

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

Miika Hannula<sup>1</sup>[0000–0002–9637–6664] and Jonni Virtema<sup>2</sup>[0000–0002–1582–3718]

<sup>1</sup> University of Helsinki, Finland [miika.hannula@helsinki.fi](mailto:miika.hannula@helsinki.fi)

<sup>2</sup> Leibniz Universität Hannover, Germany [virtema@thi.uni-hannover.de](mailto:virtema@thi.uni-hannover.de)

**Abstract.** Probabilistic team semantics is a framework for logical analysis of probabilistic dependencies. Our focus is on the 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.

## 1 Introduction

*Metafinite model theory*, introduced by Grädel and Gurevich [12], 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. Finite structures enriched with real arithmetic are called  $\mathbb{R}$ -structures. *Blum-Shub-Smale machines* [2] (BSS machine for short) are essentially random access machines with registers that can store arbitrary real numbers and which can compute rational functions over reals in a single time step, and are thus perfectly suited to compute properties of  $\mathbb{R}$ -structures. *Descriptive complexity theory* for BSS machines and logics on metafinite structures was initiated by Grädel and Meer who showed that  $\text{NP}_{\mathbb{R}}$  (i.e., non-deterministic polynomial time on BSS machines) is captured by a variant of existential second-order logic ( $\text{ESO}_{\mathbb{R}}$ ) over  $\mathbb{R}$ -structures [14]. Since the work by Grädel and Meer, others (see, e.g., [5,17,18,28]) have shed more light upon *the descriptive complexity over the reals* mirroring the development of classical descriptive complexity.

In addition to metafinite structures, the connection between logical definability encompassing numerical structures and computational complexity has received attention in *constraint databases* [1,13,27]. 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 [22].

Anew 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. [15,32] for declarative frameworks for machine learning utilizing logic, [4,30] for very recent works on logical query languages with arithmetic, and [21] for applications of descriptive complexity in machine learning.

Team semantics is the semantical framework of modern logics of dependence and independence. Introduced by Hodges [19] and adapted to dependence logic by Väänänen [31], 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 [9] and Hyttinen et al. [20], a systematic study on the topic can be traced to [7,8]. In *probabilistic team semantics* the basic semantic units are probability distributions, called *probabilistic teams*. This shift from set based semantics to distribution based semantics enables probabilistic notions of dependence to be embedded to the framework. In [8] probabilistic team semantics was studied in relation to the dependence concept that is most central in statistics: conditional independence. Mirroring [10,14,26] the expressiveness of probabilistic independence logic ( $\text{FO}(\perp_c)$ ), obtained by extending first-order logic with conditional independence, was in [8,17] characterised in terms of arithmetic variants of existential second-order logic. In [17] the data complexity of  $\text{FO}(\perp_c)$  was also identified in the context of BSS-machines and the existential theory of the reals. In [16] the focus was shifted to the expressivity hierarchies between probabilistic logics defined in terms of different quantitative dependencies.

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 [11]. In contrast, other natural team-based logics, such as dependence and independence logic, capture non-deterministic polynomial time [10,26,31], and many variants, such as team logic, have an even higher complexity [25]. Thus it should come as no surprise if quantitative variants of many team-based logics turn out to be intractable; in principle, adding arithmetical operations and/or counting cannot be a mitigating factor when it comes to complexity.

In this paper we focus on the complexity and expressivity of *probabilistic inclusion logic*, which is the extension of first-order logic with so-called *marginal identity atoms* defined on probabilistic teams. The marginal identity atom  $x \approx y$  states that the probability of  $x$  being  $a$  is the same as the probability of  $y$  being  $a$ , for all values  $a$ .

**Our contribution.** We use strong results from linear programming to obtain the following complexity results restricted to finite structures. We identify a natural fragment of additive  $\text{ESO}_{\mathbb{R}}$  (that is, *almost conjunctive*  $(\exists^* \forall^*)_{\mathbb{R}}[\leq, +, \text{SUM}, 0, 1]$ ) which captures P on ordered structures (see Theorem 4 for the precise formulation). In contrast, we show that the full additive  $\text{ESO}_{\mathbb{R}}$  captures NP. Moreover, we establish that the so-called *loose fragments*, almost conjunctive  $\text{L-}(\exists^* \forall^*)_{d[0,1]}[=, \text{SUM}, 0, 1]$  and  $\text{L-ESO}_{d[0,1]}[=, \text{SUM}, 0, 1]$ , 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 are transferred also to these fragments. Finally, we show that inclusion logic can be conservatively embedded to 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 [11]) for the fact that inclusion logic can express only polynomial time properties.

## 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,  $\tau$  and  $\sigma$  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  $\text{ar}(f)$  and  $\text{ar}(R)$ , respectively. If  $f$  is a function with domain  $\text{Dom}(f)$  and  $A$  a set, we define  $f \upharpoonright A$  to be the function with domain  $\text{Dom}(f) \cap A$  that agrees with  $f$  for each element in its domain. Given a finite set  $S$ , a function  $f: S \rightarrow [0, 1]$  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  $\text{supp}(f) = \{s \in S \mid f(s) > 0\}$ . Also,  $f$  is called *uniform* if  $f(s) = f(s')$  for all  $s, s' \in \text{supp}(f)$ .

**Definition 1 ( $\mathbb{R}$ -structure).** 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  $\tau$  is a finite relational structure, and each  $g^{\mathfrak{A}}$  is a function from  $A^{\text{ar}(g)}$  to  $\mathbb{R}$ , is called  $\mathbb{R}$ -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  $A^{\text{ar}(g)}$  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 ( $\text{ESO}_{\mathbb{R}}$ ) that is designed to describe properties of  $\mathbb{R}$  structures. As first-order terms we have only first-order variables. For a set  $\sigma$  of function symbols, the set of numerical  $\sigma$ -terms  $i$  is generated by the following grammar:  $i ::= c \mid f(\vec{x}) \mid i + i \mid i \times i \mid \text{SUM}_{\vec{y}} i$ , where the interpretations of  $+$ ,  $\times$ ,  $\text{SUM}$  are the standard addition, multiplication, and summation of real numbers, respectively.

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

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

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

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

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

*Almost conjunctive.* A formula  $\phi \in \text{ESO}_S[O, E, C]$  is *almost conjunctive*, if for every subformula  $(\psi_1 \vee \psi_2)$  of  $\phi$ , no function term occurs in  $\psi_i$ , 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  $\text{ESO}_S[O, E, C]$  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 a formula  $\phi \in \mathcal{L}$ , define  $\text{Struc}_X(\phi)$  ( $\text{Struc}_{\text{fin}}(\phi)$ , resp.) to be the class of pairs  $(\mathfrak{A}, s)$  where  $\mathfrak{A}$  is an  $\mathbb{R}$ -structure (finite structures, resp.) and  $s$  an assignment such that  $\mathfrak{A} \models_s \phi$  and  $f^{\mathfrak{A}}$  is a function with codomain  $X$  for each function variable  $f$ . Additionally,  $\text{Struc}_{d[0,1]}(\phi)$  is the class of  $(\mathfrak{A}, s) \in \text{Struc}_{[0,1]}(\phi)$  such that each  $f^{\mathfrak{A}}$  is a distribution. If  $X$  is a set of reals or from  $\{“d[0,1]”, “fin”\}$ , 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  $\text{Struc}_X(\phi) = \text{Struc}_X(\psi)$ . 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 $\text{ESO}_{\mathbb{R}}$

On finite structures  $\text{ESO}_{\mathbb{R}}[\leq, +, \times, 0, 1]$  is known to capture the complexity class  $\exists\mathbb{R}$  [3,14,29], 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  $(\ddot{\exists}^* \exists^* \forall^*)_{\mathbb{R}}[\leq, +, \text{SUM}, 0, 1]$ .

**Proposition 3.** *Let  $\phi$  be an almost conjunctive  $\text{ESO}_{\mathbb{R}}[\leq, +, \text{SUM}, 0, 1]$ -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  $\phi$  has no free function variables, the systems of linear inequations in  $\mathcal{S}$  have integer coefficients.*

*Proof.* Fix  $\phi$ . We assume, without loss of generality, that all variables that are quantified in  $\phi$  are quantified exactly once, and that the sets of free and bound variables of  $\phi$  are disjoint. Moreover, we assume that  $\phi$  is of the form  $\exists \vec{g} \exists \vec{f} \forall \vec{x} \theta$ , where  $\vec{f}$  is a tuple of function variables and  $\theta$  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  $\phi$ .

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  $\phi$ . In the sequel, each interpretation for the variables in  $\vec{y}$  yields a system of linear equations. Given an interpretation  $v: Y \rightarrow A$ , we will denote by  $S_v$  the related system of linear equations to be defined below. We then set  $\mathcal{S}_{\mathfrak{A},s} := \{S_v \mid v: Y \rightarrow A\}$ . The system of linear equations  $S_v$  is defined as  $S_v := \bigcup_{u: X \rightarrow A} S_v^u$ , 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  $\theta_v^u$  denote the formula obtained from  $\theta$  by the following simultaneous substitution: If  $(\psi_1 \vee \psi_2)$  is a subformula of  $\theta$  such that no function variable occurs in  $\psi_i$ , then  $(\psi_1 \vee \psi_2)$  is substituted with  $\top$ , if

$$\mathfrak{A} \models_{s_v^u} \psi_i, \quad (1)$$

and with  $\psi_{3-i}$  otherwise. The set  $S_v^u$  is now generated from  $\theta_v^u$  together with  $u$  and  $v$ . Note that  $\theta_v^u$  is a conjunction of first-order or numerical atoms  $\theta_i$ ,  $i \in I$ , for some index set  $I$ . For each conjunct  $\theta_i$  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:

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

Let  $\theta^*$  be the conjunction of those conjuncts of  $\theta_v^u$  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  $\phi$  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  $w: Z \rightarrow \mathbb{R}$ , where  $Z := \{z_{\vec{a},f} \mid h \in \vec{f} \text{ and } \vec{a} \in A^{\text{ar}(f)}\}$ , be the function given by a solution for  $S$ . By construction,  $S = S_v$ , for some  $v: Y \rightarrow A$ . 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  $u: X \rightarrow A$ . Now, from (1) and the related substitutions, we obtain that  $\mathfrak{A}' \models_{s_v^u} \theta$  for every  $u: X \rightarrow A$ , and hence  $\mathfrak{A}' \models_{s_v} \forall x_1 \dots \forall x_n \theta$ . 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  $\mathfrak{A}' \models_{s_v} \forall x_1 \dots \forall x_n \theta$ . 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  $u: X \rightarrow A$ . Moreover, it follows that the function defined by  $z_{\vec{a},f} \mapsto f^{\mathfrak{A}'}(\vec{a})$ , for  $f \in \vec{f}$  and  $\vec{a} \in A^{\text{ar}(f)}$ , is a solution for  $S_v$ .  $\square$

**Theorem 4.** *The data complexity of almost conjunctive  $\text{ESO}_{\mathbb{R}}[\leq, +, \text{SUM}, 0, 1]$ -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 conjunctive  $\text{ESO}_{\mathbb{R}}[\leq, +, \text{SUM}, 0, 1]$ -formula  $\phi$  of relational vocabulary  $\tau$  of the required form. Given a  $\tau \cup \emptyset$  structure  $\mathfrak{A}$  and an assignment  $s$  for the free

variables of  $\phi$ , 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 [23]. 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.  $\square$

We later show that probabilistic inclusion logic captures P on finite ordered structures (Corollary 23) and can be translated to almost conjunctive  $L-(\exists^*\forall^*)_{d[0,1]}[\leq, \text{SUM}, 0, 1]$  (Lemma 16). Hence already almost conjunctive  $L-(\exists^*\forall^*)_{\mathbb{R}}[\leq, \text{SUM}, 0, 1]$  captures P.

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

### 3.2 Full additive $\text{ESO}_{\mathbb{R}}$

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

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

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

As  $\text{ESO}_{\mathbb{R}}[\leq, +, 0, 1]$  subsumes ESO, by Fagin's Theorem, it can express all NP properties. We only need to prove that any  $\text{ESO}_{\mathbb{R}}[\leq, +, 0, 1]$ -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 ( $\text{ESO}_{\mathbb{R}}[\leq, +, \times, (r)_{r \in \mathbb{R}}]$ ) corresponds to non-deterministic polynomial time over the reals ( $\text{NP}_{\mathbb{R}}$ ) over BSS machines [14, Theorem 4.2]. To continue from this, some additional terminology is needed. Let  $C_{\mathbb{R}}$  be a complexity class over the reals.

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

A straightforward adaptation of [14, Theorem 4.2] yields the following theorem.

**Theorem 7 ([14]).**  $\text{ESO}_{\mathbb{R}}[\leq, +, 0, 1]$  captures  $\text{NP}_{\text{add}}^0$  on  $\mathbb{R}$ -structures.

If we can establish that the Boolean part  $\text{BP}(\text{NP}_{\text{add}}^0)$  of  $\text{NP}_{\text{add}}^0$  collapses to NP, we have completed the proof of Theorem 6. Observe that another variant of this theorem readily holds;  $\text{ESO}_{\mathbb{R}}[=, +, (r)_{r \in \mathbb{R}}]$ -definable properties of  $\mathbb{R}$ -structures are recognizable in  $\text{NP}_{\text{add}}$  branching on equality, which in turn, over Boolean inputs, collapses to NP [24, 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)  $\text{NP}_{\text{add}}$  equals NP/poly over Boolean inputs [24, Theorem 11]. What we show next is that disallowing machine constants other than 0 and 1, but allowing branching on inequality, is a mixture that leads to a collapse to NP. The proof adapts arguments from [24] and can be found in Appendix A.

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

*Proof.* Clearly  $\text{NP} \leq \text{BP}(\text{NP}_{\text{add}}^0)$ ; a Boolean guess for an input  $x$  can be constructed by comparing to zero each component of a real guess  $y$ , 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  $\text{BP}(\text{NP}_{\text{add}}^0)$ ; we need to show that  $L$  belongs also to  $\text{NP}$ . Let  $M$  be a BSS machine such that its running is bounded by some polynomial  $p$ , and for all Boolean inputs  $x \in \{0, 1\}^*$ ,  $x \in L$  if and only if there is  $y \in \mathbb{R}^{p(|x|)}$  such that  $M$  accepts  $(x, y)$ .

The non-deterministic computation of  $M$  on  $x$  can be described by guessing the whole non-deterministic polynomial time computation including the outcomes of the comparisons in the BSS computation. During the computation the value of each register is a linear function of the values of the register in previous time step, and ultimately from the constants 0 and 1, the input  $x$ , and the real guess  $y$  of polynomial length. Thus it is possible to construct in polynomial time a system  $S$  of linear inequations on  $y$ . A more complete proof can be found in Appendix A.  $\square$

## 4 Probabilistic team semantics and additive $\text{ESO}_{\mathbb{R}}$

### 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  $\mathbb{X}: X \rightarrow [0, 1]$ , 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  $\text{Dom}(\mathbb{X})$  and  $\text{Dom}(X)$ .  $A$  is called the *value domain* of  $X$  and  $\mathbb{X}$ .

Let  $\mathbb{X}: X \rightarrow [0, 1]$  be a probabilistic team,  $x$  a variable,  $V \subseteq \text{Dom}(\mathbb{X})$  a set of variables, and  $A$  a set. The *projection* of  $\mathbb{X}$  on  $V$  is defined as  $\text{Pr}_V(\mathbb{X}): X \upharpoonright V \rightarrow [0, 1]$  such that  $s \mapsto \sum_{t \upharpoonright V = s} \mathbb{X}(t)$ , where  $X \upharpoonright V := \{t \upharpoonright V \mid t \in X\}$ . Define  $S_{x,A}(\mathbb{X})$  as the set of all probabilistic teams  $\mathbb{Y}$  with variable domain  $\text{Dom}(\mathbb{X}) \cup \{x\}$  such that  $\text{Pr}_{\text{Dom}(\mathbb{X}) \setminus \{x\}}(\mathbb{Y}) = \text{Pr}_{\text{Dom}(\mathbb{X}) \setminus \{x\}}(\mathbb{X})$  and  $A$  is a value domain of  $\mathbb{Y} \upharpoonright \{x\}$ . We denote by  $\mathbb{X}[A/x]$  the unique  $\mathbb{Y} \in S_{x,A}(\mathbb{X})$  such that

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

If  $x$  is a fresh variable, then this equation becomes  $\mathbb{Y}(s) = \frac{\mathbb{X}(s)}{|A|}$ . 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  $\alpha$  be a real number, and  $f$  and  $g$  be functions from a shared domain into real numbers. The scalar multiplication  $\alpha f$  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 first-order formulae in negation normal form.

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

$$\begin{aligned} \mathfrak{A} \models_{\mathbb{X}} \alpha &\Leftrightarrow \forall s \in \text{supp}(\mathbb{X}) : \mathfrak{A} \models_s \alpha \\ \mathfrak{A} \models_{\mathbb{X}} (\psi \wedge \theta) &\Leftrightarrow \mathfrak{A} \models_{\mathbb{X}} \psi \text{ and } \mathfrak{A} \models_{\mathbb{X}} \theta \\ \mathfrak{A} \models_{\mathbb{X}} (\psi \vee \theta) &\Leftrightarrow \mathfrak{A} \models_{\mathbb{Y}} \psi \text{ and } \mathfrak{A} \models_{\mathbb{Z}} \theta \text{ for some } \mathbb{Y}, \mathbb{Z}, \\ &\quad \alpha \in [0, 1] \text{ such that } \alpha\mathbb{Y} + (1 - \alpha)\mathbb{Z} = \mathbb{X} \\ \mathfrak{A} \models_{\mathbb{X}} \forall x \psi &\Leftrightarrow \mathfrak{A} \models_{\mathbb{X}[A/x]} \psi \\ \mathfrak{A} \models_{\mathbb{X}} \exists x \psi &\Leftrightarrow \mathfrak{A} \models_{\mathbb{Y}} \psi \text{ for some } \mathbb{Y} \in S_{x,A}(\mathbb{X}) \end{aligned}$$

The satisfaction relation  $\models_s$  denotes the Tarski semantics of first-order logic.

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  $\mathbb{X}: X \rightarrow \mathbb{R}_{\geq 0}$ . Given a first-order formula  $\alpha$ , we write  $\mathbb{X}_{\alpha}$  for the restriction of the weighted team  $\mathbb{X}$  to the assignments of  $X$  satisfying  $\alpha$  (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  $\tau$ -structure over a finite domain  $A$ , and  $\mathbb{X}: X \rightarrow \mathbb{R}_{\geq 0}$  a weighted team. The satisfaction relation  $\models_{\mathbb{X}}^w$  for first-order logic is defined exactly as in Definition 9, except that for  $\vee$  we define instead:

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

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:

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

- **Dependence atom.** For a sequence of variables  $\vec{x}$  and a variable  $y$ ,  $=(\vec{x}, y)$  is a dependence atom with satisfaction defined as:

$$\mathfrak{A} \models_X =(\vec{x}, y) \Leftrightarrow \text{for all } s, s' \in X : \text{ if } s(\vec{x}) = s'(\vec{x}), \text{ then } s(y) = s'(y).$$

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

Observe that any dependency  $\alpha$  over team semantics can also be interpreted in probabilistic team semantics:  $\mathfrak{A} \models_{\mathbb{X}} \alpha$  iff  $\mathfrak{A} \models_{\text{supp}(\mathbb{X})} \alpha$ . For a list  $\mathcal{C}$  of dependencies, we write  $\text{FO}(\mathcal{C})$  for the extension of first-order logic with the dependencies in  $\mathcal{C}$ . The logics  $\text{FO}(\approx)$  and  $\text{FO}(\subseteq)$ , in particular, are called *probabilistic inclusion logic* and *inclusion logic*, respectively. We conclude this section with a list of useful proposition. We omit the proofs which are straightforward structural inductions.



**Proposition 12.** *Let  $\phi \in \text{FO}(\mathcal{C})$ , where  $\mathcal{C}$  is a list of dependencies over team semantics. Let  $\mathfrak{A}$  be a structure and  $\mathbb{X}$  a weighted team. Then  $\mathfrak{A} \models_{\mathbb{X}}^w \phi \Leftrightarrow \mathfrak{A} \models_{\text{supp}(\mathbb{X})} \phi$ .*

**Proposition 13 ([16]).** *Let  $\phi \in \text{FO}(\mathcal{C})$ , where  $\mathcal{C}$  is a set of dependencies over probabilistic team semantics. Let  $\mathfrak{A}$  be a structure and  $\mathbb{X}$  a weighted team. Now  $\mathfrak{A} \models_{\mathbb{X}}^w \phi \Leftrightarrow \mathfrak{A} \models_{\frac{1}{|\mathbb{X}|}\mathbb{X}} \phi$ .*

**Proposition 14.** *Let  $\phi \in \text{FO}(\approx, \mathcal{C})$  be a formula, where  $\mathcal{C}$  is a list of dependencies over team semantics. Let  $\mathfrak{A}$  be a structure,  $\mathbb{X}$  a weighted team, and  $r$  any positive real. Then  $\mathfrak{A} \models_{\mathbb{X}}^w \phi \Leftrightarrow \mathfrak{A} \models_{r\mathbb{X}}^w \phi$ .*

## 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  $\exists^*\forall^*$  quantifier prefix.

*Expressivity comparisons for probabilistic team semantics.* Fix a list of atoms  $\mathcal{C}$  over probabilistic team semantics. For a probabilistic team  $\mathbb{X}$  with variable domain  $\{x_1, \dots, x_n\}$  and value domain  $A$ , the function  $f_{\mathbb{X}} : A^n \rightarrow [0, 1]$  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  $\phi \in \text{FO}(\mathcal{C})$  of vocabulary  $\tau$  and with free variables  $\{x_1, \dots, x_n\}$ , the class  $\text{Struc}_{d[0,1]}(\phi)$  is defined as the class of  $\mathbb{R}$ -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  $\tau$ -structure underlying  $\mathfrak{A}$ .

Let  $\mathcal{L}$  be a logic. We write  $\text{FO}(\mathcal{C}) \leq \mathcal{L}$  if for every formula  $\phi \in \text{FO}(\mathcal{C})$  of vocabulary  $\tau$  there is a sentence  $\psi \in \mathcal{L}$  of vocabulary  $\tau \cup \{f\}$  such that  $\text{Struc}(\phi) = \text{Struc}_{d[0,1]}(\psi)$ . Vice versa, we write  $\mathcal{L} \leq \text{FO}(\mathcal{C})$  if for every sentence  $\psi \in \mathcal{L}$  of vocabulary  $\tau \cup \{f\}$  there is a formula  $\phi \in \text{FO}(\mathcal{C})$  of vocabulary  $\tau$  such that  $\text{Struc}_{d[0,1]}(\phi) = \text{Struc}_{d[0,1]}(\psi)$ . For two team logics  $\mathcal{L}, \mathcal{L}'$ , we write  $\mathcal{L} \leq \mathcal{L}'$ , if for every  $\phi \in \mathcal{L}$  there exists  $\psi \in \mathcal{L}'$  such that  $\mathfrak{A} \models_{\mathbb{X}} \phi \Leftrightarrow \mathfrak{A} \models_{\mathbb{X}} \psi$ , for every  $\mathfrak{A}$  and  $\mathbb{X}$ .

**Theorem 15.** *The following equivalences hold:*

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

We divide the proof of Theorem 15 into two parts. First, we consider the direction from probabilistic team semantics to existential second-order logic over the reals.

## 4.3 From probabilistic team semantics to existential second-order logic

Let  $c$  and  $d$  be two distinct constants. Let  $\phi(\vec{x}) \in \text{FO}(\approx, =(\dots))$  be a formula whose free variables are from the sequence  $\vec{x} = (x_1, \dots, x_n)$ . We construct recursively an  $\text{L-ESO}_{d[0,1]}[=, \text{SUM}, 0, 1]$ -formula  $\phi^*(f)$ , with exactly one free function variable  $f$ :

- (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) If  $\phi(\vec{x})$  is  $\vec{x}_0 \approx \vec{x}_1$ , where  $\vec{x} = \vec{x}_0 \vec{x}_1 \vec{x}_2$ , then
 
$$\phi^*(f) := \forall \vec{y} \text{SUM}_{\vec{x}_1, \vec{x}_2} f(\vec{y}, \vec{x}_1, \vec{x}_2) = \text{SUM}_{\vec{x}_0, \vec{x}_2} f(\vec{x}_0, \vec{y}, \vec{x}_2).$$
- (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 define  $\phi^*(f)$  as
 
$$\exists g \forall \vec{x} (\text{SUM}_y g(\vec{x}, y) = f(\vec{x}) \wedge \forall y (y = c \vee y = d \vee g(\vec{x}, y) = 0) \wedge \psi_0^*(g^a) \wedge \psi_1^*(g^b)),$$
 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  $\phi^*(f) := \exists g ((\forall \vec{x} \text{SUM}_y g(\vec{x}, y) = f(\vec{x})) \wedge \psi^*(g))$ .
- (7) If  $\phi(\vec{x})$  is of the form  $\forall y \psi(\vec{x}, y)$ , then
 
$$\phi^*(f) := \exists g (\forall \vec{x} (\forall y \forall z g(\vec{x}, y) = g(\vec{x}, z) \wedge \text{SUM}_y g(\vec{x}, y) = f(\vec{x})) \wedge \psi^*(g)).$$

We obtain the following lemma, the proof of which can be found in Appendix B. The proof utilizes the translation given above. Quantification of functions with total weight  $\leq 1$  are simulated by using higher arity functions. Similarly, Proposition 14 enables the encoding of multiple distributions in a single function by scaling. The claim (ii) follows when the translation for dependence atoms  $=(\vec{x}_0, x_1)$  and  $\vec{x} = \vec{x}_0 x_1 \vec{x}_2$  is modified to  $\forall \vec{x}_0 \exists x_1 \text{SUM}_{\vec{x}_2} f(\vec{x}) = \text{SUM}_{x_1 \vec{x}_2} f(\vec{x})$ . Finally (iii) follows when the case for dependence atoms is omitted.

**Lemma 16.** *The following hold:*

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

We recall from Proposition 3 that almost conjunctive  $(\ddot{\exists}^* \exists^* \forall^*)_{\mathbb{R}}[\leq, +, \text{SUM}, 0, 1]$  is in polynomial time in terms of data complexity. Since dependence logic captures NP, 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.

**Corollary 17.**  $\text{FO}(\approx, =(\cdots))$  captures NP on finite structures.

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

Let us then turn to the direction from existential second-order logic over the reals to probabilistic team semantics. We can make two simplifying assumptions without loss of generality. First, we disregard the real constants 0 and 1 as they are definable in  $\text{L-ESO}_{d[0,1]}[=, \text{SUM}]$ ; any nullary distribution variable is interpreted as the real number 1, and using this and a unary distribution variable the real number 0 is also definable. Second, we can restrict attention to formulae in Skolem normal form.<sup>3</sup>

<sup>3</sup> The corresponding Lemma 3 in [8] includes multiplication but the proof works also without it. We would like to thank Richard Wilke for noting that the construction used in [8] to prove Lemma 18 had an element that yields circularity. It is, fortunately, straightforward to mend the proof such that the issue is avoided. See Appendix C for the fixed proof.

**Lemma 18 ([8]).** *For every formula  $\phi \in \text{L-ESO}_{d[0,1]}[=, \text{SUM}]$  there is a formula  $\phi^* \in \text{L-}(\ddot{\exists}^*\forall^*)_{d[0,1]}[=, \text{SUM}]$  such that  $\text{Struc}_{d[0,1]}(\phi) = \text{Struc}_{d[0,1]}(\phi^*)$ , and any second sort identity atom in  $\phi^*$  is of the form  $f_i(\vec{w}) = \text{SUM}_{\vec{v}} f_j(\vec{u}, \vec{v})$  for distinct  $f_i$  and  $f_j$  of which at least one is quantified. Furthermore,  $\phi^*$  is almost conjunctive if  $\phi$  is.*

The translation presented next is similar to one found in [8], with the exception that probabilistic independence atoms cannot be used here. Without loss of generality each structure is enriched with two distinct constants  $c$  and  $d$ ; such constants are definable in  $\text{FO}(\approx, =(\dots))$  by  $\exists cd(=(c) \wedge =(d) \wedge c \neq d)$ , and for almost conjunctive sentences they are not needed. Given  $\phi(f) = \exists \vec{f} \forall \vec{x} \theta(f, \vec{x}) \in \text{L-}(\ddot{\exists}^*\forall^*)_{d[0,1]}[=, \text{SUM}]$  of the form described in the previous lemma, we define

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

where  $\vec{y}_i$  are sequences of variables of length  $\text{ar}(f_i)$ , and  $\theta \mapsto \Theta$  is defined recursively as follows:

- (1) If  $\theta$  is a literal of the first sort, let  $\Theta := \theta$ .
- (2) If  $\theta$  is of the form  $f_i(\vec{x}_i) = \text{SUM}_{\vec{x}_k} f_j(\vec{x}_k \vec{x}_l)$ , let  $\Theta := \exists \alpha \beta \psi$  for  $\psi$  given as

$$(\alpha = x \leftrightarrow \vec{x}_i = \vec{y}_i) \wedge (\beta = x \leftrightarrow \vec{x}_l = \vec{y}_l) \wedge \vec{x} \alpha \approx \vec{x} \beta, \quad (2)$$

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  $\vec{y}_k$  and  $\vec{y}_l$  whose respective lengths are  $|\vec{x}_k|$  and  $|\vec{x}_l|$ .

- (3) If  $\theta$  is  $\theta_0 \wedge \theta_1$ , let  $\Theta := \Theta_0 \wedge \Theta_1$
- (4) If  $\theta$  is  $\theta_0 \vee \theta_1$ , let  $\Theta := \exists z \left( =(\vec{x}, z) \wedge ((\Theta_0 \wedge z = a) \vee (\Theta_1 \wedge z = b)) \right)$ .

Alternatively, if  $\theta_0$  contains no function terms, let  $\Theta := \theta_0 \vee (\theta_0^- \wedge \Theta_1)$ , where  $\theta_0^-$  is obtained from  $\neg \theta_0$  by pushing  $\neg$  in front of atomic formulae.

The full proof of the next lemma can be found in Appendix D.

**Lemma 19.** *Let  $\phi(f) \in \text{L-}(\ddot{\exists}^*\forall^*)_{d[0,1]}[=, \text{SUM}]$  be of the form described in Lemma 18. Then there is a formula  $\Phi(\vec{x}) \in \text{FO}(\approx, =(\dots))$  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  $\Phi(\vec{x}) \in \text{FO}(\approx)$ .

This completes the proof of Theorem 15. We can generalize or modify Theorem 15 to some extent. First, note that “ $\leq$ ” is interchangeable with “ $=$ ” already in the almost conjunctive fragment. Clearly, any equality atom is expressible in terms of a conjunction of two inequality atoms. In addition, for two numerical terms  $i$  and  $j$  it holds that  $i \leq j$  if and only if  $i + r = j$  for some real  $r$  from the unit interval. Such a real can be guessed e.g. with a unary distribution variable  $f$ . Second, by [17, Lemma 6.4]<sup>4</sup>,  $\text{L-ESO}_{[0,1]}[=, +, 0, 1] \equiv_{d[0,1]} \text{L-ESO}_{d[0,1]}[=, \text{SUM}]$ .

<sup>4</sup> This lemma includes multiplication but works also without it. However, it does not preserve the almost conjunctive form.

**Corollary 20.** *The following equivalences hold:*

- (i)  $\text{FO}(\approx, =(\dots)) \equiv \text{L-ESO}_{[0,1]}[\leq, +, 0, 1]$ .
- (ii)  $\text{FO}(\approx) \equiv \text{almost conjunctive L-}(\exists^* \forall^*)_{d[0,1]}[\leq, \text{SUM}, 0, 1]$ .

## 5 (e) FO(inc) sentences can be interpreted in FO(inc)

Next we turn to the relationship between inclusion and probabilistic inclusion logics. The logics are comparable since, as shown in Propositions 12 and 13, team semantics embeds into probabilistic team semantics conservatively. The seminal result by Galliani and Hella shows that inclusion logic captures polynomial time over finite ordered structures [11]. 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* [10], 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

$$\psi^k(\vec{x}_1, \vec{x}_2) := \forall \vec{u} \exists v_1 v_2 \forall \vec{z}' \exists \vec{z} ((\vec{x}_1 = \vec{u} \leftrightarrow v_1 = y) \wedge (\vec{x}_2 = \vec{u} \leftrightarrow v_2 = y) \wedge \quad (3) \\ (\vec{z}' = \vec{y} \rightarrow \vec{z} = \vec{y}) \wedge (\neg \vec{z} = \vec{y} \vee \vec{u} v_1 \approx \vec{u} v_2)),$$

where  $\vec{z}$  and  $\vec{z}'$  are variable tuples of length  $k$ , and  $\vec{y}$  is obtained by concatenating  $k$  times some variable  $y$  in  $\vec{u}$ . For intuition behind (3), note that the focus is on one possible value  $\vec{u}$  of  $\vec{x}_1$  and  $\vec{x}_2$  at a time. Using dummy sequences  $\vec{z}$  and  $\vec{z}'$  this formula stipulates that a selected positive fraction of any positive assignment weight must be associated with the marginal identity atom in the last conjunct. Once weight differences between assignments have been adjusted, it is possible to express equiextension atom in terms of marginal identity.

We now show that  $\phi^k(\vec{x}, \vec{y})$  captures  $\vec{x} \bowtie \vec{y}$  over weighted teams in which the minimal positive weight is  $\frac{1}{n^k}$ , where  $n$  is the size of the structure. Using this result we then establish a translation from inclusion logic to its probabilistic variant. The proof of the following lemma can be found in Appendix E.

**Lemma 21.** *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) *If  $\mathfrak{A} \models_{\mathbb{X}}^w \vec{x}_1 \bowtie \vec{x}_2$  and  $\forall s \in X : \mathbb{X}(s) = 0$  or  $\mathbb{X}(s) \geq \frac{|\mathbb{X}|}{n^k}$ , then  $\mathfrak{A} \models_{\mathbb{X}}^w \phi^k(\vec{x}, \vec{y})$ .*
- (ii) *If  $\mathfrak{A} \models_{\mathbb{X}}^w \phi^k(\vec{x}, \vec{y})$ , then  $\mathfrak{A} \models_{\mathbb{X}}^w \vec{x}_1 \bowtie \vec{x}_2$ .*

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

**Theorem 22.**  $\text{FO}(\subseteq) \leq_{\text{fin}} \text{FO}(\approx)$ .

*Proof.* Since  $\text{FO}(\subseteq) \equiv \text{FO}(\bowtie)$  ([10]), it suffices to show that  $\text{FO}(\bowtie) \leq_{\text{fin}} \text{FO}(\approx)$ . By Proposition 13 we may restrict attention to weighted semantics. Let  $\phi \in \text{FO}(\bowtie)$  be a sentence, and let  $k$  be the number of disjunctions and quantifiers in  $\phi$ . Let  $\phi^*$  be obtained from  $\phi$  by replacing all equiextension atoms of the form  $\vec{x}_1 \bowtie \vec{x}_2$  with  $\psi^k(\vec{x}_1, \vec{x}_2)$ . Suppose  $\mathfrak{A} \models_{\mathbb{X}}^w \phi$  for some finite structure  $\mathfrak{A}$  with universe of size  $n$  and a weighted team  $\mathbb{X}$ . Since  $\text{FO}(\bowtie)$  is insensitive to assignment weights, the satisfaction of  $\phi$  by  $\mathbb{X}$  is witnessed by weakly uniform weighted teams  $\mathbb{Y}$  in which the assignment weights are at least  $\frac{|\mathbb{Y}|}{n^k}$ . It follows by the previous lemma and a simple inductive argument that  $\mathfrak{A} \models_{\mathbb{X}}^w \phi^*$ . The converse direction follows similarly by the previous lemma.  $\square$

Consequently, probabilistic inclusion logic captures P, since this holds already for inclusion logic [11]. Another consequence is an alternative proof, through probabilistic inclusion logic (Theorem 22) and linear programs (Corollary 20, Theorem 4), for the polynomial-time upper bound of the data complexity of inclusion logic. For this, note also that distribution quantification is clearly expressible in  $\text{ESO}_{\mathbb{R}}[\leq, \text{SUM}]$ .

**Corollary 23.** *Sentences of  $\text{FO}(\approx)$  capture P on finite ordered structures.*

Theorem 22 also extends to formulae over uniform teams. Recall that a function  $f$  is uniform if  $f(s) = f(s')$  for all  $s, s' \in \text{supp}(f)$ .

**Theorem 24.**  $\text{FO}(\subseteq) \leq \text{FO}(\approx)$  over uniform probabilistic teams.

*Proof.* Let  $\phi$  be an  $\text{FO}(\subseteq)$  formula,  $\mathfrak{A}$  a finite structure, and  $\mathbb{X}$  a uniform probabilistic team.

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

## 6 Conclusion

We studied the expressivity and complexity of logics with probabilistic team semantics and fragments of additive  $\text{ESO}_{\mathbb{R}}$ . The following picture emerged:

$$\begin{aligned} \text{FO}(\approx) &\equiv \text{almost conjunctive } L\text{-}(\ddot{\exists}^* \forall^*)_{d[0,1]}[\leq, \text{SUM}, 0, 1] \\ &< L\text{-}\text{ESO}_{[0,1]}[\leq, +, 0, 1] \equiv \text{FO}(\approx, =(\dots)), \end{aligned}$$

where the strict inclusion was shown in [16]. Moreover  $\text{FO}(\approx)$  captures P on finite ordered structures and  $\text{FO}(\approx, =(\dots))$  captures NP on finite structures. Its worth to note that almost conjunctive  $(\ddot{\exists}^* \exists^* \forall^*)_{\mathbb{R}}[\leq, +, \text{SUM}, 0, 1]$  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.

## References

1. 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.
2. Lenore Blum, Felipe Cucker, Michael Shub, and Steve Smale. *Complexity and Real Computation*. Springer-Verlag, Berlin, Heidelberg, 1997.
3. Peter Bürgisser and Felipe Cucker. Counting complexity classes for numeric computations II: algebraic and semialgebraic sets. *J. Complexity*, 22(2):147–191, 2006.
4. 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.
5. Felipe Cucker and Klaus Meer. Logics which capture complexity classes over the reals. *J. Symb. Log.*, 64(1):363–390, 1999.
6. George B. Dantzig. *Linear Programming and Extensions*. Princeton University Press, 1963.
7. 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.
8. 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.
9. Pietro Galliani. Game Values and Equilibria for Undetermined Sentences of Dependence Logic. MSc Thesis. ILLC Publications, MoL–2008–08, 2008.
10. 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.
11. 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.
12. Erich Grädel and Yuri Gurevich. Metafinite model theory. *Inf. Comput.*, 140(1):26–81, 1998.
13. 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.
14. 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.
15. 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.
16. Miika Hannula, Åsa 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.
17. 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.

18. Uffe Flarup Hansen and Klaus Meer. Two logical hierarchies of optimization problems over the real numbers. *Math. Log. Q.*, 52(1):37–50, 2006.
19. 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.
20. 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.
21. Charles Jordan and Lukasz Kaiser. Machine learning with guarantees using descriptive complexity and SMT solvers. *CoRR*, abs/1609.02664, 2016.
22. Paris C. Kanellakis, Gabriel M. Kuper, and Peter Z. Revesz. Constraint query languages. *J. Comput. Syst. Sci.*, 51(1):26–52, 1995.
23. L. G. Khachiyan. A polynomial algorithm in linear programming. *Dokl. Akad. Nauk SSSR*, 244:1093–1096, 1979.
24. Pascal Koiran. Computing over the reals with addition and order. *Theor. Comput. Sci.*, 133(1):35–47, 1994.
25. 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.
26. Juha Kontinen and Jouko Väänänen. On definability in dependence logic. *Journal of Logic, Language and Information*, 3(18):317–332, 2009.
27. 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.
28. Klaus Meer. Counting problems over the reals. *Theor. Comput. Sci.*, 242(1-2):41–58, 2000.
29. Marcus Schaefer and Daniel Stefankovic. Fixed points, nash equilibria, and the existential theory of the reals. *Theory Comput. Syst.*, 60(2):172–193, 2017.
30. 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.
31. Jouko Väänänen. *Dependence Logic*. Cambridge University Press, 2007.
32. Steffen van Bergerem and Nicole Schweikardt. Learning concepts described by weight aggregation logic. *CoRR*, abs/2009.10574, 2020.

## A Proof of Theorem 8

*Proof.* Clearly  $\text{NP} \leq \text{BP}(\text{NP}_{\text{add}}^0)$ ; a Boolean guess for an input  $x$  can be constructed by comparing to zero each component of a real guess  $y$ , 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  $\text{BP}(\text{NP}_{\text{add}}^0)$ ; we need to show that  $L$  belongs also to  $\text{NP}$ . Let  $M$  be a BSS machine such that its running is bounded by some polynomial  $p$ , and for all Boolean inputs  $x \in \{0, 1\}^*$ ,  $x \in L$  if and only if there is  $y \in \mathbb{R}^{p(|x|)}$  such that  $M$  accepts  $(x, y)$ .

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

$$\sum_{j=1}^{p(n)} a_{ij} y_j \leq 0 \quad (1 \leq i \leq m) \quad \text{and} \quad \sum_{j=1}^{p(n)} b_{ij} y_j < 0 \quad (1 \leq i \leq l), \quad (4)$$

where  $a_{ij} \in \mathbb{Z}$ , such that  $y$  is a (real-valued) solution to  $\mathcal{S}$  if and only if  $M$  accepts  $(x, y)$  with respect to the outcomes  $z$ . In (4), the variables  $y_j$  stand for elements of the real guess  $y$ , 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  $z$ .

Without loss of generality we may assume additional constraints of the form  $y_j \geq 0$  ( $1 \leq j \leq p(n)$ ) (cf. [6, p. 86]). Transform then  $\mathcal{S}$  to another system of inequalities obtained from  $\mathcal{S}'$  by replacing strict inequalities in (4) 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 [23], the algorithm runs in polynomial time. Clearly, the algorithm accepts  $x$  for some guess  $z$  if and only if  $x \in L$ .  $\square$

## B Proof of Lemma 16

*Proof.* By Proposition 13, we may use weighted semantics (Definition 10). We may assume that existential second-order quantification, with respect to universe  $A$  and arity  $n$ , ranges over  $f : A^n \rightarrow [0, 1]$  such that  $\sum_{\vec{a} \in A^n} f(\vec{a}) \leq 1$ ; such function variables  $f$  can be simulated by  $(n + 1)$ -ary distribution variables  $d$  whose occurrences are of the form  $d(t_1, \dots, t_n, c)$ , where  $t_1, \dots, t_n$  are numerical terms. Then, a straightforward



induction shows that for all structures  $\mathfrak{A}$  and non-empty weighted teams  $\mathbb{X}: X \rightarrow [0, 1]$ , with variable domain  $\vec{x}$ , such that  $|\mathbb{X}| \leq 1$ ,

$$\mathfrak{A} \models_{\mathbb{X}}^w \phi(\vec{x}) \iff (\mathfrak{A}, f_{\mathbb{X}}) \models \phi^*(f). \quad (5)$$

Furthermore, the extra constants  $c$  and  $d$  can be discarded. Define

$$\begin{aligned} \psi(f) := \exists f' \Big( & \forall \vec{x} (\text{SUM}_{c,d} f'(\vec{x}, c, d) = f(\vec{x}) \wedge \forall c d c' d' f'(\vec{x}, c, d) = f'(\vec{x}, c', d')) \wedge \\ & \forall c d (c \neq d \rightarrow \phi^{**}(f')) \Big), \end{aligned}$$

where  $\phi^{**}(f')$  is obtained from  $\phi^*(f)$  by replacing function terms  $g(t_1, \dots, t_n)$  with  $g'(t_1, \dots, t_n, c, d)$ , and function variables and symbols  $g$  with  $g'$ , where the arity of the latter increases that of former by two. The intuition is that  $f'$  takes a scaled copy of  $f$ , and by Proposition 14 each copy maintains a correct interpretation of  $\phi(\vec{x})$ . Now  $\psi(f)$  contains only universal function and existential first-order quantifiers. 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  $\psi^*(f)$  which satisfies (5) and is in  $L-(\exists^* \forall^*)_{d[0,1]}[=, \text{SUM}, 0, 1]$ .

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

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

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.

□

## C Proof of Lemma 18

The original proof of the lemma ([8, Lemma 3]) also included a case for multiplication, which is here removed. The proof below includes a real constant 0, which can be expressed using the construction below: A formula  $\theta$  using a real constant 0 can be equivalently expressed as follows:

$$\exists f \exists h \forall x \forall y \forall z \left( f(x) = h(x, x) \wedge (y = z \vee \theta(h(y, z)/0)) \right)$$

In [8, Lemma 3] a different construction was used for expressing the real constant 0, which then lead to a circular definition.

*Proof.* First we define for each second sort term  $i(\vec{x})$  a special formula  $\theta_i$  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  $\theta_i$  is defined as  $f_i(\vec{u}) = g(\vec{u})$ . (We may interpret  $g(\vec{u})$  as  $\text{SUM}_{\emptyset} g(\vec{u})$ ).

- If  $i(\vec{u})$  is  $\text{SUM}_{\vec{v}}j(\vec{u}\vec{v})$ , then  $\theta_i$  is defined as  $\theta_j \wedge f_i(\vec{u}) = \text{SUM}_{\vec{v}}f_j(\vec{u}\vec{v})$ .

The translation  $\phi \mapsto \phi^*$  then proceeds recursively on the structure of  $\phi$ .

- (i) If  $\phi$  is  $i(\vec{u}) = j(\vec{v})$ , then  $\phi^*$  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  $\phi$  is an atom or negated atom of the first sort, then  $\phi^* := \phi$ .
- (iii) If  $\phi$  is  $\psi_0 \circ \psi_1$  where  $\circ \in \{\vee, \wedge\}$ ,  $\psi_0$  is  $\exists \vec{f}_0 \forall \vec{x}_0 \theta_0$ , and  $\psi_1$  is  $\exists \vec{f}_1 \forall \vec{x}_1 \theta_1$ , then  $\phi^*$  is defined as  $\exists \vec{f}_0 \vec{f}_1 \forall \vec{x}_0 \vec{x}_1 (\theta_0 \circ \theta_1)$ .
- (iv) If  $\phi$  is  $\exists y \psi$  where  $\psi^*$  is  $\exists \vec{f} \forall \vec{x} \theta$ , then  $\phi^*$  is defined as  $\exists g \exists \vec{f} \forall \vec{x} \forall y (g(y) = 0 \vee \theta)$ .
- (v) If  $\phi$  is  $\forall y \psi$  where  $\psi^*$  is  $\exists \vec{f} \forall \vec{x} \theta$ , then  $\phi^*$  is defined as

$$\exists \vec{f}^* \exists \vec{f}_{\text{id}} \exists d \forall y y' \forall \vec{x} (d(y) = d(y') \wedge \bigwedge_{f^* \in \vec{f}^*} \text{SUM}_{\vec{x}} f^*(y, \vec{x}) = d(y) \wedge \theta^*)$$

where  $\vec{f}^*$  is obtained from  $\vec{f}$  by replacing each  $f$  from  $\vec{f}$  with  $f^*$  such that  $\text{ar}(f^*) = \text{ar}(f) + 1$ ,  $\vec{f}_{\text{id}}$  introduces new function symbol for each multiplication in  $\theta$ , and  $\theta^*$  is obtained by replacing all second sort identities  $\alpha$  of the form  $f_i(\vec{u}\vec{v}) = f_j(\vec{u}) \times f_k(\vec{v})$  with

$$f_\alpha(y, \vec{u}\vec{v}) = d(y) \times f_i^*(y, \vec{u}\vec{v}) \wedge f_\alpha(y, \vec{u}\vec{v}) = f_j^*(y, \vec{u}) \times f_k^*(y, \vec{v})$$

and  $f_i(\vec{u}) = \text{SUM}_{\vec{v}} f_j(\vec{u}\vec{v})$  with  $f_i^*(y, \vec{u}) = \text{SUM}_{\vec{v}} f_j^*(y, \vec{u}\vec{v})$

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

It is straightforward to check that  $\phi^*$  is of the correct form and equivalent to  $\phi$ . What happens in (v) is that instead of guessing for all  $y$  some distribution  $f_y$  with arity  $\text{ar}(f)$ , we guess a single distribution  $f^*$  with arity  $\text{ar}(f) + 1$  such that  $f^*(y, \vec{u}) = \frac{1}{|A|} \cdot f_y(\vec{u})$  where  $A$  is the underlying domain of the structure. This is described by the existential quantification of a unary uniform distribution  $d$  such that for all fixed  $y$ ,  $\text{SUM}_{\vec{u}} f^*(y, \vec{u})$  is  $d(y)$ . Then note that  $f_y(\vec{u}) = g_y(\vec{u}') \cdot h_y(\vec{u}'')$  iff  $\frac{1}{|A|} \cdot f^*(y, \vec{u}) = g^*(y, \vec{u}') \cdot h^*(y, \vec{u}'')$  iff  $d(y) \cdot f^*(y, \vec{u}) = g^*(y, \vec{u}') \cdot h^*(y, \vec{u}'')$ . For identities over SUM, the reasoning is analogous.

## D Proof of Lemma 19

*Proof.* By Proposition 13 we can use weighted semantics in this proof. 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  $\vec{y}_1, \dots, \vec{y}_n$ ,

$$\mathbb{A} \models_{\mathbb{Y}}^w \Theta \text{ iff } (\mathbb{A}, f, f_1, \dots, f_n) \models \theta(\vec{a}) \text{ for all } \vec{a} \in M, \quad (6)$$

where  $f_i := \mathbb{X}' \upharpoonright \vec{y}_i$ . Showing this suffices, for it implies that  $\mathbb{A} \models_{\mathbb{X}}^w \Phi$  iff  $\mathbb{A} \models^w \phi(f)$ .

We show the claim by structural induction on the construction of  $\Theta$ .

- (1) This item is self-evident.

- (2) Assume first that for all  $\vec{a} \in M$ , we have  $(\mathfrak{A}, f, f_1, \dots, f_n) \models \theta(\vec{a})$ , that is,  $f_i(\vec{a}_i) = \text{SUM}_{\vec{x}_k} f_j(\vec{x}_k \vec{a}_l)$ . To show that  $\mathbb{Y}$  satisfies  $\Theta$ , let  $\mathbb{Z}$  be an extension of  $\mathbb{Y}$  to variables  $\alpha$  and  $\beta$  such that it satisfies the first two conjuncts of (2). 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  $\alpha \approx \beta$ . Now, the following chain of equalities hold:

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

For the third equality, note that  $\vec{x}$  and  $\vec{y}_i$  are independent because  $\vec{x}$  is quantified universally after  $\vec{y}_i$ . Thus  $\alpha$  and  $\beta$  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  $\alpha$  or  $\beta$  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  $\Theta$ .

For the converse direction, assume that  $\mathbb{Y}$  satisfies  $\Theta$ , and let  $\mathbb{Z}$  be an extension of  $\mathbb{Y}$  to  $\alpha$  and  $\beta$  satisfying (2). Then for all  $\vec{a} \in M$ ,  $\mathbb{Z}_{\vec{x}=\vec{a}}$  satisfies  $\alpha \approx \beta$  and thereby for all  $\vec{a} \in M$ ,

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

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

- (3) This item is self-evident.
- (4) Assume  $(\mathfrak{A}, f, f_1, \dots, f_n) \models \theta_0 \vee \theta_1$  for all  $\vec{a} \in M$ . Then  $M$  can be partitioned to disjoint  $M_0$  and  $M_1$  such that  $(\mathfrak{A}, f, f_1, \dots, f_n) \models \theta_i$  for all  $\vec{a} \in M_i$ . 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  $\mathfrak{A} \models_{\mathbb{Y}_i}^w \Theta_i$ , 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  $\mathfrak{A} \models_{\mathbb{Z}_{\vec{z}=c}}^w \theta_0$  and  $\mathfrak{A} \models_{\mathbb{Z}_{\vec{z}=d}}^w \theta_1$  by Proposition 14. We conclude that  $\mathbb{Z}$  satisfies  $(\theta_0 \wedge z = c) \vee (\theta_1 \wedge z = d)$ , and thus  $\mathbb{Y}$  satisfies  $\Theta$ . The converse direction is shown analogously. If  $\phi(f)$  is almost conjunctive, then we use the alternative translation which avoids the use of dependence atoms. This concludes the proof.  $\square$

## E Proof of Lemma 21

*Proof.* (i) Fix a tuple of values  $\vec{b}$  of length  $m$ . We restrict attention to  $\mathbb{Y} = \mathbb{X}[\vec{b}/\vec{u}]$ . It suffices to show that  $\mathbb{Y}$  satisfies

$$\begin{aligned} \exists v_1 v_2 \forall \vec{z}' \exists \vec{z} ((\vec{x}_1 = \vec{b} \leftrightarrow v_1 = y) \wedge (\vec{x}_2 = \vec{b} \leftrightarrow v_2 = y) \wedge \\ (\vec{z}' = \vec{c} \rightarrow \vec{z} = \vec{c}) \wedge (\neg \vec{z} = \vec{c} \vee v_1 \approx v_2)), \end{aligned} \quad (7)$$

obtained from (3) by fixing  $\vec{u} \mapsto \vec{b}$  and  $y \mapsto c$ , where  $c$  is some value in  $\vec{b}$ , and by removing  $\vec{u}$  from the marginal identity atom. 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  $\mathbb{Y}' : Y' \rightarrow [0, 1]$  the probabilistic team where  $Y'$  consists of these extensions and the weights are inherited from  $\mathbb{Y}$ .

Next, we first consider the existential quantification of  $\vec{z}$  and then show how the universal quantification of  $\vec{z}'$  can be fitted in. Let  $w_1$  be the total weight of those assignments  $s \in Y'$  satisfying  $s(\vec{x}_1) = \vec{b}$  and  $s(\vec{x}_2) \neq \vec{b}$ , and similarly let  $w_2$  be the total weight of those assignments  $s \in Y'$  satisfying  $s(\vec{x}_2) = \vec{b}$  and  $s(\vec{x}_1) \neq \vec{b}$ . By assumption  $w_1$  and  $w_2$  are either both zero or both at least  $\frac{|\mathbb{X}|}{n^k}$ . In the first case, allocate the full weight of  $s' \in Y'$  to the extension  $s'(\vec{c}/\vec{z})$ . In the second case, there are three options:

- (i) if  $s'(v_1) = c$  and  $s'(v_2) = d$ , allocate respectively  $w_2$  and  $1 - w_2$  of the weight of  $s'$  to  $s'(\vec{c}/\vec{z})$  and  $s'(\vec{d}/\vec{z})$ ,
- (ii) if  $s'(v_1) = d$  and  $s'(v_2) = c$ , allocate respectively  $w_1$  and  $1 - w_1$  of the weight of  $s'$  to  $s'(\vec{c}/\vec{z})$  and  $s'(\vec{d}/\vec{z})$ , or
- (iii) otherwise, allocate the full weight of  $s'$  to  $s'(\vec{c}/\vec{z})$ .

Concluding, the weight of those assignments that map  $(v_1, v_2)$  to  $(c, d)$  is  $w_1 w_2$ , and the weight of those assignments that map  $(v_1, v_2)$  to  $(d, c)$  is also  $w_1 w_2$ . All other assignments map  $(v_1, v_2)$  to  $(c, c)$  or  $(d, d)$ . Thus it follows that  $v_1 \approx v_2$  holds. Finally, we return to the universal quantification of  $\vec{z}'$ . Since  $w_1$  and  $w_2$  are at least  $\frac{|\mathbb{X}|}{n^k}$ , and universal quantification distributes  $\frac{|\mathbb{X}|}{n^k}$  of the weight of  $s'$  to  $s'(\vec{c}/\vec{z}')$ , the weight of  $s'$  can be distributed in such a way that both the conditions (i)-(iii) and the formula  $\vec{z}' = \vec{c} \rightarrow \vec{z} = \vec{c}$  simultaneously hold. This concludes the proof of case (1).

(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 (7). It follows that the total weight of assignments in  $\mathbb{Z}$  that map  $v_1$  to  $c$  is positive. Consequently, by  $\vec{z}' = \vec{c} \rightarrow \vec{z} = \vec{c}$  where  $\vec{z}'$  is universally quantified, a positive fraction of these assignments maps also  $\vec{z}$  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.  $\square$