Tractability Frontiers in Probabilistic Team Semantics and Existential Second-Order Logic over the Reals

Hannula, Miika

2021

Hannula , M & Virtema , J T 2021 , Tractability Frontiers in Probabilistic Team Semantics and Existential Second-Order Logic over the Reals . in W Faber , G Friedrich , M Gebser & M Morak (eds) , Logics in Artificial Intelligence - 17th European Conference, JELIA 2021, Virtual Event, May 17-20, 2021, Proceedings . vol. 12678 , Lecture Notes in Computer Science , vol. 12678 , pp. 262-278 , Logics in Artificial Intelligence, 17th European Conference , 17/05/2021 . https://doi.org/10.1007/978-3-030-75775-5_18

http://hdl.handle.net/10138/351988
https://doi.org/10.1007/978-3-030-75775-5_18

unspecified
acceptedVersion

Tractability frontiers in probabilistic team semantics and existential second-order logic over the reals

Miika Hannula[a,1,*], Jonni Virtema[b,c,2]

[a]Department of Mathematics and Statistics, University of Helsinki, Helsinki, Finland
[b]Institut für Theoretische Informatik, Leibniz Universität Hannover, Hannover, Germany
[c]Department of Computer Science, University of Sheffield, Sheffield, United Kingdom

Abstract

Probabilistic team semantics is a framework for logical analysis of probabilistic dependencies. Our focus is on the axiomatizability, complexity, and expressivity of probabilistic inclusion logic and its extensions. We identify a natural fragment of existential second-order logic with additive real arithmetic that captures exactly the expressivity of probabilistic inclusion logic. We furthermore relate these formalisms to linear programming, and doing so obtain PTIME data complexity for the logics. Moreover, on finite structures, we show that the full existential second-order logic with additive real arithmetic can only express NP properties. Lastly, we present a sound and complete axiomatization for probabilistic inclusion logic at the atomic level.

Keywords: dependence logic, team semantics, metafinite structures, Blum-Shub-Smale machine

1. Introduction

Metafinite model theory, introduced by Grädel and Gurevich [20], generalizes the approach of *finite model theory* by shifting to two-sorted structures that extend finite structures with another (often infinite) domain with some arithmetic (such as the reals with multiplication and addition), and weight functions bridging the two sorts. A simple example of a metafinite structure is a graph involving numerical labels; e.g., a railway network where an edge between two adjacent stations is labeled by the distance between them. Metafinite structures are, in general, suited for modeling problems that make reference to some numerical domain, be it reals, rationals, or complex numbers.

A particularly important subclass of metafinite structures are the $\mathbb{R}$-*structures*, which extend finite structures with the real arithmetic on the second sort. The computational properties of $\mathbb{R}$-structures can be studied with *Blum-Shub-Smale machines* [6] (BSS machines for short) which are essentially register machines with registers that can store arbitrary real numbers and which can compute rational functions over reals in a single time step.

A particularly important related problem is the existential theory of the reals (ETR), which contains all Boolean combinations of equalities and inequalities of polynomials that have real solutions. Instances of ETR are closely related to the question whether a given finite structure can be extended to an $\mathbb{R}$-structure satisfying certain constraints. Moreover, as we will elaborate more shortly, ETR is also closely related to polynomial time BSS-computations.

Descriptive complexity theory for BSS machines and logics on metafinite structures was initiated by Grädel and Meer who showed that $\mathsf{NP}_{\mathbb{R}}$ (i.e., non-deterministic polynomial time on BSS machines) is captured by a variant of existential second-order logic ($\mathrm{ESO}_{\mathbb{R}}$) over $\mathbb{R}$-structures [22]. Since the work by Grädel and Meer, others (see, e.g., [11, 26, 28, 38]) have shed more light upon *the descriptive complexity over the reals* mirroring the development of classical descriptive complexity.

[*] Corresponding author
Email addresses: miika.hannula@helsinki.fi (Miika Hannula), j.t.virtema@sheffield.ac.uk (Jonni Virtema)

[1] Supported by the Academy of Finland grant 322795.
[2] Supported by the DFG grant VI 1045/1-1.

Complexity over the reals can be related to classical complexity by restricting attention to Boolean inputs. The so-called *Boolean part* of $\mathsf{NP}_{\mathbb{R}}$, written $\mathrm{BP}(\mathsf{NP}_{\mathbb{R}})$, consists of all those Boolean languages that can be recognized by a BSS machine in non-deterministic polynomial time. In contrast to NP, which is concerned with discrete problems that have discrete solutions, this class captures discrete problems with *numerical* solutions. A well studied visibility problem in computational geometry related to deciding existence of numerical solutions is the so-called *art gallery problem*. Here one is asked can a given polygon be guarded by a given number of guards whose positions can be determined with arbitrary precision. Another typical problem is the recognition of unit distance graphs, that is, to determine whether a given graph can be embedded on the Euclidean plane in such a way that two points are adjacent whenever the distance between them is one. These problems [1, 40], and an increasing number of others, have been recognized as complete for the complexity class $\exists\mathbb{R}$, defined as the closure of ETR with polynomial-time reductions [39]. The exact complexity of $\exists\mathbb{R}$ is a major open question; currently it is only known that

$$\mathsf{NP} \leq \exists\mathbb{R} \leq \mathsf{PSPACE}\ \ [8]. \tag{1}$$

Interestingly, $\exists\mathbb{R}$ can also be characterized as the Boolean part of $\mathsf{NP}^0_{\mathbb{R}}$, written $\mathrm{BP}(\mathsf{NP}^0_{\mathbb{R}})$, where $\mathsf{NP}^0_{\mathbb{R}}$ is non-deterministic polynomial time over BSS machines that allow only machine constats 0 and 1 [7, 41]. It follows that $\exists\mathbb{R}$ captures exactly those properties of finite structures that are definable in $\mathrm{ESO}_{\mathbb{R}}$ (with constants 0 and 1). That $\exists\mathbb{R}$ can be formulated in purely descriptive terms has, to the best of our knowledge, never been made explicit in the literature. Indeed, one of the aims of this paper is to promote a descriptive approach to $\exists\mathbb{R}$. In particular, our results show that certain additive fragments of $\mathrm{ESO}_{\mathbb{R}}$, which correspond to subclasses of $\exists\mathbb{R}$, collapse to NP and P.

In addition to metafinite structures, the connection between logical definability encompassing numerical structures and computational complexity has received attention in *constraint databases* [4, 21, 37]. A constraint database models (e.g., geometric data) by combining a numerical *context structure* (such as the real arithmetic) with a finite set of quantifier-free formulae defining infinite database relations [32].

Renewed interest to logics on frameworks analogous to metafinite structures, and related descriptive complexity theory, is motivated by the need to model inferences utilizing numerical data values in the fields of machine learning and artificial intelligence. See e.g. [24, 44] for declarative frameworks for machine learning utilizing logic, [10, 42] for very recent works on logical query languages with arithmetic, and [31] for applications of descriptive complexity in machine learning.

In this paper, we focus on the descriptive complexity of logics with so-called *probabilistic team semantics* as well as additive $\mathrm{ESO}_{\mathbb{R}}$. Team semantics is the semantical framework of modern logics of dependence and independence. Introduced by Hodges [29] and adapted to dependence logic by Väänänen [43], team semantics defines truth in reference to collections of assignments, called *teams*. Team semantics is particularly suitable for a formal analysis of properties, such as the functional dependence between variables, which only arise in the presence of multiple assignments. In the past decade numerous research articles have, via re-adaptations of team semantics, shed more light into the interplay between logic and dependence. A common feature, and limitation, in all these endeavors has been their preoccupation with notions of dependence that are *qualitative* in nature. That is, notions of dependence and independence that make use of quantities, such as conditional independence in statistics, have usually fallen outside the scope of these studies.

The shift to quantitative dependencies in team semantics setting is relatively recent. While the ideas of probabilistic teams trace back to the works of Galliani [16] and Hyttinen et al. [30], a systematic study on the topic can be traced to [14, 15]. In *probabilistic team semantics* the basic semantic units are probability distributions (i.e., *probabilistic teams*). This shift from set based semantics to distribution based semantics enables probabilistic notions of dependence to be embedded to the framework. In [15] probabilistic team semantics was studied in relation to the dependence concept that is most central in statistics: conditional independence. Mirroring [17, 22, 36] the expressiveness of probabilistic independence logic ($\mathrm{FO}(\perp\!\!\!\perp_{\mathrm{c}})$), obtained by extending first-order logic with conditional independence, was in [15, 26] characterised in terms of arithmetic variants of existential second-order logic. In [26] the data complexity of $\mathrm{FO}(\perp\!\!\!\perp_{\mathrm{c}})$ was also identified in the context of BSS machines and the existential theory of the reals. In [25] the focus was shifted to the expressivity hierarchies between probabilistic logics defined in terms of different quantitative dependencies. Recently, the relationship between the settings of probabilistic and relational team semantics has raised interest in the context of quantum information theory [2, 3].

Another vantage point to quantitative dependence comes from the notion of *multiteam semantics*, defined in terms of multisets of variable assignments called *multiteams*. A multiteam can be viewed as a database relation that not only allows duplicate rows (cf. SQL data tables), but also keeps track of the number of times each row is repeated. Multiteam semantics and probabilistic team semantics are close parallels, and they often exhibit similar behavior with respect to their key logics (cf. [14, 23, 45]). There are also differences, namely because the two frameworks are designed to model different situations. For instance, a probability of a random variable can be halved, but it makes no sense to consider a data row that is repeated two and half times in a data table. For this reason, the so-called split disjunction is allowed to cut an assignment weight into two halves in one framework but not (always) in the other.

Of all the dependence concepts thus far investigated in team semantics, that of *inclusion* has arguably turned out to be the most intriguing and fruitful. One reason is that *inclusion logic*, which arises from this concept, can only define properties of teams that are decidable in polynomial time [18]. In contrast, other natural team-based logics, such as dependence and independence logic, capture non-deterministic polynomial time [17, 36, 43], and many variants, such as team logic, have an even higher complexity [35]. Thus it should come as no surprise if quantitative variants of many team-based logics turn out more complex; in principle, adding arithmetical operations and/or counting cannot be a mitigating factor when it comes to complexity.

In this paper, we study *probabilistic inclusion logic*, which is the extension of first-order logic with so-called *marginal identity atoms* $x \approx y$ which state that x and y are identically distributed. Our particular focus is on the complexity and expressivity of *sentences*. It is important, at this point, to note the distinction between formulae and sentences in team-based logics: Formulae describe properties of *teams* (i.e., relations), while sentences describe properties of *structures*. This distinction is even more pointed in probabilistic team semantics, where formulae describe properties *probabilistic teams* (i.e., real-valued probability distributions). On the other hand, sentences of logics with probabilistic team semantics can express variants of important problems that are conjectured not to be expressible in the relational analogues of the logics. Decision problems related to ETR (i.e., the likes of the art gallery problem) are, in particular, these kind of problems. Another motivation to focus on sentences is our desire to *make comparison* between relational and quantitative team logics. As discussed above, the move from relational to quantitative dependence should not in principle make the associated logics weaker. There is, however, no direct mechanism to examine this hypothesis at the formula level, because the team properties of relational and quantitative team logics are essentially incommensurable. Fortunately this becomes possible at the sentence level. The reason is that sentences describe only properties of (finite) structures in *both* logical approaches.

The main takeaway of this paper is that there is no drastic difference between a relational team logic and its quantitative variant, as long as the latter makes only reference to *additive* arithmetic. While inclusion logic translates to fixed point logic, its quantitative variant, probabilistic inclusion logic, seems to require linear programming. Yet, the complexity upper bounds (NP/P) of first-order logic extended with dependence and/or inclusion atoms are preserved upon moving to quantitative variants. In contrast, earlier results indicate that this is not necessarily the case with respect to dependencies whose quantitative expression involves multiplication (such as conditional independence [26]).

Our contribution. We use strong results from linear programming to obtain the following complexity results over finite structures. We identify a natural fragment of additive $\mathrm{ESO}_{\mathbb{R}}$ (that is, *almost conjunctive* $(\ddot{\exists}^*\forall^*)_{\mathbb{R}}[\leq, +, \mathrm{SUM}, 0, 1]$) which captures P on ordered structures (see page 4 for a definition). The full additive $\mathrm{ESO}_{\mathbb{R}}$ is in turn shown to capture NP. Additionally, we establish that the so-called *loose fragments*, almost conjunctive L-$(\ddot{\exists}^*\forall^*)_{[0,1]}[=, \mathrm{SUM}, 0, 1]$ and L-$\mathrm{ESO}_{[0,1]}[=, +, 0, 1]$, of the aforementioned logics have the same expressivity as probabilistic inclusion logic and its extension with dependence atoms, respectively. The characterizations of P and NP hold also for these fragments. Over open formulae, probabilistic inclusion logic extended with dependence atoms is shown to be strictly weaker than probabilistic independence logic. Moreover, we expand from a recent analogous result by Grädel and Wilke on multiteam semantics [23] and show that probabilistic independence cannot be expressed in any logic that has access to only atoms that are relational or closed under so-called *scaled unions*. In contrast, independence logic and inclusion logic with dependence atoms are equally expressive in team semantics [17]. We also show that inclusion logic can be conservatively embedded into its probabilistic variant, when restricted to probabilistic teams that are uniformly distributed. From this we obtain an alternative proof through linear systems (that is entirely different from the original proof of Galliani and Hella [18]) for the fact that inclusion logic can express only polynomial time properties. Finally, we present a sound and complete axiomatization for marginal identity atoms. This is achieved by appending the axiom system of inclusion dependencies with a symmetricity rule.

This paper is an extended version of [27]. Here we include all the proofs that were previously omitted. In addition, the results in Sections 6 and 7 are new.

2. Existential second-order logics on $\mathbb{R}$-structures

In addition to finite relational structures, we consider their numerical extensions by adding real numbers ($\mathbb{R}$) as a second domain sort and functions that map tuples over the finite domain to $\mathbb{R}$. Throughout the paper structures are assumed to have at least two elements. In the sequel, τ and σ will always denote a finite relational and a finite functional vocabulary, respectively. The arities of function variables f and relation variables R are denoted by $\mathrm{ar}(f)$ and $\mathrm{ar}(R)$, resp. If f is a function with domain $\mathrm{Dom}(f)$ and A a set, we define $f \upharpoonright A$ to be the function with domain $\mathrm{Dom}(f) \cap A$ that agrees with f for each element in its domain. Given a finite set S, a function $f\colon S \to [0, 1]$ that maps elements of S to elements of the closed interval $[0, 1]$ of real numbers such that $\sum_{s\in S} f(s) = 1$ is called a *(probability) distribution*, and the *support* of f is defined as $\mathrm{Supp}(f) := \{s \in S \mid f(s) > 0\}$. Also, f is called *uniform* if $f(s) = f(s')$ for all $s, s' \in \mathrm{Supp}(f)$.

Definition 1 ($\mathbb{R}$-structures)**.** *A tuple $\mathfrak{A} = (A, \mathbb{R}, (R^{\mathfrak{A}})_{R\in\tau}, (g^{\mathfrak{A}})_{g\in\sigma})$, where the reduct of $\mathfrak{A}$ to τ is a finite relational structure, and each $g^{\mathfrak{A}}$ is a function from $A^{\mathrm{ar}(g)}$ to $\mathbb{R}$, is called an* $\mathbb{R}$-structure of vocabulary $\tau \cup \sigma$*. Additionally, $\mathfrak{A}$ is also called (i) an S-structure, for $S \subseteq \mathbb{R}$, if each $g^{\mathfrak{A}}$ is a function from $A^{\mathrm{ar}(g)}$ to S, and (ii) a $d[0, 1]$-structure if each $g^{\mathfrak{A}}$ is a distribution.We call $\mathfrak{A}$ a* finite structure*, if $\sigma = \emptyset$.*

Our focus is on a variant of functional existential second-order logic with numerical terms ($\mathrm{ESO}_{\mathbb{R}}$) that is designed to describe properties of $\mathbb{R}$-structures. As first-order terms we have only first-order variables. For a set σ of function symbols, the set of numerical σ-terms i is generated by the following grammar:

$$i ::= c \mid f(\vec{x}) \mid i + i \mid i \times i \mid \mathrm{SUM}_{\vec{y}}\, i,$$

where $\vec{y}$ can be any tuple of variables and include variables that do not occur in i. The interpretations of $+, \times, \mathrm{SUM}$ are the standard addition, multiplication, and summation of real numbers, respectively, and $c \in \mathbb{R}$ is a real constant denoting itself. In particular, the interpretation $[\mathrm{SUM}_{\vec{y}}\, i]^{\mathfrak{A}}_s$ of the term $\mathrm{SUM}_{\vec{y}}\, i$ is defined as follows:

$$[\mathrm{SUM}_{\vec{y}}\, i]^{\mathfrak{A}}_s := \sum_{\vec{a}\in A^{|\vec{y}|}} [i]^{\mathfrak{A}}_{s[\vec{a}/\vec{y}]},$$

where $[i]^{\mathfrak{A}}_{s[\vec{a}/\vec{y}]}$ is an interpretation of the term i. We write $i(\vec{y})$ to mean that the free variables of the term i are exactly the variables in $\vec{y}$. The free variables of a term are defined as usual. In particular, the variables in $\vec{x}$ are not free in $\mathrm{SUM}_{\vec{x}} i(\vec{y})$.

Definition 2 (Syntax of $\mathrm{ESO}_{\mathbb{R}}$)**.** *Let $O \subseteq \{+, \times, \mathrm{SUM}\}$, $E \subseteq \{=, <, \leq\}$, and $C \subseteq \mathbb{R}$. The set of $\tau \cup \sigma$-formulae of $\mathrm{ESO}_{\mathbb{R}}[O, E, C]$ is defined via the grammar:*

$$\phi ::= \ x = y \,|\, \neg x = y \,|\, i\ e\ j \,|\, \neg i\ e\ j \,|\, R(\vec{x}) \,|\, \neg R(\vec{x}) \,|\, \phi \wedge \phi \,|\, \phi \vee \phi \,|\, \exists x\phi \,|\, \forall x\phi \,|\, \exists f\psi,$$

where i and j are numerical σ-terms constructed using operations from O and constants from C; $e \in E$; $R \in \tau$ is a relation symbol; f is a function variable; x, y, and $\vec{x}$ are (tuples of) first-order variables; and ψ is a $\tau \cup (\sigma \cup \{f\})$-formula of $\mathrm{ESO}_{\mathbb{R}}[O, E, C]$.

The semantics of $\mathrm{ESO}_{\mathbb{R}}[O, E, C]$ is defined via $\mathbb{R}$-structures and assignments analogous to first-order logic, however the interpretations of function variables f range over functions $A^{\mathrm{ar}(f)} \to \mathbb{R}$. Furthermore, given $S \subseteq \mathbb{R}$, we define $\mathrm{ESO}_S[O, E, C]$ as the variant of $\mathrm{ESO}_{\mathbb{R}}[O, E, C]$ in which quantification of functions range over $h\colon A^{\mathrm{ar}(f)} \to S$.

Loose fragment. For $S \subseteq \mathbb{R}$, define L-$\mathrm{ESO}_S[O, E, C]$ as the *loose fragment* of $\mathrm{ESO}_S[O, E, C]$ in which negated numerical atoms $\neg i\ e\ j$ are disallowed.

Almost conjunctive. A formula $\phi \in \mathrm{ESO}_S[O, E, C]$ is *almost conjunctive*, if for every subformula $(\psi_1 \vee \psi_2)$ of ϕ, no numerical term occurs in ψ_i, for some $i \in \{1, 2\}$.

Prefix classes. For a regular expression L over the alphabet $\{\ddot{\exists}, \exists, \forall\}$, we denote by $L_S[O, E, C]$ the formulae of $\mathrm{ESO}_S[O, E, C]$ in prefix form whose quantifier prefix is in the language defined by L, where $\ddot{\exists}$ denotes existential function quantification, and $\exists$ and $\forall$ first-order quantification.

***Expressivity comparisons*.** Let $\mathcal{L}$ and $\mathcal{L}'$ be some logics defined above, and let $X \subseteq \mathbb{R}$. For $\phi \in \mathcal{L}$, define $\mathrm{Struc}_X(\phi)$ to be the class of pairs $(\mathfrak{A}, s)$ where $\mathfrak{A}$ is an X-structure and s an assignment such that $\mathfrak{A} \models_s \phi$. Define $\mathrm{Struc}_{\mathrm{fin}}(\phi)$ ($\mathrm{Struc}_{\mathrm{ord}}(\phi)$, resp.) analogously in terms of finite (finite ordered, resp.) structures. Additionally, $\mathrm{Struc}_{d[0,1]}(\phi)$ is the class of $(\mathfrak{A}, s) \in \mathrm{Struc}_{[0,1]}(\phi)$ such that each $f^{\mathfrak{A}}$ is a distribution. If X is a set of reals or from {"$d[0, 1]$","fin", "ord"}, we write $\mathcal{L} \leq_X \mathcal{L}'$ if for all formulae $\phi \in \mathcal{L}$ there is a formula $\psi \in \mathcal{L}'$ such that $\mathrm{Struc}_X(\phi) = \mathrm{Struc}_X(\psi)$. For formulae without free first-order variables, we omit s from the pairs $(\mathfrak{A}, s)$ above. As usual, the shorthand $\equiv_X$ stands for $\leq_X$ in both directions. For $X = \mathbb{R}$, we write simply $\leq$ and $\equiv$.

3. Data complexity of additive $\mathrm{ESO}_{\mathbb{R}}$

On finite structures $\mathrm{ESO}_{\mathbb{R}}[\leq, +, \times, 0, 1]$ is known to capture the complexity class $\exists\mathbb{R}$ [7, 22, 41], which lies somewhere between NP and PSPACE. Here we focus on the additive fragment of the logic. It turns out that the data complexity of the additive fragment is NP and thus no harder than that of ESO. Furthermore, we obtain a tractable fragment of the logic, which captures P on finite ordered structures.

3.1. A tractable fragment

Next we show P data complexity for almost conjunctive $(\ddot{\exists}^*\exists^*\forall^*)_{\mathbb{R}}[\leq, +, \mathrm{SUM}, 0, 1]$.

Proposition 3. *Let ϕ be an almost conjunctive* $\mathrm{ESO}_{\mathbb{R}}[\leq, +, \mathrm{SUM}, 0, 1]$*-formula in which no existential first-order quantifier is in a scope of a universal first-order quantifier. There is a polynomial-time reduction from $\mathbb{R}$-structures $\mathfrak{A}$ and assignments s to families of systems of linear inequations $\mathcal{S}$ such that $\mathfrak{A} \models_s \phi$ if and only if there is a system $S \in \mathcal{S}$ that has a solution. If ϕ has no free function variables, the systems of linear inequations in $\mathcal{S}$ have integer coefficients.*

Proof. Fix ϕ. We assume, w.l.o.g., that variables quantified in ϕ are quantified exactly once, the sets of free and bound variables of ϕ are disjoint, and that the domain of s is the set of free variables of ϕ. Moreover, we assume that ϕ is of the form $\exists\vec{y}\exists\vec{f}\forall\vec{x}\theta$, where $\vec{f}$ is a tuple of function variables and θ is quantifier-free. We use X and Y to denote the sets of variables in $\vec{x}$ and $\vec{y}$, respectively, and $\vec{g}$ to denote the free function variables of ϕ.

We describe a polynomial-time process of constructing a family of systems of linear inequations $\mathcal{S}_{\mathfrak{A},s}$ from a given $\tau \cup \sigma$-structure $\mathfrak{A}$ and an assignment s. We introduce

- a fresh variable $z_{\vec{a},f}$, for each k-ary function symbol f in $\vec{f}$ and k-tuple $\vec{a} \in A^k$.

In the sequel, the variables $z_{\vec{a},f}$ will range over real numbers.

Let $\mathfrak{A}$ be a $\tau \cup \sigma$-structure and s an assignment for the free variables in ϕ. In the sequel, each interpretation for the variables in $\vec{y}$ yields a system of linear equations. Given an interpretation $v\colon Y \to A$, we will denote by S_v the related system of linear equations to be defined below. We then set $\mathcal{S}_{\mathfrak{A},s} := \{S_v \mid v : Y \to A\}$. The system of linear equations S_v is defined as $S_v := \bigcup_{u\colon X \to A} S_v^u$, where S_v^u is defined as follows. Let s_v^u denote the extension of s that agrees with u and v. We let θ_v^u denote the formula obtained from θ by the following simultaneous substitution: If $(\psi_1 \vee \psi_2)$ is a subformula of θ such that no function variable occurs in ψ_i, then $(\psi_1 \vee \psi_2)$ is substituted with $\top$, if

$$\mathfrak{A} \models_{s_v^u} \psi_i, \tag{2}$$

and with ψ_{3-i} otherwise. The set S_v^u is now generated from θ_v^u together with u and v. Note that θ_v^u is a conjunction of first-order or numerical atoms θ_i, $i \in I$, for some index set I. For each conjunct θ_i in which some $f \in \vec{f}$ occurs, add $(\theta_i)_{s_v^u}$ to S_v^u, where $(\psi)_{s_v^u}$ is defined recursively as follows:

$$\begin{aligned}
(\neg\psi)_{s_v^u} &:= \neg(\psi)_{s_v^u}, & (i e j)_{s_v^u} &:= (i)_{s_v^u}\, e\, (j)_{s_v^u}, \text{ for each } e \in \{=, <, \leq, +\},\\
(f(\vec{z}))_{s_v^u} &:= z_{s_v^u(\vec{z}),f}, & (\mathrm{SUM}_{\vec{z}} i)_{s_v^u} &:= \sum_{a \in A^{|\vec{z}|}} (i)_{s_v^u(\vec{a}/\vec{z})},\\
(g(\vec{z}))_{s_v^u} &:= g^{\mathfrak{A}}(s_v^u(\vec{z})), & (x)_{s_v^u} &:= s_v^u(x), \text{ for every variable } x.
\end{aligned}$$

Let θ^* be the conjunction of those conjuncts of θ_v^u in which no $f \in \vec{f}$ occurs. If $\mathfrak{A} \not\models_{s_v^u} \theta^*$, remove S_v from $\mathcal{S}_{\mathfrak{A},s}$.

Since ϕ is fixed, it is clear that $\mathcal{S}_{\mathfrak{A},s}$ can be constructed in polynomial time with respect to $|\mathfrak{A}|$. Moreover, it is straightforward to show that there exists a solution for some $S \in \mathcal{S}_{\mathfrak{A},s}$ exactly when $\mathfrak{A} \models_s \phi$.

Assume first that there exists an $S \in \mathcal{S}_{\mathfrak{A},s}$ that has a solution. Let $w\colon Z \to \mathbb{R}$, where $Z := \{z_{\vec{a},f} \mid f \in \vec{f} \text{ and } \vec{a} \in A^{\mathrm{ar}(f)}\}$, be the function given by a solution for S. By construction, $S = S_v$, for some $v\colon Y \to A$. Let $\mathfrak{A}'$ be the expansion of $\mathfrak{A}$ that interprets each $f \in \vec{f}$ as the function $\vec{a} \mapsto w(z_{\vec{a},f})$. By construction, $\mathfrak{A}' \models_{s_v^u} \theta_v^u$ for every $u\colon X \to A$. Now, from (2) and the related substitutions, we obtain that $\mathfrak{A}' \models_{s_v^u} \theta$ for every $u\colon X \to A$, and hence $\mathfrak{A}' \models_{s_v} \forall x_1 \ldots \forall x_n \theta$. From this $\mathfrak{A} \models_s \phi$ follows.

For the converse, assume that $\mathfrak{A} \models_s \phi$. Hence there exists an extension s_v of s and an expansion $\mathfrak{A}'$ of $\mathfrak{A}$ such that $\mathfrak{A}' \models_{s_v} \forall x_1 \ldots \forall x_n \theta$. Now, by construction, it follows that $S_v \in \mathcal{S}_{\mathfrak{A},s}$ and $\mathfrak{A}' \models_{s_v^u} \theta_v^u$, for every $u\colon X \to A$. Moreover, it follows that the function defined by $z_{\vec{a},f} \mapsto f^{\mathfrak{A}'}(\vec{a})$, for $f \in \vec{f}$ and $\vec{a} \in A^{\mathrm{ar}(f)}$, is a solution for S_v.

□

The above proposition could be strengthened by relaxing the almost conjunctive requirement in any way such that (2) can be still decided (i.e., it suffices that the satisfaction of ψ_is do not depend on the interpretations of the functions in $\vec{f}$).

Theorem 4. *The data complexity of almost conjuctive* $\mathrm{ESO}_{\mathbb{R}}[\leq, +, \mathrm{SUM}, 0, 1]$*-formulae without free function variables and where no existential first-order quantifiers are in a scope of a universal first-order quantifier is in* P.

Proof. Fix an almost conjuctive $\mathrm{ESO}_{\mathbb{R}}[\leq, +, \mathrm{SUM}, 0, 1]$-formula ϕ of relational vocabulary τ of the required form. Given a $\tau \cup \emptyset$ structure $\mathfrak{A}$ and an assignment s for the free variables of ϕ, let $\mathcal{S}$ be the related polynomial size family of polynomial size systems of linear inequations with integer coefficients given by Proposition 3. Deciding whether a system of linear inequalities with integer coefficients has solutions can be done in polynomial time [33]. Thus checking whether there exists a system of linear inequalities $S \in \mathcal{S}$ that has a solution can be done in P as well, from which the claim follows.

□

We will later show that probabilistic inclusion logic captures P on finite ordered structures (Corollary 24) and can be translated to almost conjunctive $\text{L-}(\ddot{\exists}^*\forall^*)_{[0,1]}[\leq, \mathrm{SUM}, 0, 1]$ (Lemma 17). Hence already almost conjuctive $\text{L-}(\ddot{\exists}^*\forall^*)_{\mathbb{R}}[\leq, \mathrm{SUM}, 0, 1]$ captures P.

Corollary 5. *Almost conjunctive* $\text{L-}(\ddot{\exists}^*\forall^*)_{\mathbb{R}}[\leq, \mathrm{SUM}, 0, 1]$ *captures* P *on finite ordered structures.*

3.2. Full additive $\mathrm{ESO}_{\mathbb{R}}$

The goal of this subsection is to prove the following theorem:

Theorem 6. $\mathrm{ESO}_{\mathbb{R}}[\leq, +, \mathrm{SUM}, 0, 1]$ *captures* NP *on finite structures.*

First observe that SUM is definable in $\mathrm{ESO}_{\mathbb{R}}[\leq, +, 0, 1]$: Already $\mathrm{ESO}_{\mathbb{R}}[=]$ subsumes ESO, and thus we may assume a built-in successor function S and its associated minimal and maximal elements min and max on k-tuples over the finite part of the $\mathbb{R}$-structure. Then, for a k-ary tuple of variables $\vec{x}$, $\mathrm{SUM}_{\vec{x}} i$ agrees with $f(\max)$, for any function variable f satisfying $f(\min) = i(\vec{x} \mapsto \min)$ and $f(S(\vec{x})) = f(\vec{x}) + i(S(\vec{x}))$.

As $\mathrm{ESO}_{\mathbb{R}}[\leq, +, 0, 1]$ subsumes ESO, by Fagin's theorem, it can express all NP properties. Thus we only need to prove that any $\mathrm{ESO}_{\mathbb{R}}[\leq, +, 0, 1]$-definable property of finite structures is recognizable in NP. The proof relies on (descriptive) complexity theory over the reals. The fundamental result in this area is that existential second-order logic over the reals ($\mathrm{ESO}_{\mathbb{R}}[\leq, +, \times, (r)_{r\in\mathbb{R}}]$) corresponds to non-deterministic polynomial time over the reals ($\mathrm{NP}_{\mathbb{R}}$) for BSS machines [22, Theorem 4.2]. To continue from this, some additional terminology is needed. We refer the reader to Appendix A (or to the textbook [5]) for more details about BSS machines. Let $C_{\mathbb{R}}$ be a complexity class over the reals.

- C_{add} is $C_{\mathbb{R}}$ restricted to *additive* BSS machines (i.e., without multiplication).
- $C^0_{\mathbb{R}}$ is $C_{\mathbb{R}}$ restricted to BSS machines with machine constants 0 and 1 only.
- $\mathrm{BP}(C_{\mathbb{R}})$ is $C_{\mathbb{R}}$ restricted to languages of strings that contain only 0 and 1.

A straightforward adaptation of [22, Theorem 4.2] yields the following theorem.

Theorem 7 ([22])**.** $\mathrm{ESO}_{\mathbb{R}}[\leq,+,0,1]$ *captures* $\mathsf{NP}^0_{\mathrm{add}}$ *on* $\mathbb{R}$*-structures.*

If we can establish that $\mathrm{BP}(\mathsf{NP}^0_{\mathrm{add}})$, the so-called *Boolean part* of $\mathsf{NP}^0_{\mathrm{add}}$, collapses to NP, we have completed the proof of Theorem 6. Observe that another variant of this theorem readily holds; $\mathrm{ESO}_{\mathbb{R}}[=,+,(r)_{r\in\mathbb{R}}]$-definable properties of $\mathbb{R}$-structures are recognizable in $\mathsf{NP}_{\mathrm{add}}$ branching on equality, which in turn, over Boolean inputs, collapses to NP [34, Theorem 3]. Here, restricting branching to equality is crucial. With no restrictions in place (the BSS machine by default branches on inequality and can use arbitrary reals as machine constants) $\mathsf{NP}_{\mathrm{add}}$ equals $\mathsf{NP/poly}$ over Boolean inputs [34, Theorem 11]. Adapting arguments from [34], we show next that disallowing machine constants other than 0 and 1, but allowing branching on inequality, is a mixture that leads to a collapse to NP.

Theorem 8. $\mathrm{BP}(\mathsf{NP}^0_{\mathrm{add}}) = \mathsf{NP}$.

Proof. Clearly $\mathsf{NP} \leq \mathrm{BP}(\mathsf{NP}^0_{\mathrm{add}})$; a Boolean guess for an input $\vec{x}$ can be constructed by comparing to zero each component of a real guess $\vec{y}$, and a polynomial-time Turing computation can be simulated by a polynomial-time BSS computation.

For the converse, let $L \subseteq \{0,1\}^*$ be a Boolean language that belongs to $\mathrm{BP}(\mathsf{NP}^0_{\mathrm{add}})$; we need to show that L belongs also to NP. Let M be a BSS machine such that its running time is bounded by some polynomial p, and for all Boolean inputs $\vec{x} \in \{0,1\}^*$, $\vec{x} \in L$ if and only if there is $\vec{y} \in \mathbb{R}^{p(|x|)}$ such that M accepts $(\vec{x},\vec{y})$.

We describe a non-deterministic algorithm that decides L and runs in polynomial time. Given a Boolean input $\vec{x}$ of length n, first guess the outcome of each comparison in the BSS computation; this guess is a Boolean string $\vec{z}$ of length $p(n)$. Note that each configuration of a polynomial time BSS computation can be encoded by a real string of polynomial length. During the BSS computation the value of each coordinate of its configuration is a linear function on the constants 0 and 1, the input $\vec{x}$, and the real guess $\vec{y}$ of length $p(n)$. Thus it is possible to construct in polynomial time a system $\mathcal{S}$ of linear inequations on $\vec{y}$ of the form

$$\sum_{j=1}^{p(n)} a_{ij}y_j \leq 0 \quad (1 \leq i \leq m) \quad \text{and} \quad \sum_{j=1}^{p(n)} b_{ij}y_j < 0 \quad (1 \leq i \leq l), \tag{3}$$

where $a_{ij} \in \mathbb{Z}$, such that $\vec{y}$ is a (real-valued) solution to $\mathcal{S}$ if and only if M accepts $(\vec{x},\vec{y})$ with respect to the outcomes $\vec{z}$. In (3), the variables y_j stand for elements of the real guess $\vec{y}$, and $m+l$ is the total number of comparisons. Each comparison generates either a strict or a non-strict inequality, depending on the outcome encoded by $\vec{z}$.

Without loss of generality we may assume additional constraints of the form $y_j \geq 0$ $(1 \leq j \leq p(n))$ (cf. [12, p. 86]). Transform then $\mathcal{S}$ to another system of inequalities $\mathcal{S}'$ obtained from $\mathcal{S}$ by replacing strict inequalities in (3) by

$$\sum_{j=1}^{p(n)} b_{ij}y_j + \epsilon \leq 0 \quad (1 \leq i \leq l) \quad \text{and} \quad \epsilon \leq 1,$$

Then determine the solution of the linear program: maximize $(\vec{0},1)(\vec{y},\epsilon)^T$ subject to $\mathcal{S}'$ and $(\vec{y},\epsilon) \geq 0$. If there is no solution or the solution is zero, then reject; otherwise accept. Since $\mathcal{S}'$ is of polynomial size and linear programming is in polynomial time [33], the algorithm runs in polynomial time. Clearly, the algorithm accepts $\vec{x}$ for some guess $\vec{z}$ if and only if $\vec{x} \in L$. □

4. Probabilistic team semantics and additive $\mathrm{ESO}_{\mathbb{R}}$

4.1. Probabilistic team semantics

Let D be a finite set of first-order variables and A a finite set. A *team* X is a set of assignments from D to A. A *probabilistic team* is a distribution $\mathbb{X}\colon X \to [0,1]$, where X is a finite team. Also the empty function is considered a probabilistic team. We call D the variable domain of both X and $\mathbb{X}$, written $\mathrm{Dom}(\mathbb{X})$ and $\mathrm{Dom}(X)$. A is called the *value domain* of X and $\mathbb{X}$.

Let $\mathbb{X} : X \to [0,1]$ be a probabilistic team, x a variable, $V \subseteq \mathrm{Dom}(\mathbb{X})$ a set of variables, and A a set. The *projection* of $\mathbb{X}$ on V is defined as $\mathrm{Pr}_V(\mathbb{X}) : X \upharpoonright V \to [0,1]$ such that $s \mapsto \sum_{t\upharpoonright V=s} \mathbb{X}(t)$, where $X \upharpoonright V := \{t \upharpoonright V \mid t \in X\}$.

Define $\mathrm{S}_{x,A}(\mathbb{X})$ as the set of all probabilistic teams $\mathbb{Y}$ with variable domain $\mathrm{Dom}(\mathbb{X}) \cup \{x\}$ such that $\mathrm{Pr}_{\mathrm{Dom}(\mathbb{X})\setminus\{x\}}(\mathbb{Y}) = \mathrm{Pr}_{\mathrm{Dom}(\mathbb{X})\setminus\{x\}}(\mathbb{X})$ and A is a value domain of $\mathbb{Y} \upharpoonright \{x\}$. We denote by $\mathbb{X}[A/x]$ the unique $\mathbb{Y} \in \mathrm{S}_{x,A}(\mathbb{X})$ such that

$$\mathbb{Y}(s) = \frac{\mathrm{Pr}_{\mathrm{Dom}(\mathbb{X})\setminus\{x\}}(\mathbb{X})(s \upharpoonright \mathrm{Dom}(\mathbb{X}) \setminus \{x\})}{|A|}.$$

If x is a fresh variable, then this equation becomes $\mathbb{Y}(s(a/x)) = \frac{\mathbb{X}(s)}{|A|}$. We also define $X[A/x] := \{s(a/x) \mid s \in X, a \in A\}$, and write $\mathbb{X}[a/x]$ and $X[a/x]$ instead of $\mathbb{X}[\{a\}/x]$ and $X[\{a\}/x]$, for singletons $\{a\}$.

Let us also define some function arithmetic. Let α be a real number, and f and g be functions from a shared domain into real numbers. The scalar multiplication αf is a function defined by $(\alpha f)(x) := \alpha f(x)$. The addition $f + g$ is defined as $(f + g)(x) = f(x) + g(x)$, and the multiplication fg is defined as $(fg)(x) := f(x)g(x)$. In particular, if f and g are probabilistic teams and $\alpha + \beta = 1$, then $\alpha f + \beta g$ is a probabilistic team.

We define first probabilistic team semantics for first-order formulae. As is customary in the team semantics context, we restrict attention to formulae in negation normal form. If ϕ is a first-order formula, we write $\phi^{\perp}$ for the equivalent formula obtained from $\neg\phi$ by pushing the negation in front of atomic formulae. If furthermore ψ is some (not necessarily first-order) formula, we then use a shorthand $\phi \to \psi$ for the formula $\phi^{\perp} \vee (\phi \wedge \psi)$.

Definition 9 (Probabilistic team semantics)**.** *Let $\mathfrak{A}$ be a τ-structure over a finite domain A, and $\mathbb{X}\colon X \to [0,1]$ a probabilistic team. The satisfaction relation $\models_{\mathbb{X}}$ for first-order logic is defined as follows:*

$$\begin{array}{lll}
\mathfrak{A} \models_{\mathbb{X}} l & \Leftrightarrow & \forall s \in \mathrm{Supp}(\mathbb{X}) : \mathfrak{A} \models_s l, \textit{ where } l \textit{ is a literal} \\
\mathfrak{A} \models_{\mathbb{X}} (\psi \wedge \theta) & \Leftrightarrow & \mathfrak{A} \models_{\mathbb{X}} \psi \textit{ and } \mathfrak{A} \models_{\mathbb{X}} \theta \\
\mathfrak{A} \models_{\mathbb{X}} (\psi \vee \theta) & \Leftrightarrow & \mathfrak{A} \models_{\mathbb{Y}} \psi \textit{ and } \mathfrak{A} \models_{\mathbb{Z}} \theta, \textit{ for some probabilistic teams } \mathbb{Y} \textit{ and } \mathbb{Z}, \textit{ and} \\
 & & \alpha \in [0,1] \textit{ such that } \alpha\mathbb{Y} + (1-\alpha)\mathbb{Z} = \mathbb{X} \\
\mathfrak{A} \models_{\mathbb{X}} \forall x\psi & \Leftrightarrow & \mathfrak{A} \models_{\mathbb{X}[A/x]} \psi \\
\mathfrak{A} \models_{\mathbb{X}} \exists x\psi & \Leftrightarrow & \mathfrak{A} \models_{\mathbb{Y}} \psi \textit{ for some } \mathbb{Y} \in \mathrm{S}_{x,A}(\mathbb{X})
\end{array}$$

The satisfaction relation $\models_s$ denotes the Tarski semantics of first-order logic. If ϕ is a *sentence* (i.e., without free variables), then $\mathfrak{A}$ *satisfies* ϕ, written $\mathfrak{A} \models \phi$, if $\mathfrak{A} \models_{\mathbb{X}_{\emptyset}} \phi$, where $\mathbb{X}_{\emptyset}$ is the distribution that maps the empty assignment to 1.

We make use of a generalization of probabilistic team semantics where the requirement of being a distribution is dropped. A *weighted team* is any non-negative weight function $\mathbb{X}\colon X \to \mathbb{R}_{\geq 0}$. Given a first-order formula α, we write $\mathbb{X}_{\alpha}$ for the restriction of the weighted team $\mathbb{X}$ to the assignments of X satisfying α (with respect to the underlying structure). Moreover, the *total weight* of a weighted team $\mathbb{X}$ is $|\mathbb{X}| := \sum_{s \in X} \mathbb{X}(s)$.

Definition 10 (Weighted semantics)**.** *Let $\mathfrak{A}$ be a τ-structure over a finite domain A, and $\mathbb{X}\colon X \to \mathbb{R}_{\geq 0}$ a weighted team. The satisfaction relation $\models^{w}_{\mathbb{X}}$ for first-order logic is defined exactly as in Definition 9, except that for $\vee$ we define instead:*

$$\mathfrak{A} \models^{w}_{\mathbb{X}} (\psi \vee \theta) \quad \Leftrightarrow \quad \mathfrak{A} \models_{\mathbb{Y}} \psi \textit{ and } \mathfrak{A} \models_{\mathbb{Z}} \theta \textit{ for some } \mathbb{Y}, \mathbb{Z} \textit{ s.t. } \mathbb{Y} + \mathbb{Z} = \mathbb{X}.$$

We consider logics with the following atomic dependencies:

Definition 11 (Dependencies)**.** *Let $\mathfrak{A}$ be a finite structure with universe A, $\mathbb{X}$ a weighted team, and X a team.*

- ***Marginal identity and inclusion atoms**. If $\vec{x}, \vec{y}$ are variable sequences of length k, then $\vec{x} \approx \vec{y}$ is a* marginal identity atom *and $\vec{x} \subseteq \vec{y}$ is an* inclusion atom *with satisfactions defined as:*

$$\mathfrak{A} \models^{w}_{\mathbb{X}} \vec{x} \approx \vec{y} \Leftrightarrow |\mathbb{X}_{\vec{x}=\vec{a}}| = |\mathbb{X}_{\vec{y}=\vec{a}}| \textit{ for each } \vec{a} \in A^k,$$
$$\mathfrak{A} \models_X \vec{x} \subseteq \vec{y} \Leftrightarrow \textit{for all } s \in X \textit{ there is } s' \in X \textit{ such that } s(\vec{x}) = s'(\vec{y}).$$

- ***Probabilistic independence atom.** If $\vec{x}, \vec{y}, \vec{z}$ are variable sequences, then $\vec{y} \perp\!\!\!\perp_{\vec{x}} \vec{z}$ is a* probabilistic (conditional) independence atom *with satisfaction defined as:*

$$\mathfrak{A} \models_{\mathbb{X}} \vec{y} \perp\!\!\!\perp_{\vec{x}} \vec{z}$$

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

$$|\mathbb{X}_{\vec{x}\vec{y}=s(\vec{x}\vec{y})}| \cdot |\mathbb{X}_{\vec{x}\vec{z}=s(\vec{x}\vec{z})}| = |\mathbb{X}_{\vec{x}\vec{y}\vec{z}=s(\vec{x}\vec{y}\vec{z})}| \cdot |\mathbb{X}_{\vec{x}=s(\vec{x})}|.$$

We also write $\vec{x} \perp\!\!\!\perp \vec{y}$ *for the* probabilistic marginal independence atom, *defined as* $\vec{x} \perp\!\!\!\perp_{\emptyset} \vec{y}$.

- ***Dependence atom***. *For a sequence of variables* $\vec{x}$ *and a variable* y, $=(\vec{x}, y)$ *is a* dependence atom *with satisfaction defined as:*
$$\mathfrak{A} \models_X =(\vec{x}, y) \Leftrightarrow \textit{for all } s, s' \in X: \ \textit{if } s(\vec{x}) = s'(\vec{x}), \textit{ then } s(y) = s'(y).$$

For probabilistic teams $\mathbb{X}$, *the satisfaction relation is written without the superscript* w.

Observe that any dependency α over team semantics can also be interpreted in probabilistic team semantics: $\mathfrak{A} \models_{\mathbb{X}} \alpha$ iff $\mathfrak{A} \models_{\mathrm{Supp}(\mathbb{X})} \alpha$. For a list C of dependencies, we write $\mathrm{FO}(C)$ for the extension of first-order logic with the dependencies in C. The logics $\mathrm{FO}(\approx)$ and $\mathrm{FO}(\subseteq)$, in particular, are called *probabilistic inclusion logic* and *inclusion logic*, respectively. Furthermore, *probabilistic independence logic* is denoted by $\mathrm{FO}(\perp\!\!\!\perp_{\mathrm{c}})$, and its restriction to probabilistic marginal independence atoms by $\mathrm{FO}(\perp\!\!\!\perp)$. We write $\mathrm{Fr}(\phi)$ for the set free variables of $\phi \in \mathrm{FO}(C)$, defined as usual. We conclude this section with a list of useful equivalences. We omit the proofs, which are straightforward structural inductions ((ii) was also proven in [25] and (v) follows from (i) and the flatness property of team semantics).

Proposition 12. *Let* $\phi \in \mathrm{FO}(C)$, $\psi \in \mathrm{FO}(\approx, C)$, *and* $\theta \in \mathrm{FO}$, *where* C *is a list of dependencies over team semantics. Let* $\mathfrak{A}$ *be a structure,* $\mathbb{X}$ *a weighted team, and* r *any positive real. The following equivalences hold:*

(i) $\mathfrak{A} \models^{w}_{\mathbb{X}} \phi \;\Leftrightarrow\; \mathfrak{A} \models_{\mathrm{Supp}(\mathbb{X})} \phi$.

(ii) $\mathfrak{A} \models^{w}_{\mathbb{X}} \psi \;\Leftrightarrow\; \mathfrak{A} \models_{\frac{1}{|\mathbb{X}|}\mathbb{X}} \psi$.

(iii) $\mathfrak{A} \models^{w}_{\mathbb{X}} \psi \;\Leftrightarrow\; \mathfrak{A} \models^{w}_{r\mathbb{X}} \psi$.

(iv) $\mathfrak{A} \models^{w}_{\mathbb{X}} \psi \;\Leftrightarrow\; \mathfrak{A} \models^{w}_{\mathbb{X}\upharpoonright V} \psi$, *where* $\mathrm{Fr}(\psi) \subseteq V$.

(v) $\mathfrak{A} \models^{w}_{\mathbb{X}} \theta \;\Leftrightarrow\; \mathfrak{A} \models_{s} \theta$, *for all* $s \in \mathrm{Supp}(X)$.

4.2. Expressivity of probabilistic inclusion logic

We turn to the expressivity of probabilistic inclusion logic and its extension with dependence atoms. In particular, we relate these logics to existential second-order logic over the reals. We show that probabilistic inclusion logic extended with dependence atoms captures a fragment in which arithmetic is restricted to summing. Furthermore, we show that leaving out dependence atoms is tantamount to restricting to sentences in almost conjunctive form with $\ddot{\exists}^*\forall^*$ quantifier prefix.

Expressivity comparisons.. Fix a list of atoms C over probabilistic team semantics. For a probabilistic team $\mathbb{X}$ with variable domain $\{x_1, \ldots, x_n\}$ and value domain A, the function $f_{\mathbb{X}} : A^n \to [0,1]$ is defined as the probability distribution such that $f_{\mathbb{X}}(s(\vec{x})) = \mathbb{X}(s)$ for all $s \in X$. For a formula $\phi \in \mathrm{FO}(C)$ of vocabulary τ and with free variables $\{x_1, \ldots, x_n\}$, the class $\mathrm{Struc}_{d[0,1]}(\phi)$ is defined as the class of $d[0,1]$-structures $\mathfrak{A}$ over $\tau \cup \{f\}$ such that $(\mathfrak{A} \upharpoonright \tau) \models_{\mathbb{X}} \phi$, where $f_{\mathbb{X}} = f^{\mathfrak{A}}$ and $\mathfrak{A} \upharpoonright \tau$ is the finite τ-structure underlying $\mathfrak{A}$. Let $\mathcal{L}$ and $\mathcal{L}'$ be two logics of which one is defined over (probabilistic) team semantics. We write $\mathcal{L} \leq \mathcal{L}'$ if for every formula $\phi \in \mathcal{L}$ there is $\phi' \in \mathcal{L}'$ such that $\mathrm{Struc}_{d[0,1]}(\phi) = \mathrm{Struc}_{d[0,1]}(\phi')$; again, $\equiv$ is a shorthand for $\leq$ both ways.

Theorem 13. *The following equivalences hold:*

(i) $\mathrm{FO}(\approx, =(\cdots)) \equiv \text{L-ESO}_{[0,1]}[=, +, 0, 1]$.

(ii) $\mathrm{FO}(\approx) \equiv$ *almost conjunctive* $\text{L-}(\ddot{\exists}^*\forall^*)_{[0,1]}[=, \mathrm{SUM}, 0, 1]$.

We divide the proof of Theorem 13 into two parts. In Section 4.3 we consider the direction from probabilistic team semantics to existential second-order logic over the reals, and in Section 4.4 we shift attention to the converse direction. In order to simplify the presentation in the forthcoming subsections, we start by showing how to replace existential function quantification by distribution quantification. The following lemma in its original form includes multiplication (see [26, Lemma 6.4]) but works also without it.

Lemma 14 ([26])**.** $\text{L-ESO}_{[0,1]}[=, +, 0, 1] \equiv_{d[0,1]} \text{L-ESO}_{d[0,1]}[=, \mathrm{SUM}]$.

The proof, however, does not preserve the almost conjunctive form. That case is dealt with separately in Proposition 16. As shown next, we can utilize in this proposition the fact that the real constants 0 and 1 are definable in almost conjunctive $\text{L-}(\ddot{\exists}^*\forall^*)_{d[0,1]}[=, \mathrm{SUM}]$.

Lemma 15. L-ESO$_{d[0,1]}$[=, SUM] $\equiv_{\mathbb{R}}$ L-ESO$_{d[0,1]}$[=, SUM, 0, 1]. *The same holds when both logics are restricted to almost conjunctive formulae of the prefix class* $\ddot{\exists}^*\forall^*$.

Proof. Any formula θ involving 0 or 1 can be equivalently expressed as follows:

$$\exists n \exists f \exists h \forall x \forall y \forall z \big(f(x) = h(x, x) \wedge (y = z \vee \theta(h(y, z)/0, n/1)) \big),$$

where n is nullary. □

Proposition 16. L-ESO$_{[0,1]}$[=, SUM, 0, 1] $\equiv_{[0,1]}$ L-ESO$_{d[0,1]}$[=, SUM]. *The same holds when both logics are restricted to almost conjunctive formulae of the prefix class* $\ddot{\exists}^*\forall^*$.

Proof. The $\geq$-direction is trivial. We show the $\leq$-direction, which is similar to the proof of [26, Lemma 6.4]. By Lemma 15 we may assume that almost conjunctive L-$(\ddot{\exists}^*\forall^*)_{d[0,1]}$[=, SUM] (as well as L-ESO$_{d[0,1]}$[=, SUM]) contains real constants 0 and 1. Suppose ϕ is some formula in L-ESO$_{[0,1]}$[=, SUM, 0, 1]. Let k be the maximal arity of any function variable/symbol appearing in ϕ. The total sum of the weights of any interpretation of a function occurring in ϕ on a given structure, whose finite domain is of size n, is at most n^k. We now show how to obtain from ϕ an equivalent formula in L-ESO$_{d[0,1]}$[SUM, =, 0, 1]; the idea is to scale all function weights by $1/n^k$. Note first that the value $1/n^k$ can be expressed via a k-ary distribution variable g as follows:

$$\exists g \forall \vec{x}\vec{y}\, g(\vec{x}) = g(\vec{y})$$

Below, we write $\frac{1}{n^k}$ instead of $g(\vec{x})$.

Suppose ϕ is of the form $\exists f_1 \ldots f_m \forall \vec{x} \theta$, where θ is quantifier free, and let $g_1, \ldots, g_t$ be the list of (non-quantified) function symbols of ϕ. Define

$$\phi' := \exists f'_1 \ldots f'_m g'_1 \ldots g'_t \forall \vec{x}\vec{x}' \, (\psi \wedge \theta'),$$

where each f'_j ($g'(j)$, resp.) is an ar(f_j) + 1-ary (ar(g_j) + k + 1-ary, resp.) distribution variable and ψ and θ' are as defined below. The universally quantified variables $\vec{x}'$ list all of the newly introduced variables of the construction below. The formula ψ is used to express that each f'_j ($g'(j)$, resp.) is an $1/n^k$-scaled copy of f_j ($g(j)$, resp.). That is, ψ is defined as the formula

$$\bigwedge_{i \leq m} f'_j(\vec{y}, y_l) \leq \frac{1}{n^k} \wedge \bigwedge_{i \leq t} (g'_j(\vec{y}, \vec{z}, z_l) = g'_j(\vec{y}, \vec{z}', z'_l) \wedge \mathrm{SUM}_{\vec{z}}\, g'_j(\vec{y}, \vec{z}, z_l) = g_j(\vec{y})),$$

where y_l and z_l (here and below) denote the last elements of the tuples $\vec{y}$ and $\vec{z}$, respectively.[3] Finally θ' is obtained from θ by replacing expressions of the form $f_j(\vec{y})$ and $g_j(\vec{y})$ by $f'_j(\vec{y}, y_l)$ and $g_j(\vec{y}, \vec{z}, z_l)$, resp., and the real constant 1 by $\frac{1}{n^k}$. A straightforward inductive argument on the structure of formulae yields that, over [0, 1]-structures, ϕ and ϕ' are equivalent. Note that ϕ' is an almost conjunctive formula of the prefix class $\ddot{\exists}^*\forall^*$, if ϕ is. □

4.3. *From probabilistic team semantics to existential second-order logic*

Let c and d be two distinct constants. Let $\phi(\vec{x}) \in$ FO($\approx$, =($\cdots$)) be a formula whose free variables are from the sequence $\vec{x} = (x_1, \ldots, x_n)$. We now construct recursively an L-ESO$_{[0,1]}$[=, SUM, 0, 1]-formula $\phi^*(f)$ that contains one free n-ary function variable f. In this formula, a probabilistic team $\mathbb{X}$ is represented as a function $f_{\mathbb{X}}$ such that $\mathbb{X}(s) = f_{\mathbb{X}}(s(x_1), \ldots, s(x_n))$.

(1) If $\phi(\vec{x})$ is a first-order literal, then

$$\phi^*(f) := \forall \vec{x}(f(\vec{x}) = 0 \vee \phi(\vec{x})).$$

(2) If $\phi(\vec{x})$ is a dependence atom of the form $=(\vec{x}_0, x_1)$, then

$$\phi^*(f) := \forall \vec{x}\vec{x}'(f(\vec{x}) = 0 \vee f(\vec{x}') = 0 \vee \vec{x}_0 \neq \vec{x}'_0 \vee x_1 = x'_1).$$

[3] For a 0-ary function f, a construction $f'(\vec{z}, z_l) = f'(\vec{z}', z'_l)$ can be used instead.

(3) If $\phi(\vec{x})$ $\vec{x}_0 \approx \vec{x}_1$, where $\vec{x} = \vec{x}_0\vec{x}_1\vec{x}_2$, then

$$\phi^*(f) := \forall\vec{y}\,\mathrm{SUM}_{\vec{x}_1,\vec{x}_2} f(\vec{y},\vec{x}_1,\vec{x}_2) = \mathrm{SUM}_{\vec{x}_0,\vec{x}_2} f(\vec{x}_0,\vec{y},\vec{x}_2).$$

(4) If $\phi(\vec{x})$ is of the form $\psi_0(\vec{x}) \wedge \psi_1(\vec{x})$, then

$$\phi^*(f) := \psi_0^*(f) \wedge \psi_1^*(f).$$

(5) If $\phi(\vec{x})$ is of the form $\psi_0(\vec{x}) \vee \psi_1(\vec{x})$, then

$$\phi^*(f) := \exists g \forall\vec{x}(\mathrm{SUM}_y g(\vec{x},y) = f(\vec{x}) \wedge \forall y(y = c \vee y = d \vee g(\vec{x},y) = 0) \wedge \psi_0^*(g^c) \wedge \psi_1^*(g^d)),$$

where g^i is of the same arity as f and defined as $g^i(\vec{x}) := g(\vec{x}, i)$.

(6) If $\phi(\vec{x})$ is $\exists y\psi(\vec{x},y)$, then

$$\phi^*(f) := \exists g((\forall\vec{x}\,\mathrm{SUM}_y g(\vec{x},y) = f(\vec{x})) \wedge \psi^*(g)).$$

(7) If $\phi(\vec{x})$ is of the form $\forall y\psi(\vec{x},y)$, then

$$\phi^*(f) := \exists g(\forall\vec{x}(\forall y\forall z g(\vec{x},y) = g(\vec{x},z) \wedge\ \mathrm{SUM}_y g(\vec{x},y) = f(\vec{x})) \wedge \psi^*(g)).$$

This translation leads to the following lemma,

Lemma 17. *The following hold:*

(i) $\mathrm{FO}(\approx, =(\cdots)) \leq \mathrm{L}\text{-}(\ddot{\exists}^*\forall^*)_{[0,1]}[=, \mathrm{SUM}, 0, 1]$.

(ii) $\mathrm{FO}(\approx, =(\cdots)) \leq$ *almost conjunctive* $\mathrm{L}\text{-}(\ddot{\exists}^*\forall^*\exists^*)_{[0,1]}[=, \mathrm{SUM}, 0, 1]$.

(iii) $\mathrm{FO}(\approx) \leq$ *almost conjunctive* $\mathrm{L}\text{-}(\ddot{\exists}^*\forall^*)_{[0,1]}[=, \mathrm{SUM}, 0, 1]$.

Proof. By item (ii) of Proposition 12, we may use weighted semantics (Definition 10). Then, a straightforward induction shows that for all structures $\mathfrak{A}$ and non-empty weighted teams $\mathbb{X}\colon X \to [0,1]$, with variable domain $\vec{x}$, such that $|\mathbb{X}| \leq 1$,

$$\mathfrak{A} \models^w_{\mathbb{X}} \phi(\vec{x}) \iff (\mathfrak{A}, f_{\mathbb{X}}) \models \phi^*(f). \tag{4}$$

Furthermore, the extra constants c and d can be discarded. Define $\psi(f)$ as

$$\exists f'\forall cd\forall\vec{x}(f'(\vec{x},c,d) = f(\vec{x}) \wedge (c \neq d \to \phi^{**}(f'))), \tag{5}$$

where $\phi^{**}(f')$ is obtained from $\phi^*(f)$ by replacing function terms $f(t_1,\ldots,t_n)$ with $f'(t_1,\ldots,t_n,c,d)$. There are only existential function and universal first-order quantifiers in (5). By pushing these quantifiers in front, and by swapping the ordering of existential and universal quantifiers (by increasing the arity of function variables and associated function terms), we obtain a sentence $\psi^*(f) \in \mathrm{L}\text{-}(\ddot{\exists}^*\forall^*)_{d[0,1]}[=, \mathrm{SUM}, 0, 1]$ which, if substituted for $\phi^*(f)$, satisfies (4).

Let us then turn to the items of the lemma.

(i) The claim readily holds.

(ii) The claim follows if the translation for dependence atoms $=(\vec{x}_0, x_1)$ and $\vec{x} = \vec{x}_0 x_1 \vec{x}_2$ is replaced by

$$\phi^*(f) := \forall\vec{x}_0\exists x_1 \mathrm{SUM}_{\vec{x}_2} f(\vec{x}) = \mathrm{SUM}_{x_1\vec{x}_2} f(\vec{x}).$$

We conclude that $\phi^*(f)$ interprets the dependence atom in the correct way and it preserves the almost conjunctive form and the required prefix form.

(iii) For the claim, it suffices to drop the translation of the dependence atom.

□

This completes the "≤" direction of Theorem 13. For (i), this follows from (i) of Lemma 17, Proposition 16, and Lemma 14. For (ii), only (iii) of Lemma 17 is needed.

Recall from Proposition 3 that almost conjunctive $(\ddot{\exists}^*\exists^*\forall^*)_{\mathbb{R}}[\leq, +, \mathrm{SUM}, 0, 1]$ is in PTIME in terms of data complexity. Since dependence logic captures NP [43], the previous lemma indicates that we have found, in some regard, a maximal tractable fragment of additive existential second-order logic. That is, dropping either the requirement of being almost conjunctive, or that of having the prefix form $\ddot{\exists}^*\exists^*\forall^*$, leads to a fragment that captures NP; that NP is also an upper bound for these fragments follows by Theorem 6.

Corollary 18. $\mathrm{FO}(\approx, =(\cdots))$ *captures* NP *on finite structures.*

4.4. From existential second-order logic to probabilistic team semantics

Due to Lemma 14 and Proposition 16, our aim is to translate L-ESO$_{d[0,1]}[=, \mathrm{SUM}]$ and almost conjunctive L-ESO$_{d[0,1]}[=, \mathrm{SUM}]$ to FO$(\approx, =(\cdots))$ and FO$(\approx)$, respectively. The following lemmas imply that we may restrict attention to formulae in Skolem normal form.[4]

We first need to get rid of all numerical terms whose interpretation does not belong to the unit interval. The only source of such terms are summation terms of the form $\mathrm{SUM}_{\vec{x}} i(\vec{y})$, where $\vec{x}$ is a sequence of variables that contain a variable z not belonging to $\vec{y}$; we call such instances of z *dummy-sum instances*. For example, the summation term $\mathrm{SUM}_x n$, where n is the nullary distribution and x a dummy-sum instance, is always interpreted as the cardinality of the model's domain.

Lemma 19. *For every* L-ESO$_{d[0,1]}[=, \mathrm{SUM}]$*-formula ϕ there exists an equivalent formula without dummy-sum instances.*

Proof. Let k be the number of dummy sum-instances in ϕ. Without loss of generality, we may assume that each dummy sum-instance is manifested using a distinct variable in $\vec{v} = (v_1, \dots, v_k)$, whose only instance in ϕ is the related dummy sum-instance. It is straighforward to check that for any structure $\mathfrak{A}$ with cardinality n, the interpretation $t^{\mathfrak{A}}$ of any term t appearing in ϕ is at most n^k.

We start the translation $\psi \mapsto \psi^*$ by scaling each function f occurring in ϕ by $\frac{1}{n^k}$ as follows. Define $f(\vec{x}) \mapsto f^*(\vec{x}, \vec{v})$. For Boolean connectives, $=$, SUM, and first-order quantification the translation is homomorphic. In the case for existential function quantification, the functions are scaled by increasing their arity by k and stipulating that their weights are distributed evenly over the arity extension:

$$\exists f \psi \mapsto \exists f^* (\forall \vec{x} \vec{v} \vec{w} \, f^*(\vec{x}, \vec{v}) = f^*(\vec{x}, \vec{w}) \wedge \phi^*).$$

Let $f_1, \dots, f_t$ be the list of free function variables of ϕ with arities $|\vec{x}_1|, \dots, |\vec{x}_t|$, respectively. Now, define

$$\phi^+ := \exists f_1^* \dots f_t^* \Big(\bigwedge_{l \le t} (\forall \vec{x}_l \, \mathrm{SUM}_{\vec{v}} f_l^*(\vec{x}_l, \vec{v}) = f_l(\vec{x}_l) \wedge \forall \vec{x}_l \, \vec{v} \, \vec{w} \, f_l^*(\vec{x}_l, \vec{v}) = f_l^*(\vec{x}_l, \vec{w})) \wedge \exists \vec{v} \phi^* \Big).$$

It is now straightforward to check that ϕ^+ and ϕ are equivalent, and that there are no dummy-sum instances in ϕ^+. □

Lemma 20. *For every formula $\phi \in$* L-ESO$_{d[0,1]}[=, \mathrm{SUM}]$ *there is a formula $\phi^* \in$* L-$(\ddot{\exists}^*\forall^*)_{d[0,1]}[=, \mathrm{SUM}]$ *such that* $\mathrm{Struc}_{d[0,1]}(\phi) = \mathrm{Struc}_{d[0,1]}(\phi^*)$*, and any second sort identity atom in ϕ^* is of the form $f_i(\vec{w}) = \mathrm{SUM}_{\vec{v}} f_j(\vec{u}, \vec{v})$ for distinct f_i and f_j of which at least one is quantified. Furthermore, ϕ^* is almost conjunctive if ϕ is almost conjunctive and in* L-$(\ddot{\exists}^*\forall^*)_{d[0,1]}[=, \mathrm{SUM}]$.

Proof. By the previous lemma, we may assume without loss of generality that ϕ does not contain any dummy-sum instances. That is, any summation term occurring in ϕ is of the form $\mathrm{SUM}_{\vec{v}} i(\vec{u}\vec{v})$, where it is to be noted that the variables of $\vec{v}$ occur free in the term i. This, in particular, implies that the terms of ϕ can be captured by using distributions.

First we define for each second sort term $i(\vec{x})$ a special formula θ_i defined recursively using fresh function symbols f_i as follows:

- If $i(\vec{u})$ is $g(\vec{u})$ where g is a function symbol, then θ_i is defined as $f_i(\vec{u}) = g(\vec{u})$. (We may intepret $g(\vec{u})$ as $\mathrm{SUM}_{\emptyset} g(\vec{u})$).
- If $i(\vec{u})$ is $\mathrm{SUM}_{\vec{v}} j(\vec{u}\vec{v})$, then θ_i is defined as $\theta_j \wedge f_i(\vec{u}) = \mathrm{SUM}_{\vec{v}} f_j(\vec{u}\vec{v})$.

The translation $\phi \mapsto \phi^*$ then proceeds recursively on the structure of ϕ. By Lemma 15 we may use the real constant 0 in the translation.

[4] Lemma 20 was first presented in [15, Lemma 3] in a form that included multiplication. We would like to thank Richard Wilke for noting that the construction used in [15] to prove this lemma had an element that yields circularity. Furthermore, we would like to than Joni Puljujärvi for noting another issue which is circumvented by Lemma 19.

(i) If ϕ is $i(\vec{u}) = j(\vec{v})$, then ϕ^* is defined as $\exists \vec{f}(f_i(\vec{u}) = f_j(\vec{v}) \wedge \theta_i \wedge \theta_j)$ where $\vec{f}$ lists the function symbols f_k for each subterm k of i or j.

(ii) If ϕ is an atom or negated atom of the first sort, then $\phi^* := \phi$.

(iii) If ϕ is $\psi_0 \circ \psi_1$ where $\circ \in \{\vee, \wedge\}$, ψ_0^* is $\exists \vec{f_0} \forall \vec{x}_0 \theta_0$, and ψ_1^* is $\exists \vec{f_1} \forall \vec{x}_1 \theta_1$, then ϕ^* is defined as $\exists \vec{f_0} \vec{f_1} \forall \vec{x}_0 \vec{x}_1 (\theta_0 \circ \theta_1)$.

(iv) If ϕ is $\exists y \psi$ where ψ^* is $\exists \vec{f} \forall \vec{x} \theta$, then ϕ^* is defined as $\exists g \exists \vec{f} \forall \vec{x} \forall y (g(y) = 0 \vee \theta)$.

(v) Suppose ϕ is $\forall y \psi$ where ψ^* is $\exists \vec{f} \forall \vec{x} \theta$. Let $\vec{g}$ list the free distribution variables in ϕ. Then ϕ^* is defined as

$$\exists \vec{f^*} \exists \vec{g^*} \forall y y' \forall \vec{x} \Big(\bigwedge_{g^* \in \vec{g^*}} (g^*(y, \vec{x}) = g^*(y', \vec{x}) \wedge \mathrm{SUM}_y g^*(y, \vec{x}) = g(\vec{x})) \wedge \\ \bigwedge_{f^* \in \vec{f^*}} (f^*(y, \vec{x}) = f^*(y', \vec{x})) \wedge \theta^* \Big),$$

where $\vec{f^*}$ ($\vec{g^*}$, resp.) is obtained from $\vec{f}$ ($\vec{g}$, resp.) by replacing each f (g, resp.) from $\vec{f}$ ($\vec{g}$, resp.) with f^* (g^*, resp.) such that $\mathrm{ar}(f^*) = \mathrm{ar}(f) + 1$ ($\mathrm{ar}(g^*) = \mathrm{ar}(g) + 1$, resp.), and θ^* is obtained from θ by replacing all function terms $f(\vec{z})$ ($g(\vec{z})$, resp.) with $f^*(y, \vec{z})$ ($g^*(y, \vec{z})$, resp.).

(vi) If ϕ is $\exists f \psi$ where ψ^* is $\exists \vec{f} \forall \vec{x} \theta$, then ϕ^* is defined as $\exists f \psi^*$.

It is straightforward to check that ϕ^* is of the correct form and equivalent to ϕ. What happens in (v) is that instead of guessing for all y some distribution f_y with arity $\mathrm{ar}(f)$, we guess a single distribution f^* with arity $\mathrm{ar}(f) + 1$ such that $f^*(y, \vec{u}) = \frac{1}{|A|} \cdot f_y(\vec{u})$, where A is the underlying domain of the structure. Similarly, we guess a distribution g^* for each free distribution variable g such that $g^*(y, \vec{u}) = \frac{1}{|A|} \cdot g(\vec{u})$. Observe that case (iv) does not occur if ϕ is in L-$(\ddot{\exists}^* \forall^*)_{d[0,1]}$[SUM, =]; in such a case, a straightforward structural induction shows that ϕ^* is almost conjuctive if ϕ is. □

Using the obtained normal form for existential second-order logic over the reals we now proceed to the translation. This translation is similar to one found in [15], with the exception that probabilistic independence atoms cannot be used here.

Lemma 21. *Let $\phi(f) \in$ L-$(\ddot{\exists}^* \forall^*)_{d[0,1]}$[=, SUM] be of the form described in Lemma 20, with one free variable f. Then there is a formula $\Phi(\vec{x}) \in \mathrm{FO}(\approx, =(\cdots))$ such that for all structures $\mathfrak{A}$ and probabilistic teams $\mathbb{X} := f^{\mathfrak{A}}$, $\mathfrak{A} \models_{\mathbb{X}} \Phi \iff (\mathfrak{A}, f) \models \phi$. Furthermore, if $\phi(f)$ is almost conjunctive, then $\Phi(\vec{x}) \in \mathrm{FO}(\approx)$.*

Proof. By item (ii) of Proposition 12, we can use weighted semantics in this proof. Without loss of generality each structure is enriched with two distinct constants c and d; such constants are definable in $\mathrm{FO}(\approx, =(\cdots))$ by $\exists cd(=(c) \wedge =(d) \wedge c \neq d)$, and for almost conjunctive formulae they are not needed.

Let $\phi(f) = \exists \vec{f} \forall \vec{x} \theta(f, \vec{x}) \in$ L-$(\ddot{\exists}^* \forall^*)_{d[0,1]}$[=, SUM] be of the form described in the previous lemma, with one free variable f. In what follows, we build Θ inductively from θ, and then let

$$\Phi := \exists \vec{y}_1 \ldots \exists \vec{y}_n \forall \vec{x} \Theta(\vec{x}, \vec{y}_1, \ldots, \vec{y}_n),$$

where $\vec{y}_i$ are sequences of variables of length $\mathrm{ar}(f_i)$. Let $m := |\vec{x}|$. We show the following claim: For $M \subseteq A^m$ and weighted teams $\mathbb{Y} = \mathbb{X}'[M/\vec{x}]$, where the domain of $\mathbb{X}'$ extends that of $\mathbb{X}$ by $\vec{y}_1, \ldots, \vec{y}_n$,

$$\mathfrak{A} \models^w_{\mathbb{Y}} \Theta \text{ iff } (\mathfrak{A}, f, f_1, \ldots, f_n) \models \theta(\vec{a}) \text{ for all } \vec{a} \in M, \tag{6}$$

where $f_i := \mathbb{X}' \upharpoonright \vec{y}_i$. Observe that the claim implies that $\mathfrak{A} \models^w_{\mathbb{X}} \Phi$ iff $\mathfrak{A} \models \phi(f)$.

Next, we show the claim by structural induction on the construction of Θ:

(1) If θ is a literal of the first sort, we let $\Theta := \theta$, and the claim readily holds.

(2) If θ is of the form $f_i(\vec{x}_i) = \mathrm{SUM}_{\vec{x}_{j0}} f_j(\vec{x}_{j0}\vec{x}_{j1})$, let $\Theta := \exists\alpha\beta\psi$ for ψ given as

$$(\alpha = x \leftrightarrow \vec{x}_i = \vec{y}_i) \wedge (\beta = x \leftrightarrow \vec{x}_{j1} = \vec{y}_{j1}) \wedge \vec{x}\alpha \approx \vec{x}\beta, \tag{7}$$

where x is any variable from $\vec{x}$, and the first-order variable sequence $\vec{y}_j$ that corresponds to function variable f_j is thought of as a concatenation of two sequences $\vec{y}_{j0}$ and $\vec{y}_{j1}$ whose respective lenghts are $|\vec{x}_{j0}|$ and $|\vec{x}_{j1}|$.
Assume first that for all $\vec{a} \in M$, we have $(\mathfrak{A}, f, f_1, \ldots, f_n) \models \theta(\vec{a})$, that is, $f_i(\vec{a}_i) = \mathrm{SUM}_{\vec{x}_{j0}} f_j(\vec{x}_{j0}\vec{a}_{j1})$. To show that $\mathbb{Y}$ satisfies Θ, let $\mathbb{Z}$ be an extension of $\mathbb{Y}$ to variables α and β such that it satisfies the first two conjuncts of (7). Observe that $\mathbb{Z}$ satisfies $\vec{x}\alpha \approx \vec{x}\beta$ if for all $\vec{a} \in M$, $\mathbb{Z}_{\vec{x}=\vec{a}}$ satisfies $\alpha \approx \beta$. For a probabilistic team $\mathbb{X}$ and a first-order formula α, we write $|\mathbb{X}_\alpha|_{\mathrm{rel}}$ for the relative weight $|\mathbb{X}_\alpha|/|\mathbb{X}|$.
Now, the following chain of equalities hold:

$$\begin{aligned}
&|\mathbb{Z}_{\vec{x}\alpha=\vec{a}x}|_{\mathrm{rel}} = |\mathbb{Y}_{\vec{x}\vec{x}_i=\vec{a}\vec{y}_i}|_{\mathrm{rel}} = |\mathbb{Y}_{\vec{x}\vec{y}_i=\vec{a}\vec{a}_i}|_{\mathrm{rel}} = |\mathbb{Y}_{\vec{x}=\vec{a}}|_{\mathrm{rel}} \cdot |\mathbb{Y}_{\vec{y}_i=\vec{a}_i}|_{\mathrm{rel}} = \\
&|\mathbb{Y}_{\vec{x}=\vec{a}}|_{\mathrm{rel}} \cdot f_i(\vec{a}_i) = |\mathbb{Y}_{\vec{x}=\vec{a}}|_{\mathrm{rel}} \cdot \mathrm{SUM}_{\vec{x}_{j0}} f_j(\vec{x}_{j0}\vec{a}_{j1}) = |\mathbb{Y}_{\vec{x}=\vec{a}}|_{\mathrm{rel}} \cdot |\mathbb{Y}_{\vec{y}_{j1}=\vec{a}_{j1}}|_{\mathrm{rel}} \\
&|\mathbb{Y}_{\vec{x}\vec{y}_{j1}=\vec{a}\vec{a}_{j1}}|_{\mathrm{rel}} = |\mathbb{Y}_{\vec{x}\vec{x}_{j1}=\vec{a}\vec{y}_{j1}}|_{\mathrm{rel}} = |\mathbb{Z}_{\vec{x}\beta=\vec{a}x}|_{\mathrm{rel}}.
\end{aligned}$$

Note that the absolute weights $|\mathbb{Y}|$ and $|\mathbb{Z}|$ are equal. The third equality then follows since $\vec{x}$ and $\vec{y}_i$ are independent by the construction of $\mathbb{Y}$. It is also here that we need relative instead of absolute weights. Thus α and β agree with x in $\mathbb{Z}_{\vec{x}=\vec{a}}$ for the same weight. Moreover, x is some constant a in $\mathbb{Z}_{\vec{x}=\vec{a}}$, and whenever α or β disagrees with x, it can be mapped to another constant b that is distinct from a. It follows that $\mathbb{Z}_{\vec{x}=\vec{a}}$ satisfies $\alpha \approx \beta$, and thus we conclude that $\mathbb{Y}$ satisfies Θ.
For the converse direction, assume that $\mathbb{Y}$ satisfies Θ, and let $\mathbb{Z}$ be an extension of $\mathbb{Y}$ to α and β satisfying (7). Then for all $\vec{a} \in M$, $\mathbb{Z}_{\vec{x}=\vec{a}}$ satisfies $\alpha \approx \beta$ and thereby for all $\vec{a} \in M$,

$$|\mathbb{Y}_{\vec{x}=\vec{a}}|_{\mathrm{rel}} \cdot f_i(\vec{a}_i) = |\mathbb{Z}_{\vec{x}\alpha=\vec{a}x}|_{\mathrm{rel}} = |\mathbb{Z}_{\vec{x}\beta=\vec{a}x}|_{\mathrm{rel}} = |\mathbb{Y}_{\vec{x}=\vec{a}}|_{\mathrm{rel}} \cdot \mathrm{SUM}_{\vec{x}_k} f_j(\vec{x}_k, \vec{a}_l).$$

For the second equality, recall that x is a constant in $\mathbb{Z}_{\vec{x}=\vec{a}}$. Thus $(\mathfrak{A}, f, f_1, \ldots, f_n) \models \theta(\vec{a})$ for all $\vec{a} \in M$, which concludes the induction step.

(3) If θ is $\theta_0 \wedge \theta_1$, let $\Theta := \Theta_0 \wedge \Theta_1$. The claim follows by the induction hypothesis.

(4) If θ is $\theta_0 \vee \theta_1$, let $\Theta := \exists z\big(=\!(\vec{x}, z) \wedge ((\Theta_0 \wedge z = c) \vee (\Theta_1 \wedge z = d))\big)$.
Alternatively, if θ_0 contains no numerical terms, let $\Theta := \theta_0 \vee (\theta_0^{\neg} \wedge \Theta_1)$, where $\theta_0^{\neg}$ is obtained from $\neg\theta_0$ by pushing $\neg$ in front of atomic formulae.
Assume first that $(\mathfrak{A}, f, f_1, \ldots, f_n) \models \theta_0 \vee \theta_1$ for all $\vec{a} \in M$. Then M can be partitioned to disjoint M_0 and M_1 such that

$$(\mathfrak{A}, f, f_1, \ldots, f_n) \models \theta_i \text{ for all } \vec{a} \in M_i. \tag{8}$$

We have two cases:

- Suppose $\phi(f)$ is not almost conjunctive. Let $\mathbb{Z}$ be the extension of $\mathbb{Y}$ to z such that $s(z) = c$ if $s(\vec{x})$ is in M_0, and otherwise $s(z) = d$, where s is any assignment in the support of $\mathbb{Z}$. Consequently, $\mathbb{Z}$ satisfies $=\!(\vec{x}, z)$. Further, the induction hypothesis implies that $\mathfrak{A} \models^w_{\mathbb{Y}_i} \Theta_i$, where $\mathbb{Y}_i := X'[M_i/\vec{x}]$. Since $\frac{|M_0|}{|M|}\mathbb{Y}_0 = \mathbb{Z}_{\vec{z}=c}$ and $\frac{|M_1|}{|M|}\mathbb{Y}_1 = \mathbb{Z}_{\vec{z}=d}$, we obtain $\mathfrak{A} \models^w_{\mathbb{Z}_{\vec{z}=c}} \theta_0$ and $\mathfrak{A} \models^w_{\mathbb{Z}_{\vec{z}=d}} \Theta_1$ by item (iii) of Proposition 12. We conclude that $\mathbb{Z}$ satisfies $(\Theta_0 \wedge z = 0) \vee (\Theta_1 \wedge z = 1)$, and thus $\mathbb{Y}$ satisfies Θ.
- Suppose $\phi(f)$ is almost conjunctive. Without loss of generality θ_0 contains no numerical terms. Then $\mathfrak{A} \models_{\mathbb{X}'[M_0/\vec{x}]} \theta_0$ by flatness (i.e., (v) of Proposition 12). We may assume that M_0 is the maximal subset of M satisfying (8), in which case we also obtain $\mathfrak{A} \models_{\mathbb{X}'[M_1/\vec{x}]} \theta_0^{\neg}$ by flatness. Furthermore, $\mathfrak{A} \models_{\mathbb{X}'[M_1/\vec{x}]} \Theta_1$ by induction hypothesis.

The converse direction is shown analogously in both cases. This concludes the proof.

□

The "≥" direction of item (i) in Theorem 13 follows by Lemmata 14, 20, and 21; that of item (ii) follows similarly, except that Proposition 16 is used instead of Lemma 14. This concludes the proof of Theorem 13.

5. Interpreting inclusion logic in probabilistic team semantics

Next we turn to the relationship between inclusion and probabilistic inclusion logics. The logics are comparable for, as shown in Propositions 12, team semantics embeds into probabilistic team semantics conservatively. The seminal result by Galliani and Hella shows that inclusion logic captures PTIME over ordered structures [18]. We show that restricting to finite structures, or uniformly distributed probabilistic teams, inclusion logic is in turn subsumed by probabilistic inclusion logic. There are two immediate consequences for this. First, the result by Galliani and Hella readily extends to probabilistic inclusion logic. Second, their result obtains an alternative, entirely different proof through linear systems.

We utilize another result of Galliani stating that inclusion logic is equiexpressive with *equiextension logic* [17], defined as the extension of first-order logic with *equiextension* atoms $\vec{x}_1 \bowtie \vec{x}_2 := \vec{x}_1 \subseteq \vec{x}_2 \wedge \vec{x}_2 \subseteq \vec{x}_1$. In the sequel, we relate equiextension atoms to probabilistic inclusion atoms.

For a natural number $k \in \mathbb{N}$ and an equiextension atom $\vec{x}_1 \bowtie \vec{x}_2$, where $\vec{x}_1$ and $\vec{x}_2$ are variable tuples of length m, define $\psi^k(\vec{x}_1, \vec{x}_2)$ as

$$\begin{aligned}\forall \vec{u} \exists v_1 v_2 \forall \vec{z_0} \exists \vec{z}((\vec{x}_1 = \vec{u} \leftrightarrow v_1 = y) \wedge (\vec{x}_2 = \vec{u} \leftrightarrow v_2 = y) \wedge \\ (\vec{z_0} = \vec{y} \rightarrow \vec{z} = \vec{y}) \wedge (\neg \vec{z} = \vec{y} \vee \vec{u} v_1 \approx \vec{u} v_2)),\end{aligned} \tag{9}$$

where $\vec{z}$ and $\vec{z_0}$ are variable tuples of length k, and $\vec{y}$ is obtained by concatenating k times some variable y in $\vec{u}$. Intuitively (9) expresses that a probabilistic team $\mathbb{X}$, extended with universally quantified $\vec{u}$, decomposes to $\mathbb{Y} + \mathbb{Z}$, where $\mathbb{Y}(s) = f_s \mathbb{X}(s)$ for some variable coefficient $f_s \in [\frac{1}{n^k}, 1]$, and $|\mathbb{Y}_{\vec{x}_1 = \vec{u}}| = |\mathbb{Y}_{\vec{x}_2 = \vec{u}}|$, for any $\vec{u}$. Thus (9) implies that $\vec{x}_1 \bowtie \vec{x}_2$. On the other hand, $\vec{x}_1 \bowtie \vec{x}_2$ implies (9) if each assignment weight $\mathbb{X}(s)$ equals $g_s |\mathbb{X}|$ for some $g_s \in [\frac{1}{n^k}, 1]$. In this case, one finds the decomposition $\mathbb{Y} + \mathbb{Z}$ by balancing the weight differences between values of $\vec{x}_1$ and $\vec{x}_2$. More details are provided in the proof of the next lemma.

Lemma 22. *Let k be a positive integer, $\mathfrak{A}$ a finite structure with universe A of size n, and $\mathbb{X} : X \rightarrow \mathbb{R}_{\geq 0}$ a weighted team.*

(i) Suppose $\mathfrak{A} \models^w_{\mathbb{X}} \vec{x}_1 \bowtie \vec{x}_2$, $|\mathbb{X}_{\vec{x}_1 = \vec{x}_2}| = 0$, and $\mathbb{X}(s) \geq \frac{|\mathbb{X}|}{n^k}$ for all $s \in$ Supp$(\mathbb{X})$. Then $\mathfrak{A} \models^w_{\mathbb{X}} \phi^k(\vec{x}, \vec{y})$.

(ii) If $\mathfrak{A} \models^w_{\mathbb{X}} \phi^k(\vec{x}, \vec{y})$, then $\mathfrak{A} \models^w_{\mathbb{X}} \vec{x}_1 \bowtie \vec{x}_2$.

Proof. (i) Observe that $\mathbb{X}[A/\vec{u}] = \frac{1}{n^m}\mathbb{X}^*$, where $\mathbb{X}^*$ is defined as the sum $\mathbb{X}[\vec{a}_1/\vec{u}] + \ldots + \mathbb{X}[\vec{a}_l/\vec{u}]$, and $\vec{a}_1, \ldots, \vec{a}_l$ lists all elements in A^m. By Proposition 12(iii) it suffices to show that $\mathbb{X}^*$ satisfies the formula obtained by removing the outermost universal quantification of ψ^k. By Proposition 29 it suffices to show that each $\mathbb{X}[\vec{a}_i/\vec{u}]$ individually satisfies the same formula. Hence fix a tuple of values $\vec{b} \in A^m$ and define $\mathbb{Y} := \mathbb{X}[\vec{b}/\vec{u}]$. We show that $\mathbb{Y}$ satisfies

$$\begin{aligned}\exists v_1 v_2 \forall \vec{z_0} \exists \vec{z_1}((\vec{x}_1 = \vec{b} \leftrightarrow v_1 = c) \wedge (\vec{x}_2 = \vec{b} \leftrightarrow v_2 = c) \wedge \\ (\vec{z_0} = \vec{c} \rightarrow \vec{z_1} = \vec{c}) \wedge (\vec{z_1} = \vec{c} \rightarrow v_1 \approx v_2)).\end{aligned} \tag{10}$$

Observe that we have here fixed $\vec{u} \mapsto \vec{b}$ and $y \mapsto c$, where c is some value in $\vec{b}$. We have also removed $\vec{u}$ from the marginal identity atom in (9), for it has a fixed value in $\mathbb{Y}$.

Fix some $d \in A$ that is distinct from c, and denote by Y be the support of $\mathbb{Y}$. For existential quantification over v_i, extend $s \in Y$ by $v_i \mapsto c$ if $s(\vec{x}_i) = \vec{b}$, and otherwise by $v_i \mapsto d$, so as to satisfy the first two conjuncts. Denote by $\mathbb{Y}' : Y' \rightarrow \mathbb{R}_{\geq 0}$ the weighted team, where Y' consists of these extensions, and the weights are inherited from $\mathbb{Y}$.

Observe that $\mathbb{Y}'(s) \geq \frac{|\mathbb{X}|}{n^k}$ for all $s \in$ Supp$(\mathbb{Y}')$. Fix $i \in \{1, 2\}$, and assume that $|\mathbb{X}_{\vec{x}_i = \vec{b}}| > 0$. Then $|\mathbb{X}_{\vec{x}_i = \vec{b}}| \geq \frac{|\mathbb{X}|}{n^k}$, and thus using $|\mathbb{X}_{\vec{x}_1 = \vec{x}_2}| = 0$ and $|\mathbb{X}| = |\mathbb{Y}'|$ we obtain

$$w_i := |\mathbb{Y}'_{v_i = c \wedge v_{3-i} = d}| = |\mathbb{X}_{\vec{x}_i = \vec{b} \wedge \vec{x}_{3-i} \neq \vec{b}}| = |\mathbb{X}_{\vec{x}_i = \vec{b}}| \geq \frac{|\mathbb{Y}'|}{n^k}.$$

Since $\mathbb{X} \models \vec{x}_1 \bowtie \vec{x}_2$, we obtain that w_1 and w_2 are either both zero or both at least $\frac{|\mathbb{Y}'|}{n^k}$.

Next, let us describe the existential quantification of $\vec{z_1}$ (later we show how the universal quantification of $\vec{z_0}$ can be fitted in). The purpose of this step is to balance the possible weight difference between $|\mathbb{Y}'_{\vec{x}_1 = \vec{b}}|$ and $|\mathbb{Y}'_{\vec{x}_2 = \vec{b}}|$, which in turn is tantamount to balancing $|\mathbb{Y}'_{\vec{v}_1 = c \wedge v_2 = d}|$ and $|\mathbb{Y}'_{v_1 = d \wedge v_2 = c}|$. For $s' \in Y'$,

(i) if $s'(v_1) = c$ and $s'(v_2) = d$, allocate respectively $\frac{w_2}{|\mathbb{Y}'|}$ and $1 - \frac{w_2}{|\mathbb{Y}'|}$ of the weight of s' to $s'(\vec{c}/\vec{z_1})$ and $s'(\vec{d}/\vec{z_1})$;

(ii) if $s'(v_1) = d$ and $s'(v_2) = c$, allocate respectively $\frac{w_1}{|\mathbb{Y}'|}$ and $1 - \frac{w_1}{|\mathbb{Y}'|}$ of the weight of s' to $s'(\vec{c}/\vec{z_1})$ and $s'(\vec{d}/\vec{z_1})$; or

(iii) otherwise, allocate the full weight of s' to $s'(\vec{c}/\vec{z_1})$.

Denote by $\mathbb{Z}$ the probabilistic team obtained this way, and define $\mathbb{Z}' := \mathbb{Z}_{\vec{z_1}=\vec{c}}$. We observe that

$$|\mathbb{Z}'_{v_1=c\wedge v_2=d}| = |\mathbb{Z}'_{v_1=d\wedge v_2=c}| = \frac{w_1 w_2}{|\mathbb{Y}'|}.$$

Furthermore, $|\mathbb{Z}'_{v_1=c\wedge v_2=c}| = 0$ and hence $|\mathbb{Z}'_{v_1=d\wedge v_2=d}| = |\mathbb{Z}'| - \frac{2w_1w_2}{|\mathbb{Y}'|}$. We conclude that $\mathbb{Z}'$ satisfies $v_1 \approx v_2$, whence $\mathbb{Z}$ satisfies $\vec{z_1} = \vec{c} \to v_1 \approx v_2$.

Finally, let us return to the universal quantification of $\vec{z_0}$, which precedes the existential quantification of $\vec{z}$ in (10). The purpose of this step is to enforce that for each $s \in \mathrm{Supp}(\mathbb{Y}')$, the extension $s(\vec{c}/\vec{z_1})$ takes a positive weight. Observe that $\frac{w_i}{|\mathbb{Y}'|}$ is either zero or at least $\frac{1}{n^k}$, for w_i is either zero or at least $\frac{|\mathbb{Y}'|}{n^k}$. Furthermore, note that universal quantification distributes $\frac{1}{n^k}$ of the weight of s' to $s'(\vec{c}/\vec{z_0})$. Thus the weight of s' can be distributed in such a way that both the conditions (i)-(iii) and the formula $\vec{z_0} = \vec{c} \to \vec{z_1} = \vec{c}$ simultaneously hold. This concludes the proof of case (i).

(ii) Suppose that the assignments in X mapping $\vec{x_1}$ to $\vec{b}$ have a positive total weight in $\mathbb{X}$. By symmetry, it suffices to show that the assignments in X mapping $\vec{x_2}$ to $\vec{b}$ also have a positive total weight in $\mathbb{X}$. By assumption there is an extension $\mathbb{Z}$ of $\mathbb{X}[\vec{b}/\vec{u}]$ satisfying the quantifier-free part of (10). It follows that the total weight of assignments in $\mathbb{Z}$ that map v_1 to c is positive. Consequently, by $\vec{z_0} = \vec{c} \to \vec{z_1} = \vec{c}$ where $\vec{z_0}$ is universally quantified, a positive fraction of these assignments maps also $\vec{z_1}$ to $\vec{c}$. This part of $\mathbb{Z}$ is allocated to $v_1 \approx v_2$, and thus the weights of assignments mapping v_2 to c is positive as well. But then, going backwards, we conclude that the total weight of assignments mapping $\vec{x_2}$ to $\vec{b}$ is positive, which concludes the proof. □

We next establish that inclusion logic is subsumed by probabilistic inclusion logic at the level of sentences.

Theorem 23. $\mathrm{FO}(\subseteq) \leq \mathrm{FO}(\approx)$ *with respect to sentences.*

Proof. As $\mathrm{FO}(\subseteq) \equiv \mathrm{FO}(\bowtie)$ ([17]), it suffices to show $\mathrm{FO}(\bowtie) \leq \mathrm{FO}(\approx)$ over sentences. Let $\phi \in \mathrm{FO}(\bowtie)$ be a sentence, and let k be the number of disjunctions and quantifiers in ϕ. Let ϕ^* be obtained from ϕ by replacing all equiextension atoms of the form $\vec{x_1} \bowtie \vec{x_2}$ with $\psi^k(\vec{x_1}, \vec{x_2})$. We can make four simplifying assumption without loss of generality. First, we may restrict attention to weighted semantics by item (ii) of Proposition 12. Thus, we assume that $\mathfrak{A} \models^w_{\mathbb{X}} \phi$ for some weighted team $\mathbb{X}$ and a finite structure $\mathfrak{A}$ with universe of size n. Second, we may assume that the support of $\mathbb{X}$ consists of the empty assignment by item (iv) of Proposition 12. Third, since $\mathrm{FO}(\bowtie)$ is insensitive to assignment weights, we may assume that the satisfaction of ϕ by $\mathbb{X}$ is witnessed by uniform semantic operations. That is, existential and universal quantification split an assignment to at most n equally weighted extensions, and disjunction can only split an assignment to two equally weighted parts. Fourth, we may assume that any equiextension atom $\vec{x_1} \bowtie \vec{x_2}$ appears in ϕ in an equivalent form $\exists uv(u \neq v \wedge \vec{x_1}u \bowtie \vec{x_2}v)$, to guarantee that the condition $|\mathbb{X}_{\vec{x_1}=\vec{x_2}}| = 0$ holds for all appropriate subteams $\mathbb{X}$. We then obtain by the previous lemma and a simple inductive argument that $\mathfrak{A} \models^w_{\mathbb{X}} \phi^*$. The converse direction follows similarly by the previous lemma. □

Consequently, probabilistic inclusion logic captures P, for this holds already for inclusion logic [18]. Another consequence is an alternative proof, through probabilistic inclusion logic (Theorem 23) and linear programs (Theorems 13 and 4), for the PTIME upper bound of the data complexity of inclusion logic. For this, note also that quantification of functions, whose range is the unit interval, is clearly expressible in $\mathrm{ESO}_{\mathbb{R}}[\leq, \mathrm{SUM}, 0, 1]$.

Corollary 24. *Sentences of* $\mathrm{FO}(\approx)$ *capture* P *on finite ordered structures.*

Theorem 23 also extends to formulae over uniform teams. Recall that a function f is uniform if $f(s) = f(s')$ for all $s, s' \in \mathrm{Supp}(f)$.

Theorem 25. $\mathrm{FO}(\subseteq) \leq \mathrm{FO}(\approx)$ *over uniform probabilistic teams.*

Proof. Recall that $\mathrm{FO}(\subseteq) \equiv \mathrm{FO}(\bowtie)$. Let ϕ be an $\mathrm{FO}(\bowtie)$ formula, $\mathfrak{A}$ a finite structure, and $\mathbb{X}$ a uniform probabilistic team. Let * denote the translation of Theorem 23. Now

$$\begin{aligned}
\mathfrak{A} \models_{\mathbb{X}} \phi \quad &\Leftrightarrow \quad (\mathfrak{A}, R := X) \models \forall x_1 \ldots x_n(\neg R(x_1 \ldots x_n) \vee (R(x_1 \ldots x_n) \wedge \phi)) \\
&\Leftrightarrow \quad (\mathfrak{A}, R := X) \models \forall x_1 \ldots x_n(\neg R(x_1 \ldots x_n) \vee (R(x_1 \ldots x_n) \wedge \phi))^* \\
&\Leftrightarrow \quad (\mathfrak{A}, R := X) \models \forall x_1 \ldots x_n(\neg R(x_1 \ldots x_n) \vee (R(x_1 \ldots x_n) \wedge \phi^*)) \\
&\Leftrightarrow \quad \mathfrak{A} \models_{\mathbb{X}} \phi^*,
\end{aligned}$$

where X is the support of $\mathbb{X}$ and $\mathrm{Dom}(\mathbb{X}) = \{x_1, \ldots, x_n\}$.

□

6. Definability over open formulae

We now turn to definability over open formulae. In team semantics, inclusion logic extended with dependence atoms is expressively equivalent to independence logic at the level of formulae. This relationship however does not extend to probabilistic team semantics. As we will prove next, probabilistic inclusion logic extended with dependence atoms is strictly less expressive than probabilistic independence logic. The reason, in short, is that logics with marginal identity and dependence can only describe additive distribution properties, whereas the concept of independence involves multiplication.

We begin with a proposition illustrating that probabilistic independence logic has access to irrational weights.[5]

Proposition 26. *Define $\phi(x) = \exists c \exists y \forall z \theta$, where θ is defined as*

$$=(c) \wedge x \perp\!\!\!\perp y \wedge x \approx y \wedge ((x = c \wedge y = c) \leftrightarrow z = c). \tag{11}$$

Let $\mathfrak{A}$ be a finite structure with domain A of size n, and let $\mathbb{X}$ be a probabilistic team. Then

$$\mathfrak{A} \models_{\mathbb{X}} \phi(x) \implies |\mathbb{X}_{x=a}| = \frac{1}{\sqrt{n}} \textit{ for some } a \in A. \tag{12}$$

Proof. Suppose $\mathfrak{A} \models_{\mathbb{X}} \phi(x)$, and let $\mathbb{Y}$ be an extension of $\mathbb{X}$, in accord with the quantifier prefix of ϕ, that satisfies (11). Then in $\mathbb{Y}$ c is constant and z uniformly distributed over all domain values. Hence z equals c for weight $\frac{1}{n}$, and consequently x and y simultaneously equal c for the same weight. Since x and y are independent and identically distributed, in isolation they equal c for weight $\frac{1}{\sqrt{n}}$. Since $\mathbb{X}$ and $\mathbb{Y}$ agree on the weights of x, the claim follows. □

It follows, then, that independence atoms are not definable in additive existential second-order logic.

Lemma 27. $\mathrm{FO}(\perp\!\!\!\perp) \not\leq \mathrm{ESO}_{\mathbb{R}}[\leq, +, 0, 1]$.

Proof. Let $\phi(x)$ be as in the previous proposition. Assume towards contradiction that it has a translation $\Psi(f)$ in $\mathrm{ESO}_{\mathbb{R}}[\leq, +, 0, 1]$. Then Ψ contains one free unary function variable f to encode the probabilistic team over $\{x\}$. Let $\mathfrak{A}$ be a structure with universe $\{0, 1\}$ and empty vocabulary. By the previous proposition $\mathfrak{A}$ satisfies $\Psi(f)$ if and only if $\{f(0), f(1)\} = \{1/\sqrt{2}, 1 - 1/\sqrt{2}\}$.

We define a translation $\Phi \mapsto \Phi^*$ from $\mathrm{ESO}_{\mathbb{R}}[\leq, +, 0, 1]$ over $\mathfrak{A}$ to the additive existential (first-order) theory over the reals. Without loss of generality Φ has no nested function terms. In the translation, we interpret function terms of the form $g(a_1, \ldots, a_{\mathrm{ar}(g)})$, for $a_1, \ldots, a_{\mathrm{ar}(g)} \in \{0, 1\}$, as first-order variables. The translation, defined recursively, is identity for numerical inequality atoms, homomorphic for disjunction and conjunction, and otherwise defined as:

- $(\forall y \Phi)^* := \Phi^*(0/y) \wedge \Phi^*(1/y)$,
- $(\exists y \Phi)^* := \Psi^*(0/y) \vee \Phi^*(1/y)$,

[5] We thank Vadim Kulikov for the idea behind this proposition.

- $(\exists g\Phi)^* := (\exists g(a_1, \ldots, a_{\mathrm{ar}(g)})_{a_1,\ldots,a_{\mathrm{ar}(g)}\in\{0,1\}}\Phi^*$,

where $\Phi^*(a/y)$ is obtained from Φ^* by substituting variable $h(x_1, \ldots, x_{i-1}, a, x_{i+1}, \ldots x_n)$ for any variable of the form $h(x_1, \ldots, x_{i-1}, y, x_{i+1}, \ldots x_n)$. Applying the translation to $\Psi(f)$ we obtain a formula $\Psi^*(f(0), f(1))$ that contains two free first-order variables $f(0)$ and $f(1)$.

It is easy to see that $\mathfrak{A} \models \Psi(f)$ if and only if $\Psi^*(f(0), f(1))$ holds in the real arithmetic. Consequently, Ψ^* has only irrational solutions. On the other hand, Ψ^* can be transformed to the form $\exists x_1 \ldots \exists x_n \bigvee_i \bigwedge_j C_{ij}$, where each C_{ij} is a (strict or non-strict) linear inequation with integer coefficients and constants. Since Ψ^* is satisfiable, some system of linear inequations $\bigwedge_j C_{ij}$ has solutions, and thus also rational solutions. [6] Thus Ψ^* has rational solutions, which leads to a contradiction. We conclude that $\phi(x)$ does not translate into $\mathrm{ESO}_{\mathbb{R}}[\leq, +, 0, 1]$. □

The following result is now immediate.

Theorem 28. $\mathrm{FO}(=(\cdots), \approx) < \mathrm{FO}(\perp\!\!\!\perp)$.

Proof. Dependence and marginal identity atoms are definable in $\mathrm{FO}(\perp\!\!\!\perp)$ (i.e., in first-order logic extended with marginal probabilistic independence atoms) [25, Proposition 3, Theorem 10, and Theorem 11]. Furthermore, $\phi(x)$ in Proposition 26 is not definable in $\mathrm{FO}(=(\cdots), \approx)$. For this, recall that by Theorem 13, $\mathrm{FO}(=(\cdots), \approx)$ corresponds to $\text{L-ESO}_{[0,1]}[\leq, +, 0, 1]$. This logic is clearly subsumed by $\mathrm{ESO}_{\mathbb{R}}[\leq, +, 0, 1]$, which in turn cannot translate $\phi(x)$ by the previous lemma. □

There are, in fact, more than one way to prove that $\mathrm{FO}(\perp\!\!\!\perp) \not\leq \mathrm{FO}(=(\cdots), \approx)$. Above, we use the fact that probabilistic independence cannot be defined in terms of additive existential second-order logic, which in turn encompasses both dependence and marginal independence atoms. Another strategy is to apply the closure properties of these atoms.

Let ϕ be a formula over probabilistic team semantics. We say that ϕ is *closed under scaled unions* if for all parameters $\alpha \in [0, 1]$, finite structures $\mathfrak{A}$, and probabilistic teams $\mathbb{X}$ and $\mathbb{Y}$: $\mathfrak{A} \models_{\mathbb{X}} \phi$ and $\mathfrak{A} \models_{\mathbb{Y}} \phi$ imply $\mathfrak{A} \models_{\mathbb{Z}} \phi$, where $\mathbb{Z} := \alpha\mathbb{X} + (1 - \alpha)\mathbb{Y}$. In the weighted semantics, we say that ϕ is *closed under unions* if for all finite structures $\mathfrak{A}$ and weighted teams $\mathbb{X}$ and $\mathbb{Y}$: $\mathfrak{A} \models^w_{\mathbb{X}} \phi$ and $\mathfrak{A} \models^w_{\mathbb{Y}} \phi$ imply $\mathfrak{A} \models^w_{\mathbb{X}+\mathbb{Y}} \phi$. We say that ϕ is *relational* if for all finite structures $\mathfrak{A}$, and probabilistic teams $\mathbb{X}$ and $\mathbb{Y}$ such that $\mathrm{Supp}(Y) = \mathrm{Supp}(X)$: $\mathfrak{A} \models_{\mathbb{X}} \phi$ if and only if $\mathfrak{A} \models_{\mathbb{Y}} \phi$. We say that ϕ is *downwards closed* if for all finite structures $\mathfrak{A}$, and probabilistic teams $\mathbb{X}$ and $\mathbb{Y}$ such that $\mathrm{Supp}(Y) \subseteq \mathrm{Supp}(X)$: $\mathfrak{A} \models_{\mathbb{X}} \phi$ implies $\mathfrak{A} \models_{\mathbb{Y}} \phi$. Furthermore, a logic $\mathcal{L}$ is called relational (downward closed, closed under scaled union, resp.) if each formula ϕ in $\mathcal{L}$ is relational (downward closed, closed under scaled unions, resp.).

Proposition 29. *The following properties hold:*

- $\mathrm{FO}(=(\cdots))$ *is relational. [Self-evident]*
- $\mathrm{FO}(\approx)$ *is closed under scaled unions. [25]*

In the context of multiteam semantics, Grädel and Wilke have shown that probabilistic independence is not definable by any logic that extends first-order logic with a collection of atoms that are downwards closed or union closed [23, Theorem 17]. In fact, their proof works also when downwards closed atoms are replaced with relational atoms (which, in their framework as well as in the probabilistic framework, is a strictly more general notion). While their proof technique does not directly generalise to probabilistic team semantics, it can readily be adapted to weighted semantics (Definition 10).

Theorem 30 (cf. [23])**.** *Let C be a collection of relational atoms, and let $\mathcal{D}$ be a collection of atoms that are closed under unions. Then under weighted semantics* $\mathrm{FO}(\perp\!\!\!\perp) \not\leq \mathrm{FO}(C, \mathcal{D})$.

[6] To see why, observe that such a system can be expressed as a linear program in the canonical form (e.g., as in the proof of Theorem 8). Since the optimal solution of a linear program is always attained at a vertex of the feasible region, a linear program with rational coefficients and constants has at least one rational optimal solution if it has optimal solutions at all (see, e.g., [13]).

This theorem can be then transferred to probabilistic semantics by using the following observations: For any probabilistic n-ary atom D, we can define an n-ary atom D^* in the weighted semantics as follows:

$$\mathfrak{A} \models^w_{\mathbb{X}} \mathrm{D}^*(x_1, \ldots, x_n) \text{ if and only if } \mathfrak{A} \models_{\frac{1}{|\mathbb{X}|}\cdot\mathbb{X}} \mathrm{D}(x_1, \ldots, x_n)$$

It follows via a straightforward calculation that D^* is union closed, whenever D is closed under scaled unions: Assume that $\mathfrak{A} \models^w_{\mathbb{X}} \mathrm{D}^*(x_1, \ldots, x_n)$ and $\mathfrak{A} \models^w_{\mathbb{Y}} \mathrm{D}^*(x_1, \ldots, x_n)$. Fix $k = \frac{|\mathbb{X}|}{|\mathbb{X}|+|\mathbb{Y}|}$ and note that then $1 - k = \frac{|\mathbb{Y}|}{|\mathbb{X}|+|\mathbb{Y}|}$. By definition, we get $\mathfrak{A} \models_{\frac{1}{|\mathbb{X}|}\cdot\mathbb{X}} \mathrm{D}(x_1, \ldots, x_n)$ and $\mathfrak{A} \models_{\frac{1}{|\mathbb{Y}|}\cdot\mathbb{Y}} \mathrm{D}(x_1, \ldots, x_n)$, from which $\mathfrak{A} \models_{\frac{k}{|\mathbb{X}|}\cdot\mathbb{X}+\frac{1-k}{|\mathbb{Y}|}\cdot\mathbb{Y}} \mathrm{D}(x_1, \ldots, x_n)$ follows via closure under scaled unions. Finally, since $\frac{k}{|\mathbb{X}|} \cdot \mathbb{X} + \frac{1-k}{|\mathbb{Y}|} \cdot \mathbb{Y} = \frac{1}{|\mathbb{X}|+|\mathbb{Y}|} \cdot \mathbb{X} + \frac{1}{|\mathbb{X}|+|\mathbb{Y}|} \cdot \mathbb{Y} = \frac{1}{|\mathbb{X}|+|\mathbb{Y}|} \cdot (\mathbb{X} + \mathbb{Y})$, we obtain that $\mathfrak{A} \models^w_{\mathbb{X}+\mathbb{Y}} \mathrm{D}^*(x_1, \ldots, x_n)$.

The final piece of the puzzle is the following generalisation of [25, Proposition 8]. The original proposition was formulated for concrete atomic dependency statements satisfying the proposition as an atomic case for induction. The inductive argument of the original proof works with any collection of atoms that satisfy the proposition as an atomic case.

Proposition 31. *Let* D *be a collection of atoms. If* $\mathfrak{A} \models^w_{\mathbb{X}} \mathrm{D}(\vec{x}) \Leftrightarrow \mathfrak{A} \models_{\frac{1}{|\mathbb{X}|}\cdot\mathbb{X}} \mathrm{D}(\vec{x})$, *for every structure* $\mathfrak{A}$, *weighted team* $\mathbb{X} : X \to \mathbb{R}_{\geq 0}$ *of* $\mathfrak{A}$, *and* $D \in \mathrm{D}$, *then* $\mathfrak{A} \models^w_{\mathbb{X}} \phi \Leftrightarrow \mathfrak{A} \models_{\frac{1}{|\mathbb{X}|}\cdot\mathbb{X}} \phi$, *for every* $\mathfrak{A}$, $\mathbb{X}$, *and* $\phi \in \mathrm{FO}(\mathrm{D})$ *as well.*

By combining Theorem 30 and Proposition 31 with the two observation made above, we obtain the probabilistic analogue of Theorem 30.

Theorem 32. *Let* C *be a collection of relational atoms, and let* $\mathcal{D}$ *be a collection of atoms that are closed under scaled unions. Then* $\mathrm{FO}(\perp\!\!\!\perp) \not\leq \mathrm{FO}(C, \mathcal{D})$.

From this, $\mathrm{FO}(\perp\!\!\!\perp) \not\leq \mathrm{FO}(=(\cdots), \approx)$ follows as a special case by Proposition 29.

7. Axiomatization of marginal identity atoms

Next we turn to axioms of the marginal identity atom, restricting attention to atoms of the form $x_1 \ldots x_n \approx y_1 \ldots y_n$, where both $x_1 \ldots x_n$ and $y_1 \ldots y_n$ are sequences of distinct variables. It turns out that the axioms of inclusion dependencies over relational databases [9] are sound and almost complete for marginal identity; we only need one additional rule for symmetricity. Consider the following axiomatization:

1. reflexivity: $x_1 \ldots x_n \approx x_1 \ldots x_n$;
2. symmetry: if $x_1 \ldots x_n \approx y_1 \ldots y_n$, then $y_1 \ldots y_n \approx x_1 \ldots x_n$;
3. projection and permutation: if $x_1 \ldots x_n \approx y_1 \ldots y_n$, then $x_{i_1} \ldots x_{i_k} \approx y_{i_1} \ldots y_{i_k}$, where $i_1, \ldots, i_k$ is a sequence of distinct integers from $\{1, \ldots, n\}$.
4. transitivity: if $x_1 \ldots x_n \approx y_1 \ldots y_n$ and $y_1 \ldots y_n \approx z_1 \ldots z_n$, then $x_1 \ldots x_n \approx z_1 \ldots z_n$.

For a set of marginal identity atoms $\Sigma \cup \{\sigma\}$, a proof of σ from Σ is a finite sequence of marginal identity atoms such that (i) each element of the sequence is either from Σ, or follows from previous atoms in the sequence by an application of a rule, and (ii) the last element in the sequence is σ. We write $\Sigma \vdash \sigma$ if there is a proof of σ from Σ. For a probabilistic team $\mathbb{X}$ and a formula ϕ over the empty vocabulary $\tau_\emptyset$, we write $\mathbb{X} \models \phi$ as a shorthand for $\mathfrak{A} \models_{\mathbb{X}} \phi$, where $\mathfrak{A}$ is the structure over $\tau_\emptyset$ whose domain consists of the values in the support of $\mathbb{X}$. We use a shorthand $X \models \phi$, for a team X, analogously. We write $\Sigma \models \sigma$ if every probabilistic team $\mathbb{Y}$ that satisfies Σ satisfies also σ. The proof of the following theorem is an adaptation of a similar result for inclusion dependencies [9].

Theorem 33. *Let* $\Sigma \cup \{\sigma\}$ *be a finite set of marginal identity atoms. Then* $\Sigma \models \sigma$ *if and only if* $\Sigma \vdash \sigma$.

Proof. It is clear that the axiomatization is sound; we show that it is also complete.

Assume that $\Sigma \models \sigma$, where σ is of the form $x_1 \ldots x_n \approx y_1 \ldots y_n$. Let $\mathcal{V}$ consist of the variables appearing in $\Sigma \cup \{\sigma\}$. For each subset $V \subseteq \mathcal{V}$, let i_V be an auxiliary variable, called an *index*. Denote the set of all indices over subsets of $\mathcal{V}$ by $\mathcal{I}$. Define Σ^* as the set of all inclusion atoms $u_1 \ldots u_l i_U \subseteq v_1 \ldots v_l i_V$, where $U = \{u_1, \ldots, u_l\}$, $V = \{v_1, \ldots, v_l\}$, and $u_1 \ldots v_l \approx v_1 \ldots v_l$ or its inverse $v_1 \ldots v_l \approx u_1 \ldots v_l$ is in Σ.

To show that $\Sigma \vdash x_1 \ldots x_n \approx y_1 \ldots y_n$, we will first apply the chase algorithm of database theory to obtain a finite team Y that satisfies Σ^*, where the codomain of Y consists of natural numbers. The indices i_V in Y, in particular, act as multiplicity measures for values of V, making sure that both sides of any marginal identity atom in Σ appear in Y with equal frequency. This way, the probabilistic team $\mathbb{Y}$, defined as the uniform distribution over Y, will in turn satisfy Σ. Finally, we show that the chase algorithm yields a proof of σ, utilizing the fact that $\mathbb{Y}$ satisfies σ by assumption.

Next, we define a team X_0 that serves as the starting point of the chase algorithm. We also describe how assignments over $\mathcal{V}$ that are introduced during the chase are extended to $\mathcal{V} \cup \mathcal{I}$.

Let $X_0 = \{s^*\}$, where s^* is an assignment defined as follows. Let $s^*(x_i) = i$, for $1 \leq i \leq n$, and $s^*(x) = 0$, for $x \in (\mathcal{V} \cup \mathcal{I}) \setminus \{x_1, \ldots, x_n\}$. For a team Y with variable domain $\mathcal{V} \cup \mathcal{I}$ and an assignment s with variable domain $\mathcal{V}$, define $s_Y \colon \mathcal{V} \cup \mathcal{I} \to \mathbb{N}$ as the extension of s such that

$$s_Y(i_V) = |\{t \in Y \mid t \upharpoonright V = s \upharpoonright V\}|, \tag{13}$$

for $i_V \in \mathcal{I}$. That is, the value $s_Y(i_V)$ is the number of repetitions of $s \upharpoonright V$ in Y.

In what follows, we describe a chase rule to expand a team X. We say that an assignment s' *witnesses* an inclusion atom $\vec{x} \subseteq \vec{y}$ for another assignment s, if $s(\vec{x}) = s'(\vec{y})$. Consider the following chase rule:

Chase rule. Let X be a team with variable domain $\mathcal{V} \cup \mathcal{I}$, $s \in X$, and $\sigma := u_1 \ldots u_l i_U \subseteq v_1 \ldots v_l i_V \in \Sigma^*$. Suppose no assignment in X witnesses σ for s. Now let s' be the assignment with variable domain $\mathcal{V}$ that is defined as

$$s'(x) := \begin{cases} s(u_j) & \text{if } x \text{ is } v_j \text{, and} \\ 0 & \text{otherwise.} \end{cases}$$

Then we say that s and σ *generate* the assignment s'_X.

Next, let $\mathcal{S} = (X_0, X_1, X_2, \ldots)$ be a maximal sequence, where $X_{i+1} = X_i \cup \{s'_{X_i}\}$ for an assignment s'_{X_i} generated non-deterministically by some $s \in X_j$ and $\tau \in \Sigma^*$ according to the chase rule, where $j \leq i$ is minimal. Define Y as the union of all elements in $\mathcal{S}$. Note that Y is finite if $\mathcal{S}$ is. In particular, if Y is finite, then it equals X_i, where i is the least integer such that the chase rule is not anymore applicable to X_i. Below, we will show that Y is finite, which follows if the chase algorithm terminates.

It is easy the verify that the following holds, for each $i \in \mathbb{N}$: For any $U = \{u_1, \ldots, u_n\}$ and $s \in X_i$, if the team $X_s := \{t \in X_i \mid t \upharpoonright U = s \upharpoonright U\}$ is of size m, then $\{t(i_U) \mid t \in X_s\} = \{0, \ldots, m-1\}$. That is, the values of i_U in X_s form an initial segment of $\mathbb{N}$ of size $|X_s|$. Therefore, if $s \in X_i$ has no witness for $u_1 \ldots u_l i_U \subseteq v_1 \ldots v_l i_V$ in X_i, then for any $t \in X_i$ such that $s(u_1 \ldots u_l) = t(v_1 \ldots v_l)$, we have $s(i_U) > t(i_V)$. It follows that

$$s(i_U) \geq s'_{X_i}(i_V) \text{ if } s'_{X_i} \text{ is generated by } s \in X_i \text{ and } u_1 \ldots u_l i_U \subseteq v_1 \ldots v_l i_V. \tag{14}$$

We will next show how $\Sigma \vdash x_1 \ldots x_n \approx y_1 \ldots y_n$ follows from the following two claims. We will then prove the claims, which concludes the proof of the theorem.

Claim 1. *Y is finite.*

Claim 2. *If Y contains an assignment s that maps some sequence of variables z_j, for $1 \leq j \leq k$, to distinct $1 \leq i_j \leq n$, then $\Sigma \vdash x_{i_1} \ldots x_{i_k} \approx z_1 \ldots z_k$.*

It follows by construction that $Y \models \Sigma^*$. Since Y is finite by Claim 1, we may define a probabilistic team $\mathbb{Y}$ as the uniform distribution over Y. By the construction of Y and Σ^*, it follows that $\mathbb{Y} \models \Sigma$, and hence $\mathbb{Y} \models x_1 \ldots x_n \approx y_1 \ldots y_n$ follows from the assumption that $\Sigma \models \sigma$. Consequently, Y contains an assignment s which maps y_i to i, for $1 \leq i \leq n$. We conclude that by Claim 2 there is a proof of $x_1 \ldots x_n \approx y_1 \ldots y_n$ from Σ. [7]

To complete the proof, we prove Claims 1 and 2.

[7] Claim 2 is essentially from [9], with the exception that here we also need to consider symmetricity. This claim intuitively states that the chase procedure produces only assignments whose corresponding marginal identity atoms are provable from Σ

Proof of Claim 1. Assume towards contradiction that Y is infinite, which entails that the sequence $\mathcal{S} = (X_0, X_1, X_2, \ldots)$ is infinite. W.l.o.g. the chase rule is always applied to s that belongs to the intersection $X_i \cap X_j$, for minimal $j \leq i$. Define $\mathcal{S}' = (X'_0, X'_1, X'_2, \ldots)$ as the sequence, where $X'_0 = X_0$, and X'_{i+1} is defined as X_j where j is the least integer such that all $s \in X'_i$ and $\sigma \in \Sigma^*$ have a witness in X_j. Due to the application order of the chase rule, it follows that

$$\text{any assignment in } X'_{i+1} \setminus X'_i \text{ is generated by some assignment in } X'_i \setminus X'_{i-1}, \tag{15}$$

assuming $X'_{-1} = \emptyset$. Moreover, $\mathcal{S}'$ is a subsequence of $\mathcal{S}$ which is finite iff $\mathcal{S}$ is.

We first define some auxiliary concepts. For an assignment s in X, we use a shorthand $\mathrm{Base}(s)$ for $s \upharpoonright \mathcal{V}$, called the *base* of s. We also define $\mathrm{Base}(X) := \{\mathrm{Base}(s) \mid s \in X\}$. The *multiplicity in X* of an assignment s is defined as $|\{s' \in X \mid \mathrm{Base}(s) = \mathrm{Base}(s')\}|$. Note that $\mathrm{Base}(Y)$ is finite, for $\mathrm{Base}(s)$ is a mapping from $\mathcal{V}$ into $\{0, \ldots, n\}$ for all $s \in Y$. Thus, since Y is infinite, it contains assignments with infinite multiplicity in Y. Next, we associate each assignment s with the set of its *positive variables* $\mathrm{Pos}(s) := \{x \in \mathcal{V} \mid s(x) > 0\}$, the size of which is called the *degree* of s.

Let k be some integer such that X'_k contains every assignment in Y that has finite multiplicity in Y, and denote X'_k by Z. Let $M \in \{1, \ldots, n\}$ be the maximal degree of any assignment in Y with infinite multiplicity in Y, that is, the maximal degree of any assignment in $Y \setminus Z$. Then, take any $s_L \in X'_L \setminus X'_{L-1}$ of degree M, where $L > k + S$ for $S := |\mathrm{Base}(Y)|$. By property (15), we find a sequence of assignments $(s_0, \ldots, s_L)$, where $s_{i+1} \in X'_{i+1} \setminus X'_i$, for $i < L$, was generated by $s_i \in X'_i \setminus X'_{i-1}$ with the chase rule. Since S is sufficiently large, this sequence has a suffix $(s_l, \ldots, s_m, \ldots, s_L)$ in which each assignment belongs to $Y \setminus Z$, has degree M, and where $l < m$ and $\mathrm{Base}(s_l) = \mathrm{Base}(s_m)$.

It now suffices to show the following subclaim:

Subclaim. *If $t, t' \in Y \setminus Z$ are two assignments with degree M such that t' was generated by t by the chase rule, then $t(i_{\mathrm{Pos}(t)}) \geq t'(i_{\mathrm{Pos}(t')})$.*

The subclaim implies that $s_l(i_{\mathrm{Pos}(s_l)}) \geq s_m(i_{\mathrm{Pos}(s_m)})$, which leads to a contradiction. For this, observe that the assignment construction in (13), together with $\mathrm{Base}(s_l) = \mathrm{Base}(s_m)$, implies that $s_l(i) < s_m(i)$ for all indices i. In particular, we have $s_l(i_{\mathrm{Pos}(s_l)}) < s_m(i_{\mathrm{Pos}(s_m)})$ since $\mathrm{Pos}(s_l) = \mathrm{Pos}(s_m)$. Hence, the assumption that Y is infinite must be false. □

Proof of the subclaim. Suppose t' is generated by t and $u_1 \ldots u_l i_U \subseteq v_1 \ldots v_l i_V \in \Sigma^*$. Without loss of generality $\mathrm{Pos}(t) = \{u_1, \ldots, u_M\}$, in which case $\mathrm{Pos}(t') = \{v_1, \ldots, v_M\}$. We need to show that $t(i_{\mathrm{Pos}(t)}) \geq t'(i_{\mathrm{Pos}(t')})$. Now, $(t(u_1), \ldots, t(u_l))$ is a sequence of the form $(i_1, \ldots, i_M, 0 \ldots, 0)$, where i_j are positive integers. By the assumption that $t \in Y \setminus Z$, there is an integer m such that $t \in X_{m+1} \setminus X_m$ and $Z \subseteq X_m$. We obtain that

$$\begin{aligned}
t(i_{\mathrm{Pos}(t)}) &= |\{s \in X_m \mid (s(u_1), \ldots, s(u_M)) = (i_1, \ldots, i_M)\}| && (16)\\
&= \sum_{j_{M+1}, \ldots, j_l \in \{0, \ldots, n\}} |\{s \in X_m \mid (s(u_1), \ldots, s(u_l)) = (i_1, \ldots, i_M, j_{M+1}, \ldots, j_l)\}| \\
&= |\{s \in X_m \mid (s(u_1), \ldots, s(u_l)) = (i_1, \ldots, i_M, 0 \ldots, 0)\}| + \\
&\qquad \sum_{\substack{j_{M+1}, \ldots, j_l \in \{0, \ldots, n\} \\ (j_{M+1}, \ldots, j_l) \neq (0, \ldots, 0)}} |\{s \in X_m \mid (s(u_1), \ldots, s(u_l)) = (i_1, \ldots, i_M, j_{M+1} \ldots, j_l)\}| \\
&= t(i_U) + \sum_{\substack{j_{M+1}, \ldots, j_l \in \{0, \ldots, n\} \\ (j_{M+1}, \ldots, j_l) \neq (0, \ldots, 0)}} |\{s \in Z \mid (s(u_1), \ldots, s(u_l)) = (i_1, \ldots, i_M, j_{M+1}, \ldots, j_l)\}| && (17)\\
&\geq t'(i_V) + \sum_{\substack{j_{M+1}, \ldots, j_l \in \{0, \ldots, n\} \\ (j_{M+1}, \ldots, j_l) \neq (0, \ldots, 0)}} |\{s \in Z \mid (s(v_1), \ldots, s(v_l)) = (i_1, \ldots, i_M, j_{M+1}, \ldots, j_l)\}| && (18)\\
&= t'(i_{\mathrm{Pos}(t')})
\end{aligned}$$

Here, the assignment construction in (13) entails (16), and it is also used in (17). For the summation term appearing in (17), we note that each assignment whose degree is strictly greater than M must belong to Z. It remains to consider (18); the last equality is symmetrical to the composition of the first four equalities.

almost conjunctive L-$(\ddot{\exists}^{*}\forall^{*})_{[0,1]}[=, \mathrm{SUM}, 0, 1]$		L-$\mathrm{ESO}_{[0,1]}[=, +, 0, 1]$		L-$\mathrm{ESO}_{[0,1]}[=, \times, +, 0, 1]$
$⦀^{*}$		$⦀^{*}$		$⦀^{[26]}$
$\mathrm{FO}(\approx)$	$<^{[25]}$	$\mathrm{FO}(\approx, =(\cdots))$	$<^{*}$	$\mathrm{FO}(\perp\!\!\!\perp)$
				$⦀^{[25]}$
				$\mathrm{FO}(\perp\!\!\!\perp_{\mathrm{c}})$

Table 1: The known expressivity hierarchy of logics with probabilistic team semantics and corresponding ESO variants on metafinite structures. The results of this paper are marked with an asterisk (*).

To show that (18) holds, observe first that $t(i_U) \geq t'(i_V)$ by property (14). For the summation term appearing in (18), suppose $\alpha = |\{s \in Z \mid (s(v_1), \ldots, s(v_l)) = (i_1, \ldots, i_M, j_{M+1}, \ldots, j_l)\}|$, for some sequence $j_{M+1}, \ldots, j_l \in \{0, \ldots, n\}$ containing a positive integer. By the assignment construction in (13), we find an assignment $s_0 \in Z$ such that $(s_0(v_1), \ldots, s_0(v_l), s_0(i_V)) = (i_1, \ldots, i_M, j_{M+1}, \ldots, j_l, \alpha - 1)$. Observe that $v_1 \ldots v_l i_V \subseteq u_1 \ldots u_l i_U \in \Sigma^*$, because Σ^* is symmetrical. Now, since Z is subsumed by Y, which in turn satisfies $v_1 \ldots v_l i_V \subseteq u_1 \ldots u_l i_U$, we find an assignment $s_1 \in Y$ such that $(s_1(u_1), \ldots, s_1(u_l), s_1(i_U)) = (i_1, \ldots, i_M, j_{M+1}, \ldots, j_l, \alpha - 1)$. Since the degree of s_1 is greater than M, we observe that $s_1 \in Z$. This entails that $\alpha \leq |\{s \in Z \mid (s(u_1), \ldots, s(u_l)) = (i_1, \ldots, i_M, j_{M+1}, \ldots, j_l)\}|$ by the assignment construction in (13). From this, we obtain that (18) holds. This shows the subclaim. □

Proof of Claim 2. Note that, if $s \in Y$, then there exists a minimal i such that $s \in X_i \setminus X_{i-1}$. We prove the claim by induction on i. For the initial team $X_0 = \{s^*\}$, we have $s^*(x_i) = i$, for $1 \leq i \leq n$. By reflexivity we obtain $x_{i_1} \ldots x_{i_k} \approx x_{i_1} \ldots x_{i_k}$, and thus the claim holds for the base step.

For the inductive step, suppose $s \in X_{i+1} \setminus X_i$ is generated by some $s' \in X_j \setminus X_{j-1}$, $j \leq i$, and some $u_1 \ldots u_l i_U \subseteq v_1 \ldots v_l i_V$ in Σ^*. For a variable v_i from $v_1, \ldots, v_l$ we say the variable u_i from $u_1, \ldots, u_l$ is its *corresponding* variable. Let $z_1, \ldots, z_k$ be variables as in the claim, i.e., $s(z_j) = i_j \geq 1$, for $1 \leq j \leq k$. Now from the construction of s (i.e., (13)) it follows that $z_1, \ldots, z_k$ are variables from $v_1, \ldots, v_l$. Let $z'_1, \ldots, z'_k$ from $u_1, \ldots, u_l$ denote the corresponding variables of $z_1, \ldots, z_k$. Since s was constructed by s' and $u_1 \ldots u_l i_U \subseteq v_1 \ldots v_l i_V$, it follows that $s(z_1, \ldots, z_k) = s'(z'_1, \ldots, z'_k)$. By applying the induction hypothesis to s', we obtain that Σ yields a proof of $x_{i_1} \ldots x_{i_k} \approx z'_1 \ldots z'_k$. Since $u_1 \ldots u_l \approx v_1 \ldots v_l$ or its inverse is in Σ, using projection and permutation (and possibly symmetricity) we can deduce $z'_1 \ldots z'_k \approx z_1 \ldots z_k$. Thus by transitivity we obtain a proof of $x_{i_1} \ldots x_{i_k} \approx z_1 \ldots z_k$. This concludes the proof of the claim. □

□

8. Conclusion

Our investigations gave rise to the expressiveness hierarchy in Table 1. Furthermore, we established that $\mathrm{FO}(\approx)$ captures P on finite ordered structures, and that $\mathrm{FO}(\approx, =(\cdots))$ captures NP on finite structures. Its worth to note that almost conjunctive $(\ddot{\exists}^{*}\exists^{*}\forall^{*})_{\mathbb{R}}[\leq, +, \mathrm{SUM}, 0, 1]$ is in some regard a maximal tractable fragment of additive existential second-order logic, as dropping either the requirement of being almost conjunctive, or that of having the prefix form $\ddot{\exists}^{*}\exists^{*}\forall^{*}$, leads to a fragment that captures NP. We also showed that the full additive existential second-order logic (with inequality and constants 0 and 1) collapses to NP, a result which as far as we know has not been stated previously.

Lastly, extending the axiom system of inclusion dependencies with a symmetry rule, we presented a sound and complete axiomatization for marginal identity atoms. Beside this result, it is well known that also marginal independence has a sound and complete axiomatization [19]. These two notions play a central role in statistics, as it is a common assumption in hypothesis testing that samples drawn from a population are independent and identically distributed (i.i.d.). It is an interesting open question whether marginal independence and marginal identity, now known to be axiomatizable in isolation, can also be axiomatized together.

9. Acknowledgements

We would like to thank the anonymous referee for a number of useful suggestions. We also thank Joni Puljujärvi and Richard Wilke for pointing out errors in the previous manuscripts.

References

[1] Mikkel Abrahamsen, Anna Adamaszek, and Tillmann Miltzow. The art gallery problem is ∃ℝ-complete. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 65–73, 2018.

[2] Samson Abramsky, Joni Puljujärvi, and Jouko Väänänen. Team semantics and independence notions in quantum physics, 2021. arXiv, 2107.10817.

[3] Rafael Albert and Erich Grädel. Unifying hidden-variable problems from quantum mechanics by logics of dependence and independence. *CoRR*, abs/2102.10931, 2021.

[4] Michael Benedikt, Martin Grohe, Leonid Libkin, and Luc Segoufin. Reachability and connectivity queries in constraint databases. *Journal of Computer and System Sciences*, 66(1):169 – 206, 2003. Special Issue on PODS 2000.

[5] Lenore Blum, Felipe Cucker, Michael Shub, and Steve Smale. *Complexity and Real Computation*. Springer-Verlag, Berlin, Heidelberg, 1997.

[6] Lenore Blum, Mike Shub, and Steve Smale. On a theory of computation and complexity over the real numbers: np-completeness, recursive functions and universal machines. *Bull. Amer. Math. Soc. (N.S.)*, 21(1):1–46, 07 1989.

[7] Peter Bürgisser and Felipe Cucker. Counting complexity classes for numeric computations II: algebraic and semialgebraic sets. *J. Complexity*, 22(2):147–191, 2006.

[8] John F. Canny. Some algebraic and geometric computations in PSPACE. In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA*, pages 460–467, 1988.

[9] Marco A. Casanova, Ronald Fagin, and Christos H. Papadimitriou. Inclusion dependencies and their interaction with functional dependencies. *J. Comput. Syst. Sci.*, 28(1):29–59, 1984.

[10] Marco Console, Matthias F. J. Hofer, and Leonid Libkin. Queries with arithmetic on incomplete databases. In Dan Suciu, Yufei Tao, and Zhewei Wei, editors, *Proceedings of the 39th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2020, Portland, OR, USA, June 14-19, 2020*, pages 179–189. ACM, 2020.

[11] Felipe Cucker and Klaus Meer. Logics which capture complexity classes over the reals. *J. Symb. Log.*, 64(1):363–390, 1999.

[12] George B. Dantzig. *Linear Programming and Extensions*. Princeton University Press, 1963.

[13] George B. Dantzig and Mukund N. Thapa. *Linear Programming 1: Introduction*. Springer-Verlag, Berlin, Heidelberg, 1997.

[14] Arnaud Durand, Miika Hannula, Juha Kontinen, Arne Meier, and Jonni Virtema. Approximation and dependence via multiteam semantics. *Ann. Math. Artif. Intell.*, 83(3-4):297–320, 2018.

[15] Arnaud Durand, Miika Hannula, Juha Kontinen, Arne Meier, and Jonni Virtema. Probabilistic team semantics. In *Foundations of Information and Knowledge Systems - 10th International Symposium, FoIKS 2018, Budapest, Hungary, May 14-18, 2018, Proceedings*, pages 186–206, 2018.

[16] Pietro Galliani. Game Values and Equilibria for Undetermined Sentences of Dependence Logic. MSc Thesis. ILLC Publications, MoL–2008–08, 2008.

[17] Pietro Galliani. Inclusion and exclusion dependencies in team semantics: On some logics of imperfect information. *Annals of Pure and Applied Logic*, 163(1):68 – 84, 2012.

[18] Pietro Galliani and Lauri Hella. Inclusion Logic and Fixed Point Logic. In Simona Ronchi Della Rocca, editor, *Computer Science Logic 2013 (CSL 2013)*, volume 23 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 281–295, Dagstuhl, Germany, 2013. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[19] Dan Geiger, Azaria Paz, and Judea Pearl. Axioms and algorithms for inferences involving probabilistic independence. *Information and Computation*, 91(1):128–141, 1991.

[20] Erich Grädel and Yuri Gurevich. Metafinite model theory. *Inf. Comput.*, 140(1):26–81, 1998.

[21] Erich Grädel and Stephan Kreutzer. Descriptive complexity theory for constraint databases. In *Computer Science Logic, 13th International Workshop, CSL '99, 8th Annual Conference of the EACSL, Madrid, Spain, September 20-25, 1999, Proceedings*, pages 67–81, 1999.

[22] Erich Grädel and Klaus Meer. Descriptive complexity theory over the real numbers. In *Proceedings of the Twenty-Seventh Annual ACM Symposium on Theory of Computing, 29 May-1 June 1995, Las Vegas, Nevada, USA*, pages 315–324, 1995.

[23] Erich Grädel and Richard Wilke. Logics with multiteam semantics. *CoRR*, abs/2011.09834, 2020.

[24] Martin Grohe and Martin Ritzert. Learning first-order definable concepts over structures of small degree. In *32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2017, Reykjavik, Iceland, June 20-23, 2017*, pages 1–12. IEEE Computer Society, 2017.

[25] Miika Hannula, sa Hirvonen, Juha Kontinen, Vadim Kulikov, and Jonni Virtema. Facets of distribution identities in probabilistic team semantics. In *Logics in Artificial Intelligence - 16th European Conference, JELIA 2019, Rende, Italy, May 7-11, 2019, Proceedings*, pages 304–320, 2019.

[26] Miika Hannula, Juha Kontinen, Jan Van den Bussche, and Jonni Virtema. Descriptive complexity of real computation and probabilistic independence logic. In Holger Hermanns, Lijun Zhang, Naoki Kobayashi, and Dale Miller, editors, *LICS '20: 35th Annual ACM/IEEE Symposium on Logic in Computer Science, Saarbrücken, Germany, July 8-11, 2020*, pages 550–563. ACM, 2020.

[27] Miika Hannula and Jonni Virtema. Tractability frontiers in probabilistic team semantics and existential second-order logic over the reals. In Wolfgang Faber, Gerhard Friedrich, Martin Gebser, and Michael Morak, editors, *Logics in Artificial Intelligence - 17th European Conference, JELIA 2021, Virtual Event, May 17-20, 2021, Proceedings*, volume 12678 of *Lecture Notes in Computer Science*, pages 262–278. Springer, 2021.

[28] Uffe Flarup Hansen and Klaus Meer. Two logical hierarchies of optimization problems over the real numbers. *Math. Log. Q.*, 52(1):37–50, 2006.

[29] Wilfrid Hodges. Compositional Semantics for a Language of Imperfect Information. *Journal of the Interest Group in Pure and Applied Logics*, 5 (4):539–563, 1997.

[30] Tapani Hyttinen, Gianluca Paolini, and Jouko Väänänen. A Logic for Arguing About Probabilities in Measure Teams. *Arch. Math. Logic*, 56(5-6):475–489, 2017.

[31] Charles Jordan and Lukasz Kaiser. Machine learning with guarantees using descriptive complexity and SMT solvers. *CoRR*, abs/1609.02664, 2016.

[32] Paris C. Kanellakis, Gabriel M. Kuper, and Peter Z. Revesz. Constraint query languages. *J. Comput. Syst. Sci.*, 51(1):26–52, 1995.

[33] L. G. Khachiyan. A polynomial algorithm in linear programming. *Dokl. Akad. Nauk SSSR*, 244:1093–1096, 1979.

[34] Pascal Koiran. Computing over the reals with addition and order. *Theor. Comput. Sci.*, 133(1):35–47, 1994.

[35] Juha Kontinen and Ville Nurmi. Team logic and second-order logic. In Hiroakira Ono, Makoto Kanazawa, and Ruy de Queiroz, editors, *Logic, Language, Information and Computation*, volume 5514 of *Lecture Notes in Computer Science*, pages 230–241. Springer Berlin / Heidelberg, 2009.

[36] Juha Kontinen and Jouko Väänänen. On definability in dependence logic. *Journal of Logic, Language and Information*, 3(18):317–332, 2009.

[37] Stephan Kreutzer. Fixed-point query languages for linear constraint databases. In *Proceedings of the Nineteenth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, May 15-17, 2000, Dallas, Texas, USA*, pages 116–125, 2000.

[38] Klaus Meer. Counting problems over the reals. *Theor. Comput. Sci.*, 242(1-2):41–58, 2000.

[39] Marcus Schaefer. Complexity of some geometric and topological problems. In *Graph Drawing, 17th International Symposium, GD 2009, Chicago, IL, USA, September 22-25, 2009. Revised Papers*, pages 334–344, 2009.

[40] Marcus Schaefer. Realizability of graphs and linkages. In Pach J., editor, *Thirty Essays on Geometric Graph Theory*. Springer, 2013.

[41] Marcus Schaefer and Daniel Stefankovic. Fixed points, nash equilibria, and the existential theory of the reals. *Theory Comput. Syst.*, 60(2):172–193, 2017.

[42] Szymon Torunczyk. Aggregate queries on sparse databases. In Dan Suciu, Yufei Tao, and Zhewei Wei, editors, *Proceedings of the 39th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2020, Portland, OR, USA, June 14-19, 2020*, pages 427–443. ACM, 2020.

[43] Jouko Väänänen. *Dependence Logic*. Cambridge University Press, 2007.

[44] Steffen van Bergerem and Nicole Schweikardt. Learning concepts described by weight aggregation logic. In *CSL*, volume 183 of *LIPIcs*, pages 10:1–10:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

[45] Richard Wilke. On the presburger fragment of logics with multiteam semantics. Workshop on Logics of Dependence and Independence (LoDE 2020V), 2020.

Appendix A. BSS-toolbox

In this section we give a short introduction to BSS machines (see e.g. [5]). The inputs for BSS machines come from $\mathbb{R}^* := \bigcup\{\mathbb{R}^n \mid n \in \mathbb{N}\}$, which can be viewed as the real analogue of Σ^* for a finite set Σ. The *size* $|x|$ of $x \in \mathbb{R}^n$ is defined as n. We also define $\mathbb{R}_*$ as the set of all sequences $x = (x_i)_{i\in\mathbb{Z}}$ where $x_i \in \mathbb{R}$. The members of $\mathbb{R}_*$ are thus bi-infinite sequence of the form $(\ldots, x_{-2}, x_{-1}, x_0, x_1, x_2, \ldots)$. Given an element $x \in \mathbb{R}^* \cup \mathbb{R}_*$ we write x_i for the ith coordinate of x. The space $\mathbb{R}_*$ has natural shift operations. We define shift left $\sigma_l \colon \mathbb{R}_* \to \mathbb{R}_*$ and shift right $\sigma_r \colon \mathbb{R}_* \to \mathbb{R}_*$ as $\sigma_l(x)_i := x_{i+1}$ and $\sigma_r(x)_i := x_{i-1}$.

Definition 34 (BSS machines). *A BSS machine consists of an input space $\mathcal{I} = \mathbb{R}^*$, a state space $\mathcal{S} = \mathbb{R}_*$, and an output space $\mathcal{O} = \mathbb{R}^*$, together with a connected directed graph whose nodes are labelled by $1, \ldots, N$. The nodes are of five different types.*

1. Input node. *The node labeled by* 1 *is the only input node. The node is associated with a next node $\beta(1)$ and the input mapping $g_I : \mathcal{I} \to \mathcal{S}$.*

2. Output node. *The node labeled by N is the only output node. This node is not associated with any next node. Once this node is reached, the computation halts, and the result of the computation is placed on the output space by the output mapping $g_O : \mathcal{S} \to \mathcal{O}$.*

3. Computation nodes. *A computation node m is associated with a next node $\beta(m)$ and a mapping $g_m : \mathcal{S} \to \mathcal{S}$ such that for some $c \in \mathbb{R}$ and $i, j, k \in \mathbb{Z}$ the mapping g_m is identity on coordinates $l \neq i$ and on coordinate i one of the following holds:*

   - $g_m(x)_i = x_j + x_k$ *(addition),*
   - $g_m(x)_i = x_j - x_k$ *(subtraction),*
   - $g_m(x)_i = x_j \times x_k$ *(multiplication),*
   - $g_m(x)_i = c$ *(constant assignment).*

4. Branch nodes. *A branch node m is associated with nodes $\beta^-(m)$ and $\beta^+(m)$. Given $x \in \mathcal{S}$ the next node is $\beta^-(m)$ if $x_0 \leq 0$, and $\beta^+(m)$ otherwise.*

5. Shift nodes. *A shift node m is associated either with shift left σ_l or shift right σ_r, and a next node $\beta(m)$.*

The input mapping $g_I : \mathcal{I} \to \mathcal{S}$ places an input $(x_1, \ldots, x_n)$ in the state

$$(\ldots, 0, n, x_1, \ldots, x_n, 0, \ldots) \in \mathcal{S},$$

where the size of the input n is located at the zeroth coordinate. The output mapping $g_O \colon \mathcal{S} \to \mathcal{O}$ maps a state to the string consisting of its first l positive coordinates, where l is the number of consecutive ones stored in the negative coordinates starting from the first negative coordinate. For instance, g_O maps

$$(\ldots, 2, 1, 1, 1, n, x_1, x_2, x_3, x_4, \ldots) \in \mathcal{S},$$

to $(x_1, x_2, x_3) \in \mathcal{O}$. A configuration at any moment of computation consists of a node $m \in \{1, \ldots, N\}$ and a current state $x \in \mathcal{S}$. The (sometimes partial) input-output *function $f_M : \mathbb{R}^* \to \mathbb{R}^*$ of a machine M is now defined in the obvious manner. A function $f : \mathbb{R}^* \to \mathbb{R}^*$ is* computable *if $f = f_M$ for some machine M. A language $L \subseteq \mathbb{R}^*$ is* decided *by a BSS machine M if its characteristic function $\chi_L \colon \mathbb{R}^* \to \mathbb{R}^*$ is f_M.*

Deterministic complexity classes.. A machine M *runs in (deterministic) time* $t \colon \mathbb{N} \to \mathbb{N}$, if M reaches the output in $t(|x|)$ steps for each input $x \in \mathcal{I}$. The machine M runs in *polynomial time* if t is a polynomial function. The complexity class $\mathsf{P}_{\mathbb{R}}$ is defined as the set of all subsets of $\mathbb{R}^*$ that are decided by some machine M running in polynomial time.

Nondeterministic complexity classes.. A language $L \subseteq \mathbb{R}^*$ is *decided nondeterministically* by a BSS machine M, if

$$x \in L \quad \text{if and only if} \quad f_M((x, x')) = 1, \text{ for some } x' \in \mathbb{R}^*.$$

Here we assume a slightly different input mapping $g_I : \mathcal{I} \to \mathcal{S}$, which places an input $(x_1, \ldots, x_n, x'_1, \ldots, x'_m)$ in the state

$$(\ldots, 0, n, m, x_1, \ldots, x_n, x'_1, \ldots, x'_m, \ldots) \in \mathcal{S},$$

where the sizes of x and x' are respectively placed on the first two coordinates. When we consider languages that a machine M decides nondeterministically, we call M *nondeterministic*. Sometimes when we wish to emphasize that this is not the case, we call M *deterministic*. Moreover, we say that M is *[0,1]-nondeterministic*, if the guessed strings x' are required to be from $[0, 1]^*$. L is *decided in time* $t \colon \mathbb{N} \to \mathbb{N}$, if, for every $x \in L$, M reaches the output 1 in $t(|x|)$ steps for some $x' \in \mathbb{R}^*$. The machine *runs in polynomial time* if t is a polynomial function. The class $\mathsf{NP}_{\mathbb{R}}$ consists of those languages $L \subseteq \mathbb{R}^*$ for which there exists a machine M that nondeterministically decides L in polynomial time. Note that, in this case, the size of x' above can be bounded by a polynomial (e.g., the running time of M) without altering the definition. The complexity class $\mathsf{NP}_{\mathbb{R}}$ has many natural complete problems such as 4-FEAS, i.e., the problem of determining whether a polynomial of degree four has a real root [6].

Complexity classes with Boolean restrictions.. If we restrict attention to machines M that may use only $c \in \{0, 1\}$ in constant assignment nodes, then the corresponding complexity classes are denoted using an additional superscript 0 (e.g., as in $\mathsf{NP}^0_{\mathbb{R}}$). Complexity classes over real computation can also be related to standard complexity classes. For a complexity class C over the reals, the *Boolean part* of C, written $\mathrm{BP}(C)$, is defined as $\{L \cap \{0, 1\}^* \mid L \in C\}$.

Descriptive complexity.. Similar to Turing machines, also BSS machines can be studied from the vantage point of descriptive complexity. To this end, finite $\mathbb{R}$-structures are encoded as finite strings of reals using so-called rankings that stipulate an ordering on the finite domain. Let $\mathfrak{A}$ be an $\mathbb{R}$-structure over $\tau \cup \sigma$ where τ and σ are relational and functional vocabularies, respectively. A *ranking* of $\mathfrak{A}$ is any bijection $\pi \colon \mathrm{Dom}(A) \to \{1, \ldots, |A|\}$. A ranking π and the lexicographic ordering on $\mathbb{N}^k$ induce a *k-ranking* $\pi_k \colon \mathrm{Dom}(A)^k \to \{1, \ldots, |A|^k\}$ for $k \in \mathbb{N}$. Furthermore, π induces the following encoding $\mathrm{enc}_\pi(\mathfrak{A})$. First we define $\mathrm{enc}_\pi(R^{\mathfrak{A}})$ and $\mathrm{enc}_\pi(f^{\mathfrak{A}})$ for $R \in \tau$ and $f \in \sigma$:

- Let $R \in \tau$ be a k-ary relation symbol. The encoding $\mathrm{enc}_\pi(R^{\mathfrak{A}})$ is a binary string of length $|A|^k$ such that the jth symbol in $\mathrm{enc}_\pi(R^{\mathfrak{A}})$ is 1 if and only if $(a_1, \ldots, a_k) \in R^{\mathfrak{A}}$, where $\pi_k(a_1, \ldots, a_k) = j$.

- Let $f \in \sigma$ be a k-ary function symbol. The encoding $\mathrm{enc}_\pi(f^{\mathfrak{A}})$ is string of real numbers of length $|A|^k$ such that the jth symbol in $\mathrm{enc}_\pi(f^{\mathfrak{A}})$ is $f^{\mathfrak{A}}(\vec{a})$, where $\pi_k(\vec{a}) = j$.

The encoding $\mathrm{enc}_\pi(\mathfrak{A})$ is then the concatenation of the string $(1,\dots,1)$ of length $|A|$ and the encodings of the interpretations of the relation and function symbols in $\tau \cup \sigma$. We denote by $\mathrm{enc}(\mathfrak{A})$ any encoding $\mathrm{enc}_\pi(\mathfrak{A})$ of $\mathfrak{A}$.

Let C be a complexity class and $\mathrm{ESO}_S[O, E, C]$ a logic, where $O \subseteq \{+, \times, \mathrm{SUM}\}$, $E \subseteq \{=, <, \leq\}$, $C \subseteq \mathbb{R}$, and $S \subseteq \mathbb{R}$ or $S = d[0,1]$. Let $X \subseteq \mathbb{R}$ or $X = d[0,1]$, and let $\mathcal{S}$ be an arbitrary class of X-structures over $\tau \cup \sigma$ that is closed under isomorphisms. We write $\mathrm{enc}(\mathcal{S})$ for the set of encodings of structures in $\mathcal{S}$. Consider the following two conditions:

(i) $\mathrm{enc}(\mathcal{S}) = \{\mathrm{enc}(\mathfrak{A}) \mid \mathfrak{A} \in \mathrm{Struc}^X(\phi)\}$ for some $\phi \in \mathrm{ESO}_S[O, E, C][\tau \cup \sigma]\}$,

(ii) $\mathrm{enc}(\mathcal{S}) \in C$.

If (i) implies (ii), we write $\mathrm{ESO}_S[O, E, C] \leq_X C$, and if the vice versa holds, we write $C \leq_X \mathrm{ESO}_S[O, E, C]$. If both directions hold, then we write $\mathrm{ESO}_S[O, E, C] \equiv_X C$. We omit the subscript X in the notation if $X = \mathbb{R}$.

The following results due to Grädel and Meer extend Fagin's theorem to the context of real computation.

Theorem 35 ([22])**.** $\mathrm{ESO}_\mathbb{R}[+, \times, \leq, (r)_{r\in\mathbb{R}}] \equiv \mathsf{NP}_\mathbb{R}$.