fork}^{=})$ *is strictly stronger than* FO(dep)*.*

$\mathrm{FO}(\mathrm{fork}^{\geq})$ is incomparable with FO(dep), having the same closure properties as inclusion logic. However, $\mathrm{FO}(\mathrm{fork}^{\geq})$ is not equivalent with inclusion logic. For instance, by a simple Ehrenfeucht-Fraïssé argument one can show that regularity of graphs is not expressible in LFP, and hence neither in inclusion logic, but we have seen above that it is definable in $\mathrm{FO}(\mathrm{fork}^{\geq})$. We conjecture that $\mathrm{FO}(\mathrm{fork}^{\geq})$ and $\mathrm{FO}(\subseteq)$ are incomparable.

It remains to determine the expressive power of the strongest forking logic $\mathrm{FO}(\mathrm{fork}^{=})$. It turns out that, even when restricted to constant forking, it has the full power of independence logic, and thus of existential second-order logic and NP.

▶ **Theorem 8.** *On two-sorted structures,* $\mathrm{FO}(\mathrm{cfork}^{=}) \equiv \mathrm{FO}(\mathrm{fork}^{=}) \equiv \mathrm{FO}(\mathrm{indep})$.

Proof. Since FO(indep) has the full power of Σ^1_1, we just have to prove that any independence atom is equivalent to a formula in $\mathrm{FO}(\mathrm{cfork}^{=})$. We claim that

$$\overline{v}\bot\overline{w} \equiv \exists\mu\exists\nu(\lhd^{=\mu}\overline{v} \wedge \lhd^{=\nu}\overline{w}) \wedge \lhd^{\mu\nu}\overline{v}\overline{w}.$$

Indeed, it is not difficult to verify that the two formulae are just two different ways to express that $X[\overline{vw}] = X[\overline{v}] \times X[\overline{w}]$. There is an alternative way to express independence by non-constant forking, but with just one counting variable. Indeed,

$$\overline{v}\bot\overline{w} \equiv \exists\mu(\lhd^{=\mu}\overline{w} \wedge \overline{v}\lhd^{=\mu}\overline{w}).$$

For any team X, let $Y := X[\mu \mapsto m]$ for $m = |X[\overline{w}]|$. Assume that $\models_X \overline{v}\bot\overline{w}$. Clearly, $\models_Y \lhd^{=\mu}\overline{w}$. To prove that also $\overline{v}\lhd^{=\mu}\overline{w}$ holds in Y we have to show that any value $\overline{a} \in Y[\overline{v}] = X[\overline{v}]$ forks to *all* values $\overline{b} \in Y[\overline{w}] = X[\overline{w}]$. Fix $s, s' \in X$ with $s(\overline{v}) = \overline{a}$ and $s'(\overline{w}) = \overline{b}$. By $\overline{v}\bot\overline{w}$ there exists an assignment $s'' \in X$ with $s''(\overline{v}) = \overline{a}$ and $s''(\overline{w}) = \overline{b}$. Thus $t'' = s''[\mu \mapsto m] \in Y$ witnesses the forking of $\overline{a}$ to $\overline{b}$.

Conversely, assume that $\models_X \exists\mu(\lhd^{=\mu}\overline{w} \wedge \overline{v}\lhd^{=\mu}\overline{w})$, which implies that $\models_Y \overline{v}\lhd^{=\mu}\overline{w}$. Choose two assignment $s, s' \in X$. Since $s(\overline{v})$ forks in Y to all values in $Y[\overline{w}]$, it forks in particular to $s'(\overline{w})$, so there exists an assignment $t'' \in Y$ with $t''(\overline{v}) = s(\overline{v})$ and $t''(\overline{w}) = s'(\overline{w})$. The restriction of t'' to the domain of X thus witnesses the truth of $\overline{v}\bot\overline{w}$. ◀

Forking atoms thus provide a lot of power. Compared to the familiar logics with team semantics such as dependence and independence logic, which are able to express NP-complete problems but usually in a somewhat roundabout way that is rather hard to find and to read, forking atoms often lead to more direct and natural definitions. Further, the two-sorted framework makes it much easier to deal with problems that include numeric parameters, such as bounds on the size of solutions or structures with weights. To illustrate this, we consider some familiar NP-complete problems.

▶ **Example 9** (Dominating Set, Vertex Cover, and Clique)**.** The dominating set problem can be expressed in $\mathrm{FO}(\mathrm{cfork}^{\leq})$. Indeed, a graph $G = (V, E)$ admits a dominating set of size k if, and only if, $(G, k) \models \forall y\exists x(Exy \wedge \lhd^{\leq k}x)$. A similar idea works for Vertex Cover and Clique. A graph G has a vertex cover of size $\leq k$ if $(G, k) \models \forall x\forall y(\neg Exy \vee \exists z((z = x \vee z = y) \wedge \lhd^{\leq k}z)$, and it has a clique of size $\geq k$ if $(G, k) \models \forall x\forall y(Exy \vee \exists z((z = x \vee z = y) \wedge \lhd^{\leq e-k}z)$. These examples also separate $\mathrm{FO}(\mathrm{cfork}^{\leq})$ from FO(const), since for sentences the latter collapses to FO.

▶ **Example 10** (TSP). A more ambitious challenge for the use of forking atoms is a presentation of the TSP. An instance D of the TSP, with distances $d_{ij} \in \mathbb{N}$, for $i, j \in \{0, \dots, n-1\}$ with $i \neq j$, can be represented as a team $X(D)$ consisting of the $n(n-1)$ assignments $s_{ij} : (x, y, \lambda) \mapsto (i, j, d_{ij})$ into A^*, for $A = \{a_0, \dots, a_{n-1}\}$. We construct a formula $\psi(x, y, \lambda, k) \in \mathrm{FO}(\mathrm{fork}^{\leq})$ such that, for all such A, $X(D)$ and all $k \in \omega$,

$$(A, k) \models_{X(D)} \psi \iff D \text{ admits a TSP-tour of length } \leq k.$$

We first notice that for every formula φ such that $(A, k) \models_{X(D)} x \lhd^{\leq e-2} y \vee (y \lhd^{\leq 1} x \wedge \varphi)$ it follows that φ must hold in some subteam $Z \subseteq X(D)$ that covers A by a disjoint union of cycles. More precisely, this means that the directed graph (A, E_Z) with $E_Z = \{(a_i, a_j) : d_{ij} \in Z\}$ consists of disjoint cycles, and every a_i occurs in precisely one of these cycles. Indeed the formula requires a split $X(D) = Y \cup Z$, such that Y contains, for every i, at most $n-2$ assignment s_{ij}. Thus, for each i there remains at least one assignment s_{ij} that must be in Z, but on the other side, by the forking atom $y \lhd^{\leq 1} x$ at most one s_{ij} can be in Z for each j. The only way to satisfy these contraints is that E_Z defines a bijection (and thus a covering by disjoint cycles). It remains to construct φ so that it enforces, for such a subteam Z, that it in fact consists of just one cycle, and that the length of this cycle does not exceed k. This is achieved by

$$\begin{aligned}\varphi := \exists c(&\lhd^{\leq 1} c \wedge \exists \mu \exists \nu \forall x' \exists \mu' \exists \nu' (x \lhd^{\leq 1} \mu \wedge x' \lhd^{\leq 1} \mu' \wedge x \lhd^{\leq 1} \nu \wedge x' \lhd^{\leq 1} \nu' \wedge \\ &(x = x' \to \mu = \mu' \wedge \nu = \nu') \wedge (x = c \to \mu = 0 \wedge \nu = 1) \wedge \\ &(x' = y \neq c \to \mu' = \mu + \lambda \wedge \nu' = \nu + 1) \wedge (x' = y = c \to \mu + \lambda \leq k \wedge \nu = e)))\end{aligned}$$

Recall that in the team Z we have for every i precisely one assigment $s_{ij} \in Z$. The quantifiers and the first two lines of the formula thus imply that there is a node c and functions $a_i \mapsto \mu_i$ and $a_i \mapsto \nu_i$ assigning to each node two numbers. Viewing c as the beginning of the tour, and a_i as a node on the same cycle as c, it follows by induction on the path from c to a_i that φ imposes that the value of μ_i is the length of the path from c to a_i, and that the value of ν_i is the number of nodes on that path. Finally for the closing of the cycle at c, the formula says that the cycle has length $\leq k$ and contains all nodes. Thus, the subteam Z must indeed be a TSP tour of length at most k.

Since constant forking is a restriction of general forking, it inherits all locality and closure properties. Furthermore, also the relationship between functional dependence and forking can be readily translated to constancy logic and constant forking. However, in contrast to the equivalence between FO(dep) and $\mathrm{FO}(\mathrm{fork}^{\leq})$ it is possible to separate the constant variant $\mathrm{FO}(\mathrm{cfork}^{\leq})$ from FO(const). See Figure 1, at the end of this paper, for an illustration and summary of such results.

4 Counting Quantifiers

Counting quantifiers of form $\exists^{\geq \mu} x$ provide a well-known and powerful way to add counting to logics with Tarski semantics, such as first-order logic or fixed-point logic. We adapt them to team semantics, and investigate the properties and expressive power of the resulting extensions. In fact, we propose a rather general variant of such quantifiers which admits not only the counting of single elements (over the point sort) but counting of arbitrary tuples, even of mixed type.

▶ **Definition 11.** *Counting quantifiers* for teams permit to build, for every formula φ, every numeric variable μ, every closed numeric term t, and every tuple $\overline{v}$ of variables of mixed type, the new formula $\exists^{\geq\mu}\overline{v}_{<t}\varphi$. For any structure $\mathfrak{A}^*$ and every team X whose domain includes μ and all variables in $\mathrm{free}(\varphi) \setminus \{\overline{v}\}$, we have that $\mathfrak{A}^* \models_X \exists^{\geq\mu}\overline{v}_{<t}\varphi$ if, and only if, there is a function F that maps every $s \in X$ to a set $F(s)$ of at least $s(\mu)$ many, appropriately typed, tuples over $A \cup \omega$, whose numeric components are bounded by t, such that $\mathfrak{A}^* \models_{X[\overline{v}\mapsto F]} \varphi$.

▶ **Example 12** (Majority via counting quantifiers). If we define the team semantics of the majority quantifier so that $\mathfrak{A} \models_X My\varphi(\overline{v}, y)$ if $\mathfrak{A} \models_{X[y\mapsto F]} \varphi$ for a function $F : X \mapsto \mathcal{P}(A)$ with $|F(s)| \geq |A|/2$ for all $s \in X$, then this is equivalent to $\exists\mu(\mu + \mu \geq e \wedge \exists^{\geq\mu}y\varphi(\overline{v}, y))$. If we require instead that $F(s) = B$ for some fixed set B with $|B| \geq |A|/2$, then we need a combination of counting quantifiers and forking atoms, namely $\exists\mu(\mu+\mu \geq e \wedge \exists^{\geq\mu}y(\varphi(\overline{v}, y) \wedge \lhd^{\leq\mu}y))$.

▶ **Proposition 13.** *Counting quantifiers preserve the locality property.*

Notice instead that counting quantifiers of form $\exists^{\leq\mu}y$ or $\exists^{=\mu}y$ that are in common use in logics with Tarski semantics, are unsafe for team semantics since they may violate the locality principle. Indeed, even the quantifiers $\exists^{\leq 1}y$ or $\exists^{=1}y$ can be used to simulate the strict semantics for common existential quantifiers, and we have already seen that this is in conflict with the locality principle. Observe that, in contrast to forking, counting quantifiers do not express any dependency between assignments.

▶ **Proposition 14.** *Counting quantifiers preserve downwards closure and closure under unions. As a consequence,* FO *with counting quantifiers, but without any dependence atoms, is flat.*

Nevertheless, FO with counting is more expressive than without counting. It can express for instance even cardinality or regularity of graphs. We already mentioned in the introduction that counting is definable in logics whose expressive power comes sufficiently close to Σ^1_1. In particular, this is the case for dependence logic (but not, for instance, for inclusion logic). In fact dependence statements can be seen as a particular form of weak counting, and we show that these suffice, over two-sorted structures, to express counting quantifiers in a rather simple way.

▶ **Proposition 15.** *Counting quantifiers are expressible by means of dependence atoms.*

Proof. A formula $\psi := \exists^{\geq\mu}y\varphi$ is equivalent to $\forall\lambda(\lambda \geq \mu \vee \exists y(=(\mathrm{free}(\psi)y, \lambda) \wedge \varphi))$. This construction readily extends to more general counting quantifiers. ◀

5 Inclusion Logic with Counting Quantifiers

We now study the logic $\mathrm{FO}(\subseteq, \exists^{\geq\mu})$ that extends inclusion logic with counting quantifiers, and show that it is equivalent, in the sense described in the introduction, with fixed-point logic with counting (FPC). Recall that for all two-sorted structures $\mathfrak{A}^*$ we always require the point sort $\mathfrak{A}$ to be a finite structure.

5.1 Fixed-point logic with counting with only greatest fixed-points

As already mentioned, FPC is usually defined as the extension of first-order logic over two-sorted structures by counting terms and inflationary fixed-points. However, this definition is not really adequate for proving an equivalence with a logic with team semantics, for two reasons: Counting terms may violate the locality principle, and inflationary fixed-points have no direct translation into logics with team semantics (or into Σ_1^1, for that matter)[3]. We therefore work with a different definition of FPC, that is based on greatest fixed-point operators, used only positively, and on counting quantifiers of form $\exists^{\geq\mu}\overline{v}$. Of course, we have to convince ourselves that this syntactically restricted variant is semantically equivalent, i.e. that we do not lose expressive power. While this is not trivial, it can be proved by combining some well-understood techniques.

First of all, counting terms $\#_x\varphi(x)$ can readily be replaced by formulae with counting quantifiers of the form $\exists^{=\mu}x\varphi$, and these can be rewritten by $\exists^{\geq\mu}x\varphi(x) \wedge \exists\nu(\mu + \nu = e \wedge \exists^{\geq\nu}x\neg\varphi(x))$. However, this introduces negation, which poses a problem with monotonicity, and this is one of the reasons why one normally prefers inflationary fixed-points rather than least and greatest ones.

In the absence of counting it is known that, by means of the Stage Comparison Theorem, one can eliminate inflationary fixed points and prove that the logics LFP and IFP are semantically equivalent [8, 13]. On finite structures, one can even go an important step further and prove (again based on the Stage Comparison Theorem) that the negation of a greatest fixed point is equivalent to a formula using greatest fixed points only positively [11].

It is straightforward (but a bit lengthy) to verify that the proof of the Stage Comparison Theorem, and also its applications, go through for the case of two-sorted structures and in the presence of counting quantifiers. Thus we can indeed, without loss of generality, assume that all formulae in FPC are written in this restricted form. In particular, this has the advantage that we have a relatively simple description of model-checking games for FPC, as so-called threshold safety games.

5.2 Threshold games

A threshold game is a two-player game on a finite (or at least finitely branching) directed graph $G = (V, E)$ equipped with a threshold function $\theta : V \to \omega$. Let $vE = \{w : (v, w) \in E\}$ and let $\delta(v) := |vE|$ denote the out-degree of v. We assume that $\theta(v) \leq \delta(v) + 1$ for all v.

At any given node v in a play, Player 0 selects a set $X \subseteq vE$ of $\theta(v)$ successors of v, then Player 1 choses a node $w \in X$ and the play proceeds from w. When a player cannot move, she loses. This means that Player 0 wins at all nodes in $T_0 := \{v \in V : \theta(v) = 0\}$, and Player 1 wins at nodes in $T_1 := \{v \in V : \delta(v) < \theta(v)\}$.

Classical (finitely branching) graph games, where the set of nodes is partitioned into two sets, $V = V_0 \cup V_1$, such that Player 0 moves from nodes in V_0 and Player 1 from those in V_1, can be viewed as the special case of threshold games where, for all nodes v, either $\theta(v) = 1$ or $\theta(v) = \delta(v)$. In principle, we can combine threshold games with any winning condition for infinite plays, but in this paper we just consider *threshold safety games*, where Player 0 just has to avoid the positions in T_1 where she loses immediately. In particular, Player 0 wins all infinite plays.

The winning region of a player is the set of positions from which she has a winning strategy. The well-known linear-time algorithm for computing winning regions in classical reachability and safety games can be adapted to threshold games.

▶ **Proposition 16.** *The winning regions of a threshold safety game $\mathcal{G} = (V, E, \theta : V \to \omega)$ on a finite game graph can be computed in time $O(|V| + |E|)$.*

---

3 On finite structures we have an indirect translation into Σ_1^1 since fixed-point logics are in polynomial time and Σ_1^1 captures NP, but on infinite structures, LFP and IFP are on the Δ_2^1-level.

Even more relevant for us is the relationship to logics with counting. We first note that the winning region W_0 for Player 0 in threshold safety games $\mathcal{G} = (V, E, \theta : V \to \omega)$ is uniformly definable by a very simple formula in fixed-point logic with counting FPC, namely $\text{win}(x) := [\mathbf{gfp}\, Wx\,.\,\exists^{\geq \theta(x)} y(Exy \wedge Wy)](x)$. Here and in the following, $\exists^{\geq \theta(x)} y\varphi(y)$ is just an abbreviation for $\exists \mu(\mu = \theta(x) \wedge \exists^{\geq \mu} y\varphi)$. Equivalently, the winning region is definable in inclusion logic with counting, by $\text{win}'(x) := \exists^{\geq \theta(x)} y(Exy \wedge y \subseteq x)$. Then for any threshold game graph $\mathcal{G}$, the maximal team W such that $\mathcal{G} \models_W \text{win}'(x)$ is precisely the winning region for Player 0.

For the translation between the two logics, we shall use a specific kind of trap condition for initial positions.

▶ **Definition 17.** Fix a set $I \subseteq V$ of initial positions in a threshold safety game $\mathcal{G}$. An I-*trap* in $\mathcal{G}$ is a set $Z \subseteq I$ such that Player 0 has a winning strategy from Z that, moreover, avoids $I \setminus Z$.

Such a winning condition can simply be described by a subset $W \subseteq V$ such that $W \cap I = Z$, and $|vE \cap W| \geq \theta(v)$ for all $v \in W$. Indeed, at any position $v \in W$, Player 0 can then select any set $X \subseteq vE \cap W$ with $|X| = \theta(v)$ and wins since $W \cap (I \setminus Z) = \emptyset$ and $W \cap T_1 = \emptyset$. A straightforward modification of the formulae for winning regions shows that in both logics, FPC and $\text{FO}(\subseteq, \exists^{\geq \mu})$, we can define also I-traps.

But the connection between threshold games and logics with counting goes much deeper. Threshold safety games arise as the model-checking games for both inclusion logic with counting and FPC, and moreover, the model-checking games are uniformly interpretable in the structure in which the formulae are evaluated.

5.3 Model-checking games and game interpretations

It is known, and explained in detail in [6], how to construct classical safety games as evaluation games for inclusion logic and for the posGFP fragment of fixed-point logic. We extend these constructions to obtain threshold safety games as model-checking games, on finite structures, for inclusion logic with counting and for FPC.

We first sketch the construction of a threshold safety game $\mathcal{T}(\mathfrak{A}^*, \psi)$ for a two-sorted structure $\mathfrak{A}^*$ and a formula $\psi(\overline{x}, \overline{\mu})$ of fixed-point logic with counting (which uses only greatest fixed points). Let $G(\psi) = (\text{Sf}(\psi), E_\psi)$ be the syntax graph of ψ. Positions of the game are pairs (φ, s) where $\varphi \in \text{Sf}(\psi)$ is a subformula of ψ and $s : \text{free}(\varphi) \to (A \cup \omega)$ is an (appropriately typed) assignment on the free variables of φ. The immediate successors of a position (φ, s), are the pairs (φ', s') where $(\varphi, \varphi') \in E_\psi$ and s and s' coincide on the common variables. A position $([\mathbf{gfp}\, Z\overline{x}\overline{\mu}_{<t}\,.\,\eta(Z, \overline{x}, \overline{\mu})](\overline{x}, \overline{\mu}), s)$ has the unique successor (η, s). For any fixed-point atom $Z\overline{y}\overline{\nu}$ in the scope of η, the unique successor of a position $(Z\overline{y}\overline{\nu}, s)$ is (η, s') with $s'(\overline{x}, \overline{\mu}) = s(\overline{y}, \overline{\nu})$. Thresholds $\theta(\varphi, s)$ are assigned as follows:

1. In the case that φ is a first-order literal, we set $\theta(\varphi, s) := 0$ if $\mathfrak{A} \models_s \varphi$ and $\theta(\varphi, s) := 1$ if $\mathfrak{A} \not\models_s \varphi$.
2. We set $\theta(\varphi, s) := 1$ in all cases where φ is either a fixed-point atom $Z\overline{x}\overline{\mu}$, a fixed-point formula $[\mathbf{gfp}\, Z\overline{x}\overline{\mu}_{<t}\,.\,\eta(Z, \overline{x}, \overline{\mu})](\overline{x}, \overline{\mu})$, a disjunction $\varphi_1 \vee \varphi_2$, or an existentially quantified formula $(\exists \mu < t)\varphi'$.
3. We set $\theta(\varphi, s) := 2$ if φ is a conjunction $\varphi_1 \wedge \varphi_2$.
4. For formulae $\varphi := (\forall \mu < t)\varphi'$, we set $\theta(\varphi, s) = t^{\mathfrak{A}^*}$, i.e. the value of the numeric term t in $\mathfrak{A}^*$
5. For formulae with counting quantifiers $\varphi := \exists^{\geq \mu} \overline{v} \varphi'$, we put $\theta(\varphi, s) = s(\mu)$.

▶ **Theorem 18.** *For every structure $\mathfrak{A}^*$, every formula $\psi(\overline{x},\overline{\mu}) \in$ FPC, and every appropriately typed assignment $s : \text{free}(\varphi) \to (A \cup \omega)$, we have that $\mathfrak{A}^* \models_s \psi(\overline{x},\overline{\mu})$ if, and only if, Player 0 has a winning strategy for the threshold safety game $\mathcal{T}(\mathfrak{A}^*,\psi)$ from position (ψ, s).*

For the relationship with inclusion logic we shall need a more refined result, concerning sentences in FPC of vocabulary $\tau \cup \{X\}$ of the form $\psi := \forall\overline{x}\forall\overline{\mu}(X\overline{x}\overline{\mu} \to \varphi(\overline{x},\overline{\mu}))$, such that X occurs only positively in φ. In that case, the model checking game $\mathcal{T}((\mathfrak{A}^*, X), \psi)$ is a threshold safety game with unique initial position $(\psi, \emptyset)$. We modify these games by eliminating the explicit reference to the relation X and associate the model checking problem of whether $(\mathfrak{A}, X) \models \psi$ with a trap condition for a modified game $\mathcal{T}^{\#}(\mathfrak{A}^*,\varphi)$. To do this, we identify every position of form $(X\overline{y}\overline{\nu}, t)$ with the position $(\varphi(\overline{x},\overline{\mu}), s)$ such that $s(\overline{x},\overline{\mu}) = t(\overline{y},\overline{\nu})$; this means that every edge in the game graph to a position $(X\overline{y}\overline{\nu}, t)$ is replaced by an edge to $(\varphi(\overline{x},\overline{\mu}), s)$, and the node $(X\overline{y}\overline{\nu}, t)$ is deleted. The set I of initial positions now consists of all pairs of form $(\varphi(\overline{x},\overline{\mu}), s)$. Given any interpretation for the relation X, let $X^* \subseteq I$ be the set of positions (φ, s) where $s(\overline{x},\overline{\mu}) \in X$.

▶ **Proposition 19.** $(\mathfrak{A}^*, X) \models \forall\overline{x}\forall\overline{\mu}(X\overline{x}\overline{\mu} \to \varphi(\overline{x},\overline{\mu}))$ *if, and only if, X^* is an I-trap in $\mathcal{T}^{\#}(\mathfrak{A}^*,\varphi)$.*

The construction for inclusion logic with counting is similar. It extends the construction given in [6] of safety games for FO($\subseteq$). With every formula $\psi(\overline{x},\overline{\mu})$ in FO($\subseteq, \exists^{\geq\mu}$) and every structure $\mathfrak{A}^*$, we construct a threshold safety game $\mathcal{T}(\mathfrak{A}^*,\psi)$, played on a forest of game trees, where again, positions are pairs (φ, s) where φ is an occurrence of a subformula in ψ and s an assignment with domain free(φ). The set I of initial positions is the set of roots of the game trees in $\mathcal{T}(\mathfrak{A}^*,\psi)$; it contains all pairs (ψ, s). Hence, a team X for ψ defines the subset $I(X) := \{(\psi, s) : s \in X\}$ of I. Instead of the regeneration of fixed-points we here have a regeneration mechanism for inclusion atoms. The threshold game graph is set up so that, informally, at a position $((\overline{x}\overline{\mu} \subseteq \overline{y}\overline{\nu}), s)$ associated with an inclusion atom, Player 0 selects an assignment t such that $t(\overline{y},\overline{\nu}) = s(\overline{x},\overline{\mu})$, moves to $((\overline{x}\overline{\mu} \subseteq \overline{y}\overline{\nu}), t)$ and then Player 1 takes the play to an ancestor of that node in the corresponding game tree. (A more formal construction requires that we duplicate all nodes in the game, so that the moves going upwards in the game tree actually take place in a separate copy of the tree. For details of this, in the context of classical safety games for inclusion logic, see [6]). To win, Player 0 has to make sure that the play remains inside those trees with a root in $I(X)$, which means that these roots form an I-trap.

▶ **Theorem 20.** *For every formula $\psi(\overline{x},\overline{\mu}) \in$ FO($\subseteq, \exists^{\geq\mu}$), every structure $\mathfrak{A}^*$ and every team X, we have that $\mathfrak{A}^* \models_X \psi(\overline{x},\overline{\mu})$ if, and only if, $I(X)$ is an I-trap in $\mathcal{T}(\mathfrak{A}^*,\psi)$.*

Proof. Suppose that $\mathfrak{A}^* \models_X \psi(\overline{x},\overline{\mu})$. Then, according to the rules defining the semantics of ψ, we can assign to every occurrence of a subformula φ in ψ a team $Y(\varphi)$ such that $Y(\psi) = X$ and $\mathfrak{A}^* \models_{Y(\varphi)} \varphi$ for all φ. In particular, for $\varphi = \varphi_1 \vee \varphi_2$ we have that $Y(\varphi) = Y(\varphi_1) \cup Y(\varphi_2)$, for $\varphi = \exists^{\geq\mu}\overline{v}\eta$ we have that $Y(\varphi) = Y(\eta)[\overline{v} \mapsto F]$ for a function $F : Y(\eta) \to \mathcal{P}((A^*)^k)$ with $|F(s)| \geq s(\mu)$, and so on for the other types of formulae. We define a strategy W for Player 0 in $\mathcal{T}(\mathfrak{A}^*,\psi)$ by

$$W := \{(\varphi, s) : \varphi \in \text{Sf}(\psi), s \in Y(\varphi)\}.$$

It is straightforward to verify that for any position $(\varphi, s) \in W$, there are at least $\theta(\varphi, s)$ many successors of (φ, s) inside W, and if $(\varphi, s) \notin W$, then none of the nodes in the subtree rooted at (φ, s) belongs to W. In particular, for any inclusion atom $\alpha := (\overline{x}\overline{\mu} \subseteq \overline{y}\overline{\nu})$ and any position of form $(\alpha, s) \in W$ we infer that, since α is true in $Y(\alpha)$ there exist an assignment $t \in Y(\alpha)$ with $t(\overline{y}, \overline{\nu}) = s(\overline{x}, \overline{\mu})$, so Player 0 can indeed take the game to an occurrence of (α, t) in W. Further, since W only contains nodes in trees with a root (ψ, s) such that $s \in X$, Player 1 can force the game, by going upwards the game trees, only to initial positions in $I(X)$. Thus $I(X)$ is indeed an I-trap in $\mathcal{T}(\mathfrak{A}^*, \psi)$.

Conversely, suppose that W describes a strategy for Player 0 showing that $I(X)$ is an I-trap in $\mathcal{T}(\mathfrak{A}^*, \psi)$. For every node (φ, s) in $\mathcal{T}(\mathfrak{A}^*, \psi)$, let

$$\mathrm{Team}(W, \varphi) := \{s : (\varphi, s) \in W\}.$$

In particular, $\mathrm{Team}(W, \psi) = X$. By induction on the syntax of ψ, one easily verifies that for every $\varphi \in S(\psi)$,

$$\mathfrak{A}^* \models_{\mathrm{Team}(W,\varphi)} \varphi.$$

We just discuss the most interesting cases.

- If φ is a first-order literal, then all nodes (φ, s) are terminal nodes, so W can contain (φ, s) only if $\theta(\varphi, s) = 0$ which is the case if, and only if, $\mathfrak{A}^* \models_s \varphi$. Thus $\mathfrak{A}^* \models_{\mathrm{Team}(W,\varphi)} \varphi$.
- Let φ be an inclusion atom $(\overline{x\mu} \subseteq \overline{y\nu})$. For all s such that $(\varphi, s) \in W$, there exists an assignment t with $t(\overline{y}, \overline{\nu}) = s(\overline{x}, \overline{\mu})$ such that also $(\varphi, t) \in W$. But this means that $\mathrm{Team}(W, \varphi)$ satisfies $(\overline{x\mu} \subseteq \overline{y\nu})$.
- If $\varphi = \varphi_1 \vee \varphi_2$, then for every node $(\varphi, s) \in W$, at least one of its two successors must also belong to W. Let s_1 and s_2 be the restrictions of s to the free variables of φ_1 and φ_2, respectively, and let Y_i be the team of all assignments s with domain $\mathrm{free}(\varphi)$ such that $s_i \in \mathrm{Team}(W, \varphi_i)$. It follows that $\mathrm{Team}(W, \varphi) = Y_1 \cup Y_2$. By induction hypothesis, we have that $\mathfrak{A}^* \models_{\mathrm{Team}(W,\varphi_i)} \varphi_i$ for $i = 1, 2$ and, by the locality principle, also $\mathfrak{A}^* \models_{Y_i} \varphi_i$. It follows that $\mathfrak{A}^* \models_{\mathrm{Team}(W,\varphi)} \varphi$.
- For $\varphi = \exists^{\geq\mu}\overline{v}\eta$ we have that $\theta(\varphi, s) = s(\mu)$. Thus, if $(\varphi, s) \in W$ then there exist at least $s(\mu)$ many assignments $t = s[\overline{v} \mapsto \overline{c}]$ such that $(\eta, t) \in W$, and hence $t \in \mathrm{Team}(W, \eta)$. By induction hypothesis, $\mathfrak{A}^* \models_{\mathrm{Team}(W,\eta)} \eta$ which implies that there is a function $F : \mathrm{Team}(W, \varphi) \to \mathcal{P}(A^*)^k$ with $|F(s)| \geq s(\mu)$ for all $s \in \mathrm{Team}(W, \varphi)$ and $\mathfrak{A}^* \models_{\mathrm{Team}(W,\varphi)[\overline{v}\mapsto F]} \eta$. But this means that $\mathfrak{A}^* \models_{\mathrm{Team}(W,\varphi)} \varphi$.

The remaining cases are routine. Since $\mathrm{Team}(W, \psi) = X$ this implies that $\mathfrak{A}^* \models_X \psi$. ◀

Observe that, for any fixed formula $\psi(\overline{x}, \overline{\mu})$ in any of these logics, the construction of the threshold safety games is done in a very uniform way. This intuition is made precise by the notion of a *game interpretation*. In our case, such an interpretation is a quadruple $J = (\delta, e, \mathrm{in}, \vartheta)$ consisting of first-order formulae $\delta(\lambda, \overline{v})$, $e(\lambda, \overline{v}; \lambda', \overline{v}')$, and $\mathrm{in}(\lambda, \overline{v})$ for the nodes, edges, and initial positions of the game graph, and a formula $\vartheta(\lambda, \overline{v}, \mu)$ for the thresholds. Given a structure $\mathfrak{A}^*$, we obtain a graph whose set of nodes is $\delta^{\mathfrak{A}^*} := \{(i, \overline{c}) : \mathfrak{A}^* \models \delta(i, \overline{c})\}$ and whose sets of edges $e^{\mathfrak{A}^*}$ and initial positions $\mathrm{in}^{\mathfrak{A}^*}$ are defined in an analogous way.

We say that J interprets the threshold game $\mathcal{T} = (V, E, I, \theta : V \to \omega)$ in $\mathfrak{A}^*$, if there is an isomorphism $h : (\delta^{\mathfrak{A}^*}, e^{\mathfrak{A}^*}, \mathrm{in}^{\mathfrak{A}^*}) \to (V, E, I)$, such that thresholds are defined by $\vartheta(\lambda, \overline{v}, \mu)$ as follows: For every node $v \in V$ and every number $t \in \omega$ we have that $\mathfrak{A}^* \models \vartheta(h^{-1}(v), t)$ if, and only if, $\theta(v) = t$.

▶ **Theorem 21.** *For every formula $\psi(\overline{x}, \overline{\mu})$ in $\mathrm{FO}(\subseteq, \exists^{\geq\mu})$ there is a first-order interpretation $J(\psi)$ that interprets, for any structure $\mathfrak{A}^*$, the game $\mathcal{T}(\mathfrak{A}^*, \psi)$ in $\mathfrak{A}^*$.*

Proof. For simplicity, we assume that the vocabulary of ψ and $\mathfrak{A}$ contains a constant c. We enumerate the subformulae of ψ as $\varphi_0, \dots, \varphi_\ell$, with $\varphi_0 = \psi$, and assume that $\overline{v}$ is the tuple of all variables occurring in ψ. For every formula φ_i we extend every assignment $s : \text{free}(\varphi_i) \to A^*$ to an assignment s^* on all variables in $\overline{v}$ by setting $s^*(y) = c$ and $s^*(\nu) = 0$ for all variables y or ν that appear in $\overline{v}$ but not in free(φ_i). The interpretation $J(\psi)$ then represent a position (φ_i, s) of $\mathcal{T}(\mathfrak{A}^*, \psi)$ by the tuple $(i, s^*(\overline{v}))$ in A^*. With this representation, it is completely straightforward to construct quantifier-free formulae $\delta(\lambda, \overline{v})$, $e(\lambda, \overline{v}; \lambda', \overline{v}')$, in$(\lambda, \overline{v})$, and $\vartheta(\lambda, \overline{v}, \mu)$ with the required properties. ◀

Analogous statements hold for the games for FPC.

5.4 Game-based translations between $\text{FO}(\subseteq, \exists^{\geq\mu})$ and FPC

A first-order interpretation J may be seen as a function mapping a structure $\mathfrak{C}$ to the interpreted structure $J(\mathfrak{C})$, and by the coordinate map h of J, every element $b \in J(\mathfrak{C})$ is associated with a tuple $h^{-1}(b) \in \mathfrak{C}$. But in the other direction, J also gives a translation from formulae $\varphi(x_1, \dots, x_k)$ over $J(\mathfrak{C})$ to formulae $\varphi^J(\overline{x}_1, \dots, \overline{x}_k)$ over $\mathfrak{C}$ such that

$$J(\mathfrak{C}) \models \varphi(b_1, \dots, b_k) \Longleftrightarrow \mathfrak{C} \models \varphi^J(h^{-1}(b_1), \dots, h^{-1}(b_k)).$$

This is called the Interpretation Lemma, which is of course a general and well-known fact that holds (in appropriate form) for all kinds of interpretations, not just the particular form of game interpretations that we use here.

We are now ready to prove the main theorem of this paper. We shall combine the interpretation argument for threshold safety games and the definability of I-traps, by means of the Interpretation Lemma, to provide effective translations between the two logics.

▶ **Theorem 22.** *There exist effective translations in both directions between formulae $\psi(\overline{x}, \overline{\mu})$ in $\text{FO}(\subseteq, \exists^{\geq\mu})$, and formulae $\varphi(X, \overline{x}, \overline{\mu})$ in FPC (with only positive occurrences of X), such that, for every structure $\mathfrak{A}^*$ and every X,*

$$\mathfrak{A}^* \models_X \psi(\overline{x}, \overline{\mu}) \quad \Longleftrightarrow \quad (\mathfrak{A}^*, X) \models \forall\overline{x}\forall\overline{\mu}(X\overline{x}\overline{\mu} \to \varphi(X, \overline{x}, \overline{\mu})).$$

In particular, on every structure $\mathfrak{A}^*$, the maximal team satisfying ψ coincides with the greatest fixed-point of φ. For sentences, $\text{FO}(\subseteq, \exists^{\geq\mu})$ and FPC have the same expressive power.

Proof. We first describe how to translate $\psi(\overline{x}, \overline{\mu}) \in \text{FO}(\subseteq, \exists^{\geq\mu})$ into an appropriate formula $\varphi(X, \overline{x}, \overline{\mu})$ in FPC.

Take a formula in FPC that defines I-traps in threshold games, so that in particular

$$(\mathcal{T}(\mathfrak{A}^*, \psi), Z) \models \forall x(Zx \to \text{itrap}(Z, x)) \iff Z \text{ is an I-trap in } \mathcal{T}(\mathfrak{A}^*, \psi).$$

The interpretation $J(\psi)$ defines a copy of $\mathcal{T}(\mathfrak{A}^*, \psi)$ inside $\mathfrak{A}^*$ and maps formulae on the game back to formulae on $\mathfrak{A}^*$. Thus, $J(\psi) : \text{itrap}(Z, x) \mapsto \text{itrap}^*(Y, \overline{y}, \overline{\nu})$. If $Y \subseteq A^k \times \omega^\ell$ is the set of tuples in $\mathfrak{A}^*$ associated with $Z \subseteq I$, then $(\mathfrak{A}^*, Y) \models \forall\overline{y\nu}(Y\overline{y\nu} \to \text{itrap}^*(Y, \overline{y}, \overline{\nu}))$ if, and only if, Z is an I-trap in $\mathcal{T}(\mathfrak{A}^*, \psi)$.

The tuples describing positions (ψ, s) with $s \in X$ are first-order definable in $(\mathfrak{A}^*, X)$. We can thus massage itrap$^*(Y, \overline{y}, \overline{\nu})$ into a formula $\varphi(X, \overline{x}, \overline{\mu}) \in$ FPC such that

$$(\mathfrak{A}^*, X) \models \forall\overline{x\mu}(X\overline{x\mu} \to \varphi(X, \overline{x}, \overline{\mu})) \iff I(X) \text{ is an I-trap in } \mathcal{T}(\mathfrak{A}^*, \psi) \iff \mathfrak{A} \models_X \psi.$$

An analogous construction works for the translation of FPC into $\text{FO}(\subseteq, \exists^{\geq\mu})$, making use of the fact that I-traps are definable also in inclusion logic with counting. Let $\mathcal{T}^{\#}(\mathfrak{A}^*, \varphi)$

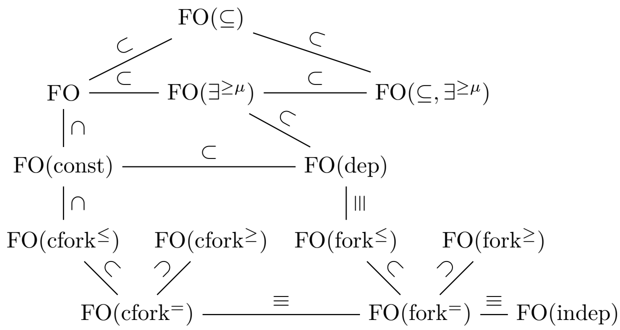

Figure 1 Overview of extensions of first-order logic by counting constructs over teams.

be the game from Proposition 19 such that $(\mathfrak{A}^*, X) \models \forall\overline{x}\forall\overline{\mu}(X\overline{x\mu} \to \varphi(\overline{x}, \overline{\mu}))$ if, and only if, X^* is an I-trap in $\mathcal{T}^\#(\mathfrak{A}, \varphi)$. Further, let $J(\varphi)$ be the interpretation, with coordinate map h, which, for every structure $\mathfrak{A}^*$, interprets $\mathcal{T}^\#(\mathfrak{A}^*, \varphi)$ in $\mathfrak{A}^*$.

Take now a formula $\text{itrap}(x)$ of $\text{FO}(\subseteq, \exists^{\geq\mu})$ such that, for all threshold games $\mathcal{T}$, we have that $\mathcal{T} \models_Z \text{itrap}(x)$ if, and only if, Z defines an I-trap in $\mathcal{T}$ (see Sect. 5.2). By the Interpretation Lemma we get a formula $\text{itrap}^{J(\varphi)}(\lambda, \overline{v})$, also from $\text{FO}(\subseteq, \exists^{\geq\mu})$, which is true in $\mathfrak{A}^*$ precisely for those teams $Y = h^{-1}(Z)$ where Z defines an I-trap of $\mathcal{T}^\#(\mathfrak{A}^*, \varphi)$. Specifically, we can write $\overline{v} = (\overline{xx}', \overline{\mu\mu}')$, so that for the team that defines $X^* = \{(\varphi, s) : s(\overline{x}, \overline{\mu}) \in X\}$ we get that $Y(X) = h^{-1}(X^*)$ is the set of all assignments $(\lambda, \overline{xx}', \overline{\mu\mu}') \mapsto (0, \overline{ac}, \overline{m0})$ for some constant c and with $(\overline{a}, \overline{m}) \in X$. We now set

$$\psi(\overline{x}, \overline{\mu}) := \exists\lambda\exists\overline{x}'\exists\overline{\mu}'(\lambda = 0 \wedge \overline{x}' = \overline{c} \wedge \overline{\mu}' = \overline{0} \wedge \text{itrap}^{J(\varphi)}(\lambda, \overline{xx}', \overline{\mu\mu})).$$

This implies that $\mathfrak{A}^* \models_X \psi(\overline{x}, \overline{\mu}) \iff \mathfrak{A}^* \models_{Y(X)} \text{itrap}^{J(\varphi)}(\lambda, \overline{xx}', \overline{\mu\mu}')$. Putting everything together, we have

$$\begin{aligned}(\mathfrak{A}^*, X) \models &\forall\overline{x\mu}(X\overline{x\mu} \to \varphi(\overline{x}, \overline{\mu})) \iff X^* \text{ is an } I\text{-trap in } \mathcal{T}^\#(\mathfrak{A}^*, \varphi)\\ &\iff \mathcal{T}^\#(\mathfrak{A}^*, \varphi) \models_{X^*} \text{itrap}(x) \iff \mathfrak{A}^* \models_{Y(X)} \text{itrap}^{J(\varphi)}(\lambda, \overline{xx}', \overline{\mu\mu}')\\ &\iff \mathfrak{A}^* \models_X \psi(\overline{x}, \overline{\mu}).\end{aligned}$$ ◀

6 Conclusion

We have explored two main variants for counting constructs in team semantics: forking atoms and counting quantifiers. We have seen that forking atoms are rather powerful, and indeed, dependence and independence logic are equivalent to variants of logics with forking. Counting quantifiers, on the other side, are most interesting in the context of inclusion logic, and we have shown that the extension of inclusion logic by counting quantifiers captures, in a precise sense, fixed-point logic with counting, which provides further interesting connections between team semantics and descriptive complexity theory. To establish the relationship between these two logics we have introduced a new variant of model checking games, threshold safety games, and interpretation arguments for studying them, which we believe to be of intrinsic interest beyond the results of this paper.

An open problem in this context concerns the relationship of different logics that all share the property of closure under unions. Inclusion atoms are closed under arbitrary unions of teams, and the same is true for forking atoms of type $\overline{v} \triangleleft^{\geq\mu} \overline{w}$. Further, counting quantifiers $\exists^{\geq\mu}$ preserve closure under union. By using different combinations of inclusion atoms, forking atoms, and counting quantifiers we thus obtain a number of logics, all of which are closed under unions of teams, but whose relationship and expressive power is not really clear. We conjecture in particular that inclusion logic and $\mathrm{FO}(\mathrm{fork}^{\geq})$ are incomparable.

We remark that, due to the second-order features of team semantics, there is a rich variety of other potential counting constructs related to teams (some of which take the logics beyond Σ^1_1), and the research that we presented here is just a starting point. For instance, one may count the number of subteams of the given team that satisfy a given property, or develop an approach via generalized quantifiers. In [2] a kind of second-order majority quantifier has been considered, which counts witness functions for extending a given team by a new variable, and leads to a logic that captures the polynomial counting hierarchy.

Finally, one of the most important current challenges in the field of logics of dependence and independence is the systematic development of multi-team semantics, where assignments in teams may occur with multiplicities. This is fundamental for instance for reasoning about statistical (rather than logical) dependence and independence. It is obvious that the framework of two-sorted structures and counting constructs may be relevant for this project.

References

1. A. Dawar. The nature and power of fixed-point logic with counting. *ACM SIGLOG News*, 2(1):8–21, 2015.
2. A. Durand, J. Ebbing, J. Kontinen, and H. Vollmer. Dependence logic with a majority quantifier. In *Proceedings of FSTTCS 2011*, pages 252–263, 2011.
3. F. Engström. Generalized quantifiers in dependence logic. *Journal of Logic, Language, and Information*, 2012.
4. P. Galliani. Inclusion and exclusion in team semantics – on some logics of imperfect information. *Annals of Pure and Applied Logic*, 163:68–84, 2012.
5. P. Galliani and L. Hella. Inclusion logic and fixed-point logic. In *Computer Science Logic 2013*, pages 281–295, 2013.
6. E. Grädel. Games for inclusion logic and fixed-point logic. In S. Abramsky et al., editor, *Dependence Logic. Theory and Applications.* Birkhäuser, 2016.
7. E. Grädel and J. Väänänen. Dependence and independence. *Studia Logica*, 101(2):399–410, 2013.
8. Y. Gurevich and S. Shelah. Fixed-point extensions of first-order logic. *Annals of Pure and Applied Logic*, 32:265–280, 1986.
9. W. Hodges. *Model Theory.* Cambridge University Press, 1993.
10. W. Hodges. Compositional semantics for a logic of imperfect information. *Logic Journal of IGPL*, 5:539–563, 1997.
11. N. Immerman. Relational queries computable in polynomial time. *Information and Control*, 68:86–104, 1986.
12. J. Kontinen and J. Väänänen. On definability in dependence logic. *Journal of Logic, Language, and Information*, 18:317–241, 2009.
13. S. Kreutzer. Expressive equivalence of least and inflationary fixed point logic. *Annals of Pure and Applied Logic*, 130:61–78, 2004.
14. R. Rönnholm. Capturing k-ary existential second-order logic with k-ary inclusion-exclusion logic. arXiv:1502.05632v2, 2015.
15. J. Väänänen. *Dependence Logic.* Cambridge University Press, 2007.