Dependence Logic vs. Constraint Satisfaction

Lauri Hella[*1] **and Phokion G. Kolaitis**[†2]

1 **School of Information Sciences, University of Tampere, Finland**
lauri.hella@uta.fi
2 **University of California Santa Cruz and IBM Research – Almaden, USA**
kolaitis@cs.ucsc.edu

Abstract

During the past decade, dependence logic has emerged as a formalism suitable for expressing and analyzing notions of dependence and independence that arise in different scientific areas. The sentences of dependence logic have the same expressive power as those of existential second-order logic, hence dependence logic captures NP on the class of all finite structures. In this paper, we identify a natural fragment of universal dependence logic and show that, in a precise sense, it captures constraint satisfaction. This tight connection between dependence logic and constraint satisfaction contributes to the descriptive complexity of constraint satisfaction and elucidates the expressive power of universal dependence logic.

1998 ACM Subject Classification F.4.1 Mathematical Logic, F.1.3 Complexity Measures and Classes

Keywords and phrases Dependence logic, constraint satisfaction, computational complexity, expressive power

Digital Object Identifier 10.4230/LIPIcs.CSL.2016.14

1 Introduction

Dependence logic is a formalism for expressing and analyzing notions of dependence and independence that are encountered across different areas of computer science and mathematics, from functional dependencies in relational databases to independence in linear algebra and in probability theory. Even though its origins can be traced back to Henkin quantifiers [10] and to independence-friendly logic [11], dependence logic was fully developed by Väänänen in his monograph [17], which became the catalyst for numerous subsequent investigations (see, e.g., [6, 7, 8, 13, 14]). The syntax of dependence logic uses *dependence atoms* as the main building blocks; these atoms assert that a functional dependency between variables holds, i.e., that a certain variable is a function of some other variables. The semantics of dependence logic uses sets of assignments, called *teams*, instead of single assignments of values to variables. In terms of expressive power and as regards sentences, dependence logic has the same expressive power as existential second-order logic [13]. Combined with Fagin's Theorem [4], this result implies that, on classes of finite structures, the sentences of dependence logic can express precisely all decision problems in NP.

Constraint satisfaction comprises a set of algorithmic problems that are ubiquitous in several different areas of computer science. An influential paper by Feder and Vardi [5] provided the impetus for an in-depth and still ongoing investigation of the connections

* The research of Lauri Hella was partially supported by a Professor Pool's Grant of the Finnish Cultural Foundation.
† The research of Phokion Kolaitis was partially supported by NSF Grant IIS-1217869.

25th EACSL Annual Conference on Computer Science Logic (CSL 2016).
Editors: Jean-Marc Talbot and Laurent Regnier; Article No. 14; pp. 14:1–14:17
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

between constraint satisfaction, computational complexity, logic, and universal algebra (see, e.g., [1, 9]). Feder and Vardi argued convincingly that, in its most general form, constraint satisfaction can be identified with the Homomorphism Problem: given two relational structures $\mathfrak{A}$ and $\mathfrak{B}$, is there a homomorphism from $\mathfrak{A}$ to $\mathfrak{B}$? Clearly, the Homomorphism Problem is NP-complete, since it contains, for example, 3-Satisfiability as a special case. Moreover, each fixed relational structure $\mathfrak{B}$ gives rise to the *non-uniform* constraint satisfaction problem CSP($\mathfrak{B}$): given a relational structure $\mathfrak{A}$, is there a homomorphism from $\mathfrak{A}$ to $\mathfrak{B}$? The computational complexity of each such problem depends on the structure $\mathfrak{B}$. Feder and Vardi conjectured that the family of all constraint satisfaction problems CSP($\mathfrak{B}$) exhibits the following dichotomy: for each $\mathfrak{B}$, either CSP($\mathfrak{B}$) is NP-complete or CSP($\mathfrak{B}$) is solvable in polynomial time. This conjecture remains open to date, in spite of concerted efforts by different groups of researchers that, so far, have established only special cases of it.

Feder and Vardi [5] also investigated the descriptive complexity of constraint satisfaction. To this effect, they identified a fragment of existential second-order logic, called *monadic monotone strict* NP *without inequality* or, in short, MMSNP, and showed that it captures, in a precise sense, the family CSP($\mathfrak{B}$) of all non-uniform constraint satisfaction problems. MMSNP consists of all sentences of existential second order logic that have the following properties (where it is assumed that all negation symbols occurring in the sentences have been pushed inward, so that they apply to atomic formulas only): (a) all second-order quantifiers are monadic; (b) all first-order quantifiers are universal; (c) no inequalities occur in the formula; (d) all occurrences of relation symbols from the underlying vocabulary are preceded by the negation symbol. MMSNP captures constraint satisfaction in the following way. First, it is easy to see that if $\mathfrak{B}$ is a relational structure, then CSP($\mathfrak{B}$) is expressible in MMSNP. Second, Feder and Vardi showed that every MMSNP-expressible problem is equivalent to a CSP($\mathfrak{B}$), for some relational structure $\mathfrak{B}$, under polynomial-time reductions (originally, this equivalence was proved under randomized polynomial-time reductions, which, however, were subsequently derandomized [15]). Note that Feder and Vardi also showed that if one of the aforementioned properties (a), (b), (c), (d) defining MMSNP is dropped, then every problem in NP is equivalent under polynomial-time reductions to a problem in the resulting fragment of existential second-order logic. Combined with Ladner's Theorem [16], this implies that if one of these four properties is dropped, then the resulting fragment can express decision problems that are neither NP-complete, nor solvable in polynomial time (unless P = NP).

As seen from the preceding discussion, dependence logic captures existential second-order logic, while constraint satisfaction is captured by a proper fragment of existential second-order logic. This state of affairs gives rise to the following question: is there a natural fragment of dependence logic that captures constraint satisfaction? In this paper, we show that this is indeed the case. In fact, we identify a fragment of a variant of dependence logic consisting of universal sentences and show that it can capture, in a precise sense, constraint satisfaction. In what follows in this section, we present a high-level description of our main results.

The building blocks of dependence logic, as developed by Väänänen, are dependence atoms $\mathrm{dep}(\boldsymbol{x}; y)$, where $\boldsymbol{x}$ is a tuple of variables and y is a single variable. A team (i.e., a set of assignments) satisfies such an atom if whenever two assignments in the team agree on the variables in $\boldsymbol{x}$, they must also agree on the variable y. Here, we introduce a variant of dependence atoms, which we call *uniform* dependence atoms; they are expressions of the form $\mathrm{udep}(x_1, \ldots, x_n; \alpha_1, \ldots, \alpha_n)$ with the following semantics: a team T satisfies $\mathrm{udep}(x_1, \ldots, x_n; \alpha_1, \ldots, \alpha_n)$ if there is a unary function f such that for every assignment s in T, we have that $s(\alpha_i) = f(s(x_i))$, for $1 \leq i \leq n$. Even though uniform dependence atoms have not been studied in their own right in earlier work on dependence logic, we believe that they are very natural as they express scenarios in which n different observers use sensors or measuring instruments to collect data in different sites, and then each observer applies the same function to the data collected to obtain a value. As a concrete example, each x_i may represent a list of temperature values collected at site i at regular intervals of time each day, while α_i may stand for the maximum temperature at site i.

We consider *k-valued* uniform dependence atoms in which the variables $\alpha_1, \ldots, \alpha_n$ take values in a domain with k elements, for some fixed $k \geq 1$. We define the *universal monotone uniform dependence logic* $\forall$-MUD$[k]$ as the closure under universal quantification of all quantifier-free formulas that contain all k-valued uniform dependence atoms, all equalities between k-valued variables and constants, and all negated relational atoms, and are closed under disjunctions and conjunctions. The semantics of the logic $\forall$-MUD$[k]$ are given using teams as in (standard) dependence logic.

Our first main result asserts that every non-uniform constraint satisfaction problem CSP($\mathfrak{B}$) such that $\mathfrak{B}$ has a single relation is expressible by a sentence of $\forall$-MUD$[k]$, where k is the number of elements in the universe of $\mathfrak{B}$. Our second main result asserts that every sentence of $\forall$-MUD$[k]$, $k \geq 1$, is equivalent to a sentence of MMSNP. Since, as described earlier, every MMSNP-expressible problem is polynomial-time equivalent to some non-uniform constraint satisfaction problem [5] and since, as shown in [5] and in [15], every non-uniform constraint satisfaction problem is polynomial-time equivalent to some non-uniform constraint satisfaction problem on a structure with a single relation, our two main results imply that universal monotone uniform dependence logic captures, in a precise sense, all non-uniform constraint satisfaction problems CSP($\mathfrak{B}$).

Our results establish a tight connection between constraint satisfaction and a natural fragment of dependence logic. From the standpoint of constraint satisfaction, they contribute to the investigation of the descriptive complexity of constraint satisfaction. From the standpoint of dependence logic, they reveal that a dichotomy theorem for the computational complexity of the universal fragment of uniform dependence logic is as difficult as a dichotomy theorem for constraint satisfaction, which, to date, remains an elusive goal.

2 Background and Basic Notions

All structures considered in this paper are *finite* and *relational*. Thus, a vocabulary τ is a finite set of $\{R_1, \ldots, R_n\}$ of relation symbols, and the domain dom($\mathfrak{A}$) of each τ-structure $\mathfrak{A} = (\text{dom}(\mathfrak{A}), R_1^{\mathfrak{A}}, \ldots, R_n^{\mathfrak{A}})$ is assumed to be finite. However, to interpret k-valued dependence atoms, we add k constant symbols to the vocabulary; see Subsection 2.3 below. We will usually denote dom($\mathfrak{A}$) by A, dom($\mathfrak{B}$) by B, etc. For any integer $k \geq 1$, we will use the notation $[k] = \{1, \ldots, k\}$ throughout.

2.1 Constraint Satisfaction and MMSNP

A *homomorphism* between two τ-structures $\mathfrak{A}$ and $\mathfrak{B}$ is a function h from the universe A of $\mathfrak{A}$ to the universe B of $\mathfrak{B}$ such that for every relation symbol R of τ and every tuple $(a_1, \ldots, a_n)$ of elements of A, if $(a_1, \ldots, a_n) \in R^{\mathfrak{A}}$, then $(h(a_1), \ldots, h(a_n)) \in R^{\mathfrak{B}}$. Every τ-structure $\mathfrak{B}$ gives rise to the following constraint satisfaction problem CSP($\mathfrak{B}$):

Given a τ-structure $\mathfrak{A}$, is there a homomorphism from $\mathfrak{A}$ to $\mathfrak{B}$?

According to the usual practise, we identify the problem CSP($\mathfrak{B}$) with the class of its positive instances. Thus, we write $\mathfrak{A} \in$ CSP($\mathfrak{B}$), if the answer to the question above is "yes".

Clearly, each constraint satisfaction problem CSP($\mathfrak{B}$) is in NP. Moreover, numerous natural computational problems can be viewed as constraint satisfaction problems for a suitable choice of $\mathfrak{B}$. For example, if K_k is the *complete* graph on k nodes (i.e., K_k is the k*-clique*), $k \geq 2$, then CSP(K_k) is the k-Colorability problem. Furthermore, several variants of Satisfiability can be viewed as constraint satisfaction problems. We now give two such examples.

First, consider a vocabulary τ consisting of four ternary relation symbols R_0, R_1, R_2, R_3 and let $\mathfrak{B}$ be the τ-structure with universe $\{0,1\}$ and relations $R_0^{\mathfrak{B}} = \{0,1\}^3 \setminus \{(0,0,0)\}$, $R_1^{\mathfrak{B}} = \{0,1\}^3 \setminus \{(1,0,0)\}$, $R_2^{\mathfrak{B}} = \{0,1\}^3 \setminus \{(1,1,0)\}$, $R_3^{\mathfrak{B}} = \{0,1\}^3 \setminus \{(1,1,1)\}$. It is easy to see that CSP($\mathfrak{B}$) amounts to 3-Sat, where a 3CNF-formula φ is encoded as a τ-structure $\mathfrak{A}_\varphi$ with universe the set of its variables and where the relation $R_i^{\mathfrak{A}_\varphi}$ interpreting R_i consists of the triples of variables occurring in a clause with i negative literals, $i = 0,1,2,3$.

Next, consider a vocabulary τ consisting of a single ternary relation symbol R and let $\mathfrak{B}$ be the τ-structure with universe $\{0,1\}$ and relation $R^{\mathfrak{B}} = \{(1,0,0),(0,1,0),(0,0,1)\}$. It is easy to see that CSP($\mathfrak{B}$) amounts to Positive 1-in-3 Sat: given a 3CNF-formula φ consisting entirely of positive clauses, is there a truth assignment t such that, for every clause c of φ, the assignment t makes true exactly one of the three variables of c? Here, φ is encoded as a τ-structure $\mathfrak{A}_\varphi$ with universe the set of its variables and where the relation $R^{\mathfrak{A}_\varphi}$ consists of all triples (x,y,z) of variables such that $(x \vee y \vee z)$ is a clause of φ.

As mentioned in the Introduction, Feder and Vardi [5] conjectured that, for every fixed τ-structure $\mathfrak{B}$, either CSP($\mathfrak{B}$) is NP-complete or CSP($\mathfrak{B}$) is solvable in polynomial time. Moreover, they showed that, for every τ-structure $\mathfrak{B}$, there is a structure $\mathfrak{B}'$ over a vocabulary consisting of a single binary relation such that CSP($\mathfrak{B}$) and CSP($\mathfrak{B}'$) are equivalent via polynomial-time reductions. Thus, to settle the Feder-Vardi conjecture, it is enough to settle it for structures with a single binary relation (i.e., for directed graphs).

Every constraint satisfaction problem CSP($\mathfrak{B}$) is expressible by a sentence of existential second-order logic that also obeys certain syntactic restrictions. For example, as discussed earlier, CSP(K_3), which is the same as 3-Colorability, is expressible by the sentence $\exists B \exists R \exists G \forall x \forall y\, \theta$, where θ is the quantifier-free formula

$$(B(x) \vee R(x) \vee G(x)) \wedge \neg(B(x) \wedge R(x)) \wedge \neg(B(x) \wedge G(x)) \wedge \neg(R(x) \wedge G(x)) \\ \wedge \big(\neg E(x,y) \vee (\neg(B(x) \wedge B(y)) \wedge \neg(R(x) \wedge R(y)) \wedge \neg(G(x) \wedge G(y)))\big).$$

Similarly, Positive 1-in-3 Sat is expressible by the sentence $\exists S \forall x \forall y \forall z\, \eta$, where η is the formula

$$\neg R(x,y,z) \vee (S(x) \wedge \neg S(y) \wedge \neg S(z)) \vee (\neg S(x) \wedge S(y) \wedge \neg S(z)) \vee (\neg S(x) \wedge \neg S(y) \wedge S(z)).$$

The preceding sentences of existential second-order logic obey the following syntactic restrictions: (a) all second-order quantifiers are monadic; (b) all first-order quantifiers are universal; (c) no inequalities occur; (d) all occurrences of relation symbols from the underlying vocabulary τ are preceded by the negation symbol. Taken together, these syntactic restrictions define the fragment of existential second-order logic known as MMSNP.

MMSNP has strictly higher expressive power than constraint satisfaction, in the sense that there are problems that are definable by a MMSNP-sentence, but are not expressible as a CSP($\mathfrak{B}$) problem for any structure $\mathfrak{B}$ over the same vocabulary. Indeed, as pointed out in [15], the problem "given a graph, is it triangle-free?" is expressible by the sentence

$$\forall x \forall y \forall z (\neg E(x,y) \vee \neg E(x,z) \vee \neg E(y,z)),$$

which is in the first-order part of MMSNP, but there is no graph H such that a graph G is triangle-free if and only if there is a homomorphism from G to H. Towards a contradiction, assume that such a graph H exists. Erdös [3] showed that there are graphs of arbitrarily large girth and chromatic number. It follows that there is a graph G that is triangle-free (i.e., G has girth at least 4) and chromatic number bigger than that of H. Thus, G is triangle-free, but there is no homomorphism from G to H, else we could color G with at most the number of colors needed to color H.

As mentioned in the Introduction, however, Feder and Vardi [5] showed that every MMSNP-definable problem is equivalent under polynomial-time reductions to a constraint satisfaction problem $\mathrm{CSP}(\mathfrak{B})$, for some structure $\mathfrak{B}$ over the same vocabulary. Consequently, establishing a dichotomy theorem for the complexity of model checking MMSNP-sentences is precisely as hard as affirming the Feder-Vardi dichotomy conjecture for constraint satisfaction.

2.2 Dependence logic

Dependence logic D is the extension of first-order logic augmented with dependence atoms $\mathrm{dep}(x_1, \ldots, x_n; y)$. Since dependence atoms are allowed to occur only positively in formulas of D, it is natural assume that all formulas are in negation normal form. Thus, we define the syntax of D by the following grammar:

$$\begin{aligned} \varphi \quad ::= \quad & x_1 = x_2 \mid \neg\, x_1 = x_2 \mid R(x_1, \ldots, x_n) \mid \neg R(x_1, \ldots, x_n) \mid \\ & \mathrm{dep}(x_1, \ldots, x_n; y) \mid (\varphi_1 \wedge \varphi_1) \mid (\varphi_1 \vee \varphi_2) \mid \forall x \varphi \mid \exists x \varphi. \end{aligned}$$

The semantics of D is defined with respect to *teams*, i.e., sets of assignments, instead of single assignments. If $\mathfrak{A}$ is a structure with domain A and V is a set of first-order variables, then an *assignment* on $\mathfrak{A}$ is a function $s : V \to A$. A *team* on $\mathfrak{A}$ is a set T of assignments on some fixed set $V = \mathrm{dom}(T)$ of variables. In particular, if $V = \emptyset$, then there are two teams on $\mathfrak{A}$ with domain V: the empty team $\emptyset$, and the team $T = \{\emptyset\}$ consisting of the empty assignment $\emptyset : \emptyset \to A$.

To define the semantics of universal quantification, we use the following notation: $T[A/x] = \{s[a/x] \mid s \in T, a \in A\}$, where $s[a/x]$ is the assignment such that it agrees with s on all $y \in \mathrm{dom}(s) \setminus \{x\}$, and $s[a/x](x) = a$.

To define the semantics for existential quantification, we need the notion of a *choice function* $F : T \to A$. The idea is that F picks an element $F(s)$ from the domain A of a structure $\mathfrak{A}$ for each assignment s in a team T. The element $F(s)$ is then used to interpret a variable x, thus obtaining the new assignment $s[F(s)/x]$. We write $T[F/x]$ for the team $\{s[F(s)/x] \mid s \in T\}$ obtained from T by making this change to each $s \in T$.

▶ **Definition 1.** Let $\mathfrak{A}$ be a model and T a team on $\mathfrak{A}$. The truth relation $\mathfrak{A}, T \models \varphi$ for dependence logic is defined as follows.

$$\begin{aligned}
\mathfrak{A}, T \models x_1 = x_2 \quad & \iff \quad s(x_1) = s(x_2) \text{ for all } s \in T. \\
\mathfrak{A}, T \models \neg x_1 = x_2 \quad & \iff \quad s(x_1) \neq s(x_2) \text{ for all } s \in T. \\
\mathfrak{A}, T \models R(x_1, \ldots, x_n) \quad & \iff \quad (s(x_1), \ldots, s(x_n)) \in R^{\mathfrak{A}} \text{ for all } s \in T. \\
\mathfrak{A}, T \models \neg R(x_1, \ldots, x_n) \quad & \iff \quad (s(x_1), \ldots, s(x_n)) \notin R^{\mathfrak{A}} \text{ for all } s \in T. \\
\mathfrak{A}, T \models \mathrm{dep}(x_1, \ldots, x_n; y) \quad & \iff \quad \text{there is a function } f : A^n \to A \text{ such that} \\
& \qquad s(y) = f(s(x_1), \ldots, s(x_n)) \text{ for all } s \in T. \\
\mathfrak{A}, T \models \varphi \wedge \psi \quad & \iff \quad \mathfrak{A}, T \models \varphi \text{ and } \mathfrak{A}, T \models \psi. \\
\mathfrak{A}, T \models \varphi \vee \psi \quad & \iff \quad \text{there are } T', T'' \subseteq T \text{ such that } T \cup T' = T'', \\
& \qquad \mathfrak{A}, T' \models \varphi \text{ and } \mathfrak{A}, T'' \models \psi. \\
\mathfrak{A}, T \models \forall x \psi \quad & \iff \quad \mathfrak{A}, T[A/x] \models \psi. \\
\mathfrak{A}, T \models \exists x \psi \quad & \iff \quad \text{there is a function } F \colon T \to A \text{ s.t. } \mathfrak{A}, T[F/x] \models \psi.
\end{aligned}$$

The set $\mathrm{Fr}(\varphi)$ of free variables of a formula $\varphi \in \mathsf{D}$ is defined in the standard way. The formula φ is a *sentence* if $\mathrm{Fr}(\varphi) = \emptyset$. A sentence $\varphi \in \mathsf{D}$ is *true* in a structure $\mathfrak{A}$, in symbols $\mathfrak{A} \models \varphi$, if $\mathfrak{A}, \{\emptyset\} \models \varphi$.

Note that in the literature (see, e.g., [17]), the semantics of the dependence atom is usually stated in the following equivalent form:

$$\mathfrak{A}, T \models \mathrm{dep}(x_1, \ldots, x_n; y) \iff \text{for all } s, s' \in T, \text{ if } s(x_i) = s'(x_i) \text{ for all } i \in \{1, \ldots, n\}, \text{ then } s(y) = s'(y).$$

Note also that, in database terminology, $\mathfrak{A}, T \models \mathrm{dep}(x_1, \ldots, x_n; y)$ means that the team T, viewed as an n-ary relation, satisfies the functional dependency $x_1, \ldots, x_n \to y$.

We review here briefly the basic properties of dependence logic. The first property is that the team semantics for first-order formulas in D (i.e., formulas without dependence atoms) can be reduced to the standard Tarski semantics. We write $\mathfrak{A}, s \models \varphi$ if the first-order formula φ is satisfied by the assignment s in the structure $\mathfrak{A}$.

▶ **Fact 1** (Flatness, [17]). *Let φ be a formula of D without dependence atoms, and let $\mathfrak{A}$ be a structure and T a team on $\mathfrak{A}$. Then $\mathfrak{A}, T \models \varphi$ if and only if $\mathfrak{A}, s \models \varphi$ for all $s \in T$.*

The second property is that the semantics of every D-formula is downwards closed in the following sense.

▶ **Fact 2** (Downward closure, [17]). *Let φ be a formula of D. If T and T' are teams on a structure $\mathfrak{A}$ such that $\mathfrak{A}, T \models \varphi$ and $T' \subseteq T$, then $\mathfrak{A}, T' \models \varphi$.*

The formulas φ of D also have the desirable property that the truth of φ only depends on the interpretation of its free variables $\mathrm{Fr}(\varphi)$. We use here the notation $T \upharpoonright V = \{s \upharpoonright V \mid s \in T\}$ for a team T and a set V of variables.

▶ **Fact 3** (Locality, [17]). *Let φ be a formula of D with $\mathrm{Fr}(\varphi) = V$. If T is a team on a structure $\mathfrak{A}$ and $T' = T \upharpoonright V$, then $\mathfrak{A}, T \models \varphi$ if and only if $\mathfrak{A}, T' \models \varphi$.*

Finally, as mentioned in the Introduction, dependence logic D has the same expressive power as existential second-order logic Σ^1_1.

▶ **Fact 4** (D captures Σ^1_1, [17]). *For every sentence φ of D, there is an equivalent sentence ψ of Σ^1_1; vice versa, for every sentence ψ of Σ^1_1, there is an equivalent sentence φ of D.*

As a consequence of Fact 4 and Fagin's Theorem [4], dependence logic D captures the complexity class NP. In particular, this means that NP-complete problems, such as k-Colorability and k-Sat, $k \geq 3$, are expressible in D. Perhaps surprisingly, it turns out that the model-checking problem of D-formulas can be NP-complete already at the quantifier-free level. Specifically, Jarmo Kontinen [12] proved that the problem "does a team T on a structure $\mathfrak{A}$ (with empty vocabulary) satisfy the formula $\mathrm{dep}(x; y) \vee \mathrm{dep}(u; v) \vee \mathrm{dep}(u; v)$?" is NP-complete. On the other hand, he proved that the model-checking problem for the disjunction of any two dependence atoms is in $\mathsf{NLOGSPACE}$.

The complexity of model-checking for quantifier-free formulas of D has been further investigated by Durand et al. [2]. Extending the ideas of Kontinen [12], they give sufficient syntactic criteria for the tractability and the NP-completeness of such model-checking problems. In the present paper, we focus on the relationship between the universal fragment of dependence logic, constraint satisfaction problems and MMSNP, and unveil a tight connection.

2.3 Logics with k-valued variables

In the next subsection, we will define uniform k-valued dependence atoms. To do this, in addition to the usual first-order variables, we need a separate supply of k-valued variables. Furthermore, to interpret the k-valued variables, we will extend structures by a standard part consisting of the numbers $1, \dots, k$. Thus, if $\mathfrak{A} = (A, R_1^{\mathfrak{A}}, \dots, R_n^{\mathfrak{A}})$ is a τ-structure, then we define $\mathfrak{A}[k]$ to be the two-sorted structure $(\mathfrak{A}; [k], \underline{1}^{\mathfrak{A}}, \dots, \underline{k}^{\mathfrak{A}})$. Here $[k]$ is the domain of the second sort and $\underline{1}, \dots, \underline{k}$ are constant symbols over the second sort such that $\underline{i}^{\mathfrak{A}} = i$ for each $i \in [k]$.

We will use the Greek letters α, β, γ, with or without subscripts, as *k-valued* variables, while we will use x, y, u, v as ordinary *first-order* variables. The intuition is that k-valued variables always range over the second sort $[k]$ of a structure $\mathfrak{A}[k]$, while the first-order variables range over the domain A of $\mathfrak{A}$. We often use the boldface notation $\boldsymbol{x}$ ($\boldsymbol{\alpha}$, or $\boldsymbol{a}$) for a tuple $(x_1, \dots, x_n)$ of variables (a tuple $(\alpha_1, \dots, \alpha_n)$ of k-valued variables, or a tuple $(a_1, \dots, a_n)$ of elements, respectively). If not explicitly defined, the length n of the tuple will be clear from the context.

For logics with k-valued variables and team semantics, the notion of a team needs to be adapted. If $\mathfrak{A}$ is a structure, and V is a finite set of first-order and k-valued variables, then an assignment on $\mathfrak{A}[k]$ with domain V is a function $s : V \to A \cup [k]$ such that $s(x) \in A$ for each first-order variable $x \in V$ and $s(\alpha) \in [k]$ for each k-valued variable $\alpha \in V$. A *team* on $\mathfrak{A}[k]$ with domain V is a set T of assignments $s : V \to A \cup [k]$.

We will next introduce some useful notation.

▶ **Definition 2.** Let T be a team on a structure $\mathfrak{A}[k]$ with domain V.

- If $\boldsymbol{x} \in V^n$ and $\boldsymbol{\alpha} \in V^m$, then we use the notation $R_{T,\boldsymbol{x}\boldsymbol{\alpha}}$ for the $(n+m)$-ary relation $\{s(\boldsymbol{x}\boldsymbol{\alpha}) \mid s \in T\} \subseteq A^n \times [k]^m$.
- In case $m = 0$, we write simply $R_{T,\boldsymbol{x}} = \{s(\boldsymbol{x}) \mid s \in T\}$. Similarly, in case $n = 0$, we write $R_{T,\boldsymbol{\alpha}} = \{s(\boldsymbol{\alpha}) \mid s \in T\}$.
- Furthermore, if $\boldsymbol{a} \in A^n$, then $T[\boldsymbol{x} = \boldsymbol{a}]$ denotes the subteam $\{s \in T \mid s(\boldsymbol{x}) = \boldsymbol{a}\} \subseteq T$. Similarly, if $\boldsymbol{\ell} \in [k]^m$, then $T[\boldsymbol{\alpha} = \boldsymbol{\ell}]$ denotes the subteam $\{s \in T \mid s(\boldsymbol{\alpha}) = \boldsymbol{\ell}\} \subseteq T$.

Note that, in database terminology, $R_{T,\boldsymbol{x}\boldsymbol{\alpha}}$ is the *projection* $\pi_{\boldsymbol{x}\boldsymbol{\alpha}}(T)$ of the team T on the variables $\boldsymbol{x}\boldsymbol{\alpha}$, where T is viewed as a relation. Moreover, $T[\boldsymbol{x} = \boldsymbol{a}]$ is the *selection* $\sigma_{\boldsymbol{x}=\boldsymbol{a}}(T)$ of the team T, where T is viewed as a relation; similarly, $T[\boldsymbol{\alpha} = \boldsymbol{\ell}]$ is the selection $\sigma_{\boldsymbol{\alpha}=\boldsymbol{\ell}}(T)$.

To simplify the notation, henceforth we will denote the structures $\mathfrak{A}[k]$ simply by $\mathfrak{A}$. This should not cause any confusion, since it is always clear from the context, whether the symbol $\mathfrak{A}$ refers to a usual structure, or the extension of such structure with the second sort $[k]$.

2.4 Uniform k-valued dependence atoms

We are now ready to define the *uniform k-valued dependence atoms*, which we will use in the rest of the paper. These atoms differ from the standard dependence atoms in two ways: first, they are k-valued; second, the functional dependence is generated by a single unary function.

▶ **Definition 3.** If $\boldsymbol{x} = (x_1, \dots, x_n)$ is an n-tuple of first-order variables and $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n)$ is an n-tuple of k-valued variables, then $\mathrm{udep}[k](\boldsymbol{x}; \boldsymbol{\alpha})$ is an atomic formula with the semantics

$$\mathfrak{A}, T \models \mathrm{udep}[k](\boldsymbol{x}; \boldsymbol{\alpha}) \iff \begin{array}{l} \text{there is a function } f : A \to [k] \text{ such that} \\ s(\alpha_i) = f(s(x_i)), \text{ for all } i \in [n] \text{ and } s \in T. \end{array}$$

Note that in the case $n = 1$, the uniform k-valued dependence atom $\mathrm{udep}[k](x; \alpha)$ is equivalent with the k-valued version $\mathrm{dep}[k](x; \alpha)$ of the ordinary dependence atom $\mathrm{dep}(x; y)$.

The semantics of universal and existential quantification of k-valued variables can be defined in the same way as for quantification of first-order variables by defining $T[[k]/\alpha] = \{s[i/\alpha] \mid s \in T, i \in [k]\}$, and $T[G/\alpha] = \{s[G(s)/\alpha] \mid s \in T\}$ for a choice function $G : T \to [k]$. However, we will not consider existential quantification in this paper, as our main focus is on a quantifier-free fragment of the full logic with uniform k-valued dependence atoms, and its closure with respect to universal quantifiers.

▶ **Definition 4.** The *quantifier-free monotone dependence logic with uniform k-valued dependence atoms*, $\mathsf{QF\text{-}MUD}[k]$, is defined by the following grammar:

$$\varphi \quad ::= \quad \alpha = \underline{i} \mid \neg R(\boldsymbol{x}) \mid \mathrm{udep}[k](\boldsymbol{x}; \boldsymbol{\alpha}) \mid (\varphi_1 \wedge \varphi_2) \mid (\varphi_1 \vee \varphi_2), \quad \text{where } i \in [k].$$

Universal monotone dependence logic with uniform k-valued dependence atoms, $\forall\text{-}\mathsf{MUD}[k]$, is the extension of $\mathsf{QF\text{-}MUD}[k]$ defined by the grammar

$$\varphi \quad ::= \quad \psi \mid \forall x \varphi \mid \forall \alpha \varphi, \quad \text{where } \psi \in \mathsf{QF\text{-}MUD}[k].$$

The union of $\forall\text{-}\mathsf{MUD}[k]$ over all $k \geq 1$ is denoted by $\forall\text{-}\mathsf{MUD}[\omega]$. Similarly, $\mathsf{QF\text{-}MUD}[\omega]$ is the union of $\mathsf{QF\text{-}MUD}[k]$ over all $k \geq 1$.

Thus, analogously to MMSNP, the logics $\mathsf{QF\text{-}MUD}[k]$ and $\forall\text{-}\mathsf{MUD}[k]$ admit no inequalities and only negative occurrences of relation symbols in the vocabulary. Note that there is no need to include equalities of the form $\alpha = \beta$, since they can be expressed as $\bigvee_{i \in [k]} (\alpha = \underline{i} \wedge \beta = \underline{i})$. Furthermore, inequalities between k-valued variables are also expressible: $\alpha \neq \beta$ is equivalent to $\bigvee_{i \in [k]} \big(\alpha = \underline{i} \wedge \bigvee_{j \in [k], j \neq i} \beta = \underline{j}\big)$.

For the sake of completeness, we state here the definition of the semantics of $\forall\text{-}\mathsf{MUD}[k]$.

▶ **Definition 5.** Let $\mathfrak{A}$ be a structure and T a team on $\mathfrak{A}$. The *truth* relation $\mathfrak{A}, T \models \varphi$ for universal monotone uniform k-valued dependence logic is defined as follows.

$$\begin{array}{lcl}
\mathfrak{A}, T \models \alpha = \underline{i} & \iff & s(\alpha) = i \text{ for all } s \in T. \\
\mathfrak{A}, T \models \neg R(\boldsymbol{x}) & \iff & (s(x_1), \ldots, s(x_n)) \notin R^{\mathfrak{A}} \text{ for all } s \in T. \\
\mathfrak{A}, T \models \mathrm{udep}[k](\boldsymbol{x}; \boldsymbol{\alpha}) & \iff & \text{there is a function } f : A \to [k] \text{ such that} \\
 & & s(\alpha_i) = f(s(x_i)) \text{ for all } i \in [n] \text{ and } s \in T. \\
\mathfrak{A}, T \models \varphi \wedge \psi & \iff & \mathfrak{A}, T \models \varphi \text{ and } \mathfrak{A}, T \models \psi. \\
\mathfrak{A}, T \models \varphi \vee \psi & \iff & \text{there are } T', T'' \subseteq T \text{ such that } T' \cup T'' = T, \\
 & & \mathfrak{A}, T' \models \varphi \text{ and } \mathfrak{A}, T'' \models \psi. \\
\mathfrak{A}, T \models \forall x \varphi & \iff & \mathfrak{A}, T[A/x] \models \varphi. \\
\mathfrak{A}, T \models \forall \alpha \varphi & \iff & \mathfrak{A}, T[[k]/\alpha] \models \varphi.
\end{array}$$

Since dependence logic has the same expressive power as existential second-order logic, it is clear that uniform k-valued dependence atoms are definable in D (in the setting with k-valued variables). Indeed, it is straightforward to check that $\mathrm{udep}[k](x_1, \ldots, x_n; \alpha_1, \ldots, \alpha_n)$ is equivalent to the formula

$$\forall y \exists \beta \big(\mathrm{dep}[k](y; \beta) \wedge \bigwedge_{i \in [n]} (y = x_i \to \beta = \alpha_i)\big).$$

Note however, that this formula violates the syntactic restrictions of $\forall\text{-}\mathsf{MUD}[k]$ in two different ways: it contains existential quantification of a k-valued variable and inequalities between first-order variables.

As in the case of dependence logic D, a formula φ of $\forall\text{-}\mathsf{MUD}[k]$ is a sentence, if the set $\mathrm{Fr}(\varphi)$ of its free variables is empty. Furthermore, a sentence φ is true in a structure $\mathfrak{A}$, in symbols $\mathfrak{A} \models \varphi$, if $\mathfrak{A}, \{\emptyset\} \models \varphi$.

Clearly any $\forall$-MUD$[k]$-sentence φ is equivalent to a sentence of the form $\forall \boldsymbol{x} \forall \boldsymbol{\alpha} \psi$, where ψ is a QF-MUD$[k]$-formula. As a matter of fact, we can assume without loss of generality that φ is the *universal closure* of ψ, i.e., the tuple $\boldsymbol{x\alpha}$ is repetition-free and consists of the free variables of ψ. Using the truth conditions for universal quantification of first-order and k-valued variables repeatedly, we obtain the following simple connection between the semantics of φ and ψ:

- $\mathfrak{A} \models \varphi$ if and only if $\mathfrak{A}, F \models \psi$, where F is the team consisting of all assignments $s : V \to A \cup [k]$ with $V = \mathrm{Fr}(\psi)$.

We will call F the *full team* (on $\mathfrak{A}$ with domain V) in the sequel. If there is need to emphasize the domain V of F, we denote the full team by F_V.

The full team has a special role in the semantics of QF-MUD$[k]$ also in another way. It is straightforward to verify that Facts 2 and 3 (see Subsection 2.2) remain true for $\forall$-MUD$[k]$. Specifically, for every formula $\psi \in$ QF-MUD$[k]$, the following statements are true:

1. if $\mathfrak{A}, T \models \psi$ and $T' \subseteq T$, then $\mathfrak{A}, T' \models \psi$.
2. if $T' = T \restriction \mathrm{Fr}(\psi)$, then $\mathfrak{A}, T \models \psi$ if and only if $\mathfrak{A}, T' \models \psi$.

Thus, to decide whether a formula is satisfied by every team in a given structure, it suffices to check whether it is satisfied by the full team.

We summarize the two observations concerning the full team in the following lemma.

▶ **Lemma 6.** *Let ψ be a* QF-MUD$[k]$*-formula with $\boldsymbol{x}$ and $\boldsymbol{\alpha}$ as its free variables. Then the following statements are equivalent:*

1. $\mathfrak{A} \models \forall \boldsymbol{x} \forall \boldsymbol{\alpha} \psi$.
2. $\mathfrak{A}, F \models \psi$.
3. $\mathfrak{A}, T \models \psi$*, for every team T on $\mathfrak{A}$ with* $\mathrm{Fr}(\psi) \subseteq \mathrm{dom}(T)$

3 From Constraint Satisfaction to Dependence Logic

Our aim in this section is to prove that every constraint satisfaction problem CSP$(\mathfrak{B})$ is captured by a sentence of $\forall$-MUD$[\omega]$. To do this, we will prove that CSP$(\mathfrak{B})$ is definable in $\forall$-MUD$[\omega]$, assuming that $\mathfrak{B}$ is of the form $(B, R^{\mathfrak{B}})$, i.e., $\mathfrak{B}$ has only one relation. This suffices, since as mentioned in Subsection 2.1, every constraint satisfaction problem CSP$(\mathfrak{B})$ is equivalent, via polynomial-time reductions, to a CSP$(\mathfrak{B}')$ in which $\mathfrak{B}'$ is a structure with a single binary relation.

We start by observing that the truth of a $[k]$-valued uniform dependence atom on a given structure $\mathfrak{A}$ and a given team T implies the existence of a homomorphism between the two structures $(A, R_{T,\boldsymbol{x}})$ and $([k], R_{T,\boldsymbol{\alpha}})$.

▶ **Lemma 7.** *If* $\mathfrak{A}, T \models \mathrm{udep}[k](\boldsymbol{x}; \boldsymbol{\alpha})$*, then* $(A, R_{T,\boldsymbol{x}}) \in \mathrm{CSP}([k], R_{T,\boldsymbol{\alpha}})$.

Proof. Assume that $\mathfrak{A}, T \models \mathrm{udep}[k](\boldsymbol{x}; \boldsymbol{\alpha})$. Then there is a function $f : A \to [k]$ such that

- $f(s(x_i)) = s(\alpha_i)$ for all $i \in [n]$ and $s \in T$.

This condition implies that f is a homomorphism from $(A, R_{T,\boldsymbol{x}})$ to $([k], R_{T,\boldsymbol{\alpha}})$. Indeed, if $\boldsymbol{a} = (a_1, \ldots, a_n) \in R_{T,\boldsymbol{x}}$, then there exists $s \in T$ such that $s(x_i) = a_i$ for all $i \in [n]$. But then also $s(\alpha_i) = f(a_i)$ holds for all $i \in [n]$, whence $(f(a_1), \ldots, f(a_n)) \in R_{T,\boldsymbol{\alpha}}$. ◀

Note that the converse implication of Lemma 7 is not true. As an example, consider the team $T = \{s, s'\}$, where $s(x_1) = s'(x_1)$, $s(x_2) = s'(x_2)$, $s(\alpha_1) = s(\alpha_2) = 1$ and $s'(\alpha_1) = s'(\alpha_2) = 2$. Then the function $h : A \to [k]$ such that $h(a) = 1$ for all $a \in A$, is a homomorphism $(A, R_{T,x_1 x_2}) \to ([k], R_{T,\alpha_1 \alpha_2})$, but clearly $\mathfrak{A}, T \not\models \mathrm{udep}[k](x_1, x_2; \alpha_1, \alpha_2)$. Thus, uniform dependence atoms are different from *homomorphism* atoms.

For each positive integer n, let τ_n be a vocabulary consisting of a single n-ary relation symbol R. Let $\mathfrak{B} = ([k], R^{\mathfrak{B}})$ be a τ_n-structure, where $R^{\mathfrak{B}} \neq \emptyset$. The crucial step in the proof that CSP($\mathfrak{B}$) is expressible in $\forall$-MUD[k], is the following. Assume that $\mathfrak{A}$ is a τ_n-structure and T is a team on $\mathfrak{A}$ such that $\boldsymbol{a}\boldsymbol{\ell} \in R_{T,\boldsymbol{x}\boldsymbol{\alpha}}$ for all $\boldsymbol{a} \in R^{\mathfrak{A}}$ and $\boldsymbol{\ell} \in R^{\mathfrak{B}}$ and $R_{T,\boldsymbol{\alpha}} \subseteq R^{\mathfrak{B}}$. We define a formula $\theta_{R^{\mathfrak{B}}} \in$ QF-MUD[k] that is true in a subteam T' of T if and only if for each $\boldsymbol{a} \in R^{\mathfrak{A}}$, $R_{T'[\boldsymbol{x}=\boldsymbol{a}],\boldsymbol{\alpha}}$ contains all except exactly one of the tuples $\boldsymbol{\ell} \in R^{\mathfrak{B}}$. If the complement $T'' = T \setminus T'$ of T' satisfies the dependence atom udep[k]($\boldsymbol{x}; \boldsymbol{\alpha}$), it follows then from Lemma 7, that there is a homomorphism $\mathfrak{A} \to \mathfrak{B}$. Conversely, if there is a homomorphism $\mathfrak{A} \to \mathfrak{B}$, then T can be divided into subteams T' and T'' that satisfy $\theta_{R^{\mathfrak{B}}}$ and udep[k]($\boldsymbol{x}; \boldsymbol{\alpha}$), respectively. The precise statement and proof of this equivalence is given in Lemma 9, below.

Before defining $\theta_{R^{\mathfrak{B}}}$, we introduce some useful auxiliary formulas. If y and β are variables, we let $\chi_1(y, \beta) :=$ udep[k]($y; \beta$), and define recursively $\chi_{r+1}(y, \beta) := (\chi_r(y, \beta) \vee \chi_1(y, \beta))$. Thus, $\chi_r(y, \beta)$ is the disjunction of r copies of the dependence atom udep[k]($y; \beta$) with itself. Furthermore, define $\chi_0(y, \beta)$ to be a formula that is always false (e.g., $\beta = \underline{1} \wedge \beta = \underline{2}$). It is easy to show, by induction on r, that

$$\mathfrak{A}, T \models \chi_r(y, \beta) \quad \text{if and only if } |R_{T[y=a],\beta}| \leq r, \text{ for every } a \in R_{T,y}.$$

Next, for each nonempty relation $S \subseteq [k]^n$, we define a formula θ_S by induction on the arity n of S:

1. For $n = 1$, we let $\theta_S := \chi_{s-1}(x_1, \alpha_1)$, where $s = |S|$.
2. Let S be a nonempty n-ary relation on $[k]$, $n > 1$. Then there is a unique set $I \subseteq [k]$ and unique nonempty relations $S_\ell \subseteq [k]^{n-1}$, $\ell \in I$, such that $S = \bigcup_{\ell \in I} S_\ell \times \{\ell\}$. We define

$$\theta_S := \chi_{m-1}(x_n, \alpha_n) \vee \bigvee_{\ell \in I} (\alpha_n = \underline{\ell} \wedge \theta_{S_\ell}), \quad \text{where } m = |I|.$$

To illustrate the preceding notions, consider the structure $\mathfrak{B} = (\{0,1\}, R^{\mathfrak{B}})$, where $R^{\mathfrak{B}} = \{(1,0,0), (0,1,0), (0,0,1)\}$. As mentioned in Section 2.1, the constraint satisfaction problem CSP($\mathfrak{B}$) amounts to the satisfiability problem Positive 1-in-3 Sat. We are interested in computing the formula $\theta_{R^{\mathfrak{B}}}$. Clearly, $R^{\mathfrak{B}} = \{(0,1), (1,0)\} \times \{0\} \cup \{(0,0)\} \times \{1\}$. Thus, by taking $I = \{0,1\}$, $S_0 = \{(0,1), (1,0)\}$, and $S_1 = \{(0,0)\}$, we have that

$$\theta_{R^{\mathfrak{B}}} := \chi_1(x_3, \alpha_3) \vee (\alpha_3 = \underline{0} \wedge \theta_{S_0}) \vee (\alpha_3 = \underline{1} \wedge \theta_{S_1}).$$

We leave it to the reader to verify that, after unraveling further the definitions and eliminating disjuncts containing formulas that are always false, we have that

$$\theta_{R^{\mathfrak{B}}} := \text{udep}[2](x_3; \alpha_3) \vee (\alpha_3 = \underline{0} \wedge \text{udep}[2](x_2; \alpha_2)).$$

▶ **Lemma 8.** *Let $S \subseteq [k]^n$ be a nonempty relation, and let T be a team on a structure $\mathfrak{A}$.*

1. *If $\mathfrak{A}, T \models \theta_S$, then for every $\boldsymbol{a} \in R_{T,\boldsymbol{x}}$, we have that $S \setminus R_{T[\boldsymbol{x}=\boldsymbol{a}],\boldsymbol{\alpha}} \neq \emptyset$.*
2. *If $R_{T,\boldsymbol{\alpha}} \subseteq S$ and $f : A \to [k]$ is a function such that $(f(a_1), \ldots, f(a_n)) \in S \setminus R_{T[\boldsymbol{x}=\boldsymbol{a}],\boldsymbol{\alpha}}$ for all $\boldsymbol{a} \in R_{T,\boldsymbol{x}}$, then $\mathfrak{A}, T \models \theta_S$.*

Proof. Both claims are proved by induction on n.

1. If $n = 1$ and $\mathfrak{A}, T \models \theta_S$, then by the definition of θ_S and the observation above, $|R_{T[x_1=a],\alpha_1}| \leq |S| - 1$ for all $a \in R_{T,x_1}$. Thus, $S \setminus R_{T[x_1=a],\alpha_1} \neq \emptyset$ for all $a \in R_{T,x_1}$. Assume then that $S = \bigcup_{\ell \in I} S_\ell \times \{\ell\}$ is a nonempty n-ary relation on $[k]$, and the claim is true for the relations $S_\ell \subseteq [k]^{n-1}$, $\ell \in I$. If $\mathfrak{A}, T \models \theta_S$, then there are subsets T_0 and T_ℓ, $\ell \in I$, such that

- $T = T_0 \cup \bigcup_{\ell \in I} T_\ell$;
- $\mathfrak{A}, T_0 \models \chi_{m-1}(x_n, \alpha_n)$ for $m = |I|$;
- $\mathfrak{A}, T_\ell \models \alpha_n = \underline{\ell} \wedge \theta_{S_\ell}$ for each $\ell \in I$.

Let $\boldsymbol{a} \in R_{T,\boldsymbol{x}}$. By the second clause above, $|R_{T_0[x_n=a_n],\alpha_n}| < |I|$, whence there exists $\ell \in I$ such that $\ell \notin R_{T_0[x_n=a_n],\alpha_n}$. Since $\mathfrak{A}, T_\ell \models \theta_{S_\ell}$, by induction hypothesis there exists a tuple $\boldsymbol{\ell}^* \in S_\ell \setminus R_{T_\ell[\boldsymbol{x}^*=\boldsymbol{a}^*],\boldsymbol{\alpha}^*}$, where $\boldsymbol{a}^* = (a_1, \ldots, a_{n-1})$, $\boldsymbol{x}^* = (x_1, \ldots, x_{n-1})$ and $\boldsymbol{\alpha}^* = (\alpha_1, \ldots, \alpha_{n-1})$.

Consider now the tuple $\boldsymbol{\ell} = \boldsymbol{\ell}^*\ell$. Since $\boldsymbol{\ell}^* \in S_\ell$ and $S_\ell \times \{\ell\} \subseteq S$, we have $\boldsymbol{\ell} \in S$. On the other hand, $\boldsymbol{\ell} \notin R_{T_0[\boldsymbol{x}=\boldsymbol{a}],\boldsymbol{\alpha}}$, since $\ell \notin R_{T_0[x_n=a_n],\alpha_n}$. Furthermore, $\mathfrak{A}, T_j \models \alpha_n = \underline{j}$, whence $\boldsymbol{\ell} \notin R_{T_j[\boldsymbol{x}=\boldsymbol{a}],\boldsymbol{\alpha}}$ for all $j \in I \setminus \{\ell\}$. Finally, $\boldsymbol{\ell} \notin R_{T_\ell[\boldsymbol{x}=\boldsymbol{a}],\boldsymbol{\alpha}}$, since $\boldsymbol{\ell}^* \in S_\ell \setminus R_{T_\ell[\boldsymbol{x}^*=\boldsymbol{a}^*],\boldsymbol{\alpha}^*}$. Thus, we conclude that $\boldsymbol{\ell} \notin R_{T[\boldsymbol{x}=\boldsymbol{a}],\boldsymbol{\alpha}}$, as $T = T_0 \cup \bigcup_{\ell \in I} T_\ell$.

2. Assume that $R_{T,\boldsymbol{\alpha}} \subseteq S$ and $f : A \to [k]$ is a function satisfying the condition

$$(f(a_1), \ldots, f(a_n)) \in S \setminus R_{T[\boldsymbol{x}=\boldsymbol{a}],\boldsymbol{\alpha}} \text{ for all } \boldsymbol{a} \in R_{T,\boldsymbol{x}}.$$

If $n = 1$, this implies that $R_{T[x_1=a_1],\alpha_1} \subseteq R_{T,\alpha_1} \setminus \{f(a_1)\} \subseteq S \setminus \{f(a_1)\}$, whence $|R_{T[x_1=a_1],\alpha_1}| \leq |S| - 1$ for all $a_1 \in R_{T,x_1}$. Thus, $\mathfrak{A}, T \models \theta_S$ in the case $n = 1$.

Assume then that $S = \bigcup_{\ell \in I} S_\ell \times \{\ell\}$ is a nonempty n-ary relation on $[k]$, and the claim is true for the relations $S_\ell \subseteq [k]^{n-1}$, $\ell \in I$. We define subteams T_0 and T_ℓ, $\ell \in I$, as follows:

- $T_0 := \{s \in T \mid s(\alpha_n) \neq f(s(x_n))\}$, and
- $T_\ell := \{s \in T \mid s(\alpha_n) = f(s(x_n)) = \ell\}$, for $\ell \in I$.

It is clear from these definitions that $T = T_0 \cup \bigcup_{\ell \in I} T_\ell$.

Since $R_{T_0[x_n=a_n],\alpha_n} \subseteq R_{T,\alpha_n} \setminus \{f(a_n)\} \subseteq I \setminus \{f(a_n)\}$, we have $|R_{T_0[x_n=a_n],\alpha_n}| \leq |I| - 1$, and hence $\mathfrak{A}, T_0 \models \chi_{m-1}(x_n, \alpha_n)$ for $m = |I|$. Furthermore, it follows from the definition of T_ℓ that $\mathfrak{A}, T_\ell \models \alpha_n = \underline{\ell}$ for each $\ell \in I$.

It remains to prove that $\mathfrak{A}, T_\ell \models \theta_{S_\ell}$ for each $\ell \in I$. Assume for this purpose that $\boldsymbol{a}^* \in R_{T_\ell,\boldsymbol{x}^*}$, where $\boldsymbol{x}^*$ is the initial segment $(x_1, \ldots, x_{n-1})$ of $\boldsymbol{x}$. Then there is an extension $\boldsymbol{a}$ of $\boldsymbol{a}^*$ by an n-th component a_n such that $\boldsymbol{a} \in R_{T_\ell,\boldsymbol{x}}$. By the assumption on f, we have $(f(a_1), \ldots, f(a_n)) \in S \setminus R_{T[\boldsymbol{x}=\boldsymbol{a}],\boldsymbol{\alpha}} \subseteq S \setminus R_{T_\ell[\boldsymbol{x}=\boldsymbol{a}],\boldsymbol{\alpha}}$. On the other hand, by the definition of T_ℓ, we have $f(a_n) = \ell \in R_{T_\ell[x_n=a_n],\alpha_n}$, whence it must be the case that $(f(a_1), \ldots, f(a_{n-1})) \in S_\ell \setminus R_{T_\ell[\boldsymbol{x}^*=\boldsymbol{a}^*],\boldsymbol{\alpha}^*}$. Since this holds for every $\boldsymbol{a}^* \in R_{T_\ell,\boldsymbol{x}^*}$, it follows from the induction hypothesis that $\mathfrak{A}, T_\ell \models \theta_{S_\ell}$. ◀

We can now define a formula $\eta_{\mathfrak{B}}$ of QF-MUD$[k]$ with free variables $x_1, \ldots, x_n, \alpha_1, \ldots, \alpha_n$ that expresses CSP($\mathfrak{B}$) on teams that satisfy the two conditions mentioned in the discussion after Lemma 7. The formula is defined as follows:

$$\eta_{\mathfrak{B}} := \theta_{R^{\mathfrak{B}}} \vee \mathrm{udep}[k](x_1, \ldots, x_n; \alpha_1, \ldots, \alpha_n).$$

▶ **Lemma 9.** *Let $\mathfrak{A}$ be a τ_n-structure and let T be a team on A such that $R^{\mathfrak{A}} \times R^{\mathfrak{B}} \subseteq R_{T,\boldsymbol{x}\boldsymbol{\alpha}}$ and $R_{T,\boldsymbol{\alpha}} \subseteq R^{\mathfrak{B}}$. Then the following statements are equivalent:*

1. $\mathfrak{A} \in \mathrm{CSP}(\mathfrak{B})$.
2. $\mathfrak{A}, T \models \eta_{\mathfrak{B}}$.

Proof. Assume first that h is a homomorphism from $\mathfrak{A}$ to $\mathfrak{B}$. We define two subteams T'' and T' of T as follows:

$$T'' := \{s \in T \mid s(\alpha_i) = h(s(x_i)) \text{ for all } i \in [n]\} \quad \text{and} \quad T' = T \setminus T''.$$

Clearly, $(h(a_1), \ldots, h(a_n)) \in R^{\mathfrak{B}} \setminus R_{T'[\boldsymbol{x}=\boldsymbol{a}],\boldsymbol{\alpha}}$, for all $\boldsymbol{a} \in R_{T',\boldsymbol{x}}$, whence by Lemma 8.2, $\mathfrak{A}, T' \models \theta_{R^{\mathfrak{B}}}$. Furthermore, $\mathfrak{A}, T'' \models \mathrm{udep}[k](\boldsymbol{x}; \boldsymbol{\alpha})$ by the definition of T''. Thus, $\mathfrak{A}, T \models \eta_{\mathfrak{B}}$.

For the other direction, assume that $\mathfrak{A}, T \models \eta_{\mathfrak{B}}$. Then there are subteams T' and T'' of T such that $T = T' \cup T''$, $\mathfrak{A}, T' \models \theta_{\mathfrak{B}}$, and $A, T'' \models \mathrm{udep}[k](\boldsymbol{x}; \boldsymbol{\alpha})$. By Lemma 7, there is a homomorphism $h : (A, R_{T'',\boldsymbol{x}}) \to ([k], R_{T'',\boldsymbol{\alpha}})$. Since $T'' \subseteq T$, we have $R_{T'',\boldsymbol{\alpha}} \subseteq R_{T,\boldsymbol{\alpha}} \subseteq R^{\mathfrak{B}}$. Furthermore, by Lemma 8.1, $R^{\mathfrak{B}} \setminus R_{T'[\boldsymbol{x}=\boldsymbol{a}],\boldsymbol{\alpha}} \neq \emptyset$, for every $\boldsymbol{a} \in R_{T,\boldsymbol{x}}$. On the other hand, by the assumption $R^{\mathfrak{A}} \times R^{\mathfrak{B}} \subseteq R_{T,\boldsymbol{x\alpha}}$, we have $R^{\mathfrak{B}} \setminus R_{T[\boldsymbol{x}=\boldsymbol{a}],\boldsymbol{\alpha}} = \emptyset$, for each $\boldsymbol{a} \in R_{T,\boldsymbol{x}}$. Thus, $R_{T''[\boldsymbol{x}=\boldsymbol{a}],\boldsymbol{\alpha}} \neq \emptyset$, for every $\boldsymbol{a} \in R_{T,\boldsymbol{x}}$, whence $R^{\mathfrak{A}} \subseteq R_{T,\boldsymbol{x}} = R_{T'',\boldsymbol{x}}$. As $R_{T'',\boldsymbol{\alpha}} \subseteq R^{\mathfrak{B}}$ and $R^{\mathfrak{A}} \subseteq R_{T'',\boldsymbol{x}}$, we conclude that h is a homomorphism $\mathfrak{A} \to \mathfrak{B}$. ◀

We observe next that for each τ_n-structure $\mathfrak{B} = ([k], R^{\mathfrak{B}})$, the properties $R_{T,\boldsymbol{\alpha}} \subseteq R^{\mathfrak{B}}$ and $R_{T,\boldsymbol{\alpha}} \cap R^{\mathfrak{B}} = \emptyset$ of teams are definable in QF-MUD$[k]$. Indeed, if

$$\psi_{\mathfrak{B}} := \bigvee_{\boldsymbol{\ell} \in R^{\mathfrak{B}}} \Big(\bigwedge_{i \in [n]} \alpha_i = \underline{\ell}_i \Big) \qquad \text{and} \qquad \nu_{\mathfrak{B}} := \bigvee_{\boldsymbol{\ell} \notin R^{\mathfrak{B}}} \Big(\bigwedge_{i \in [n]} \alpha_i = \underline{\ell}_i \Big),$$

then clearly for any structure $\mathfrak{A}$ and team T on A, $\mathfrak{A}, T \models \psi_{\mathfrak{B}}$ if and only if $R_{T,\boldsymbol{\alpha}} \subseteq R^{\mathfrak{B}}$, and $\mathfrak{A}, T \models \nu_{\mathfrak{B}}$ if and only if $R_{T,\boldsymbol{\alpha}} \cap R^{\mathfrak{B}} = \emptyset$.

▶ **Theorem 10** ($\forall$-MUD$[\omega]$ captures CSP). *Let $\mathfrak{B} = ([k], R^{\mathfrak{B}})$ be a τ_n-structure. There is a sentence $\varphi_{\mathfrak{B}} \in \forall$-MUD$[k]$ such that for every τ_n-structure $\mathfrak{A}$, the following statements are equivalent:*
1. $\mathfrak{A} \in \mathrm{CSP}(\mathfrak{B})$.
2. $\mathfrak{A} \models \varphi_{\mathfrak{B}}$.

Proof. Let $\xi_{\mathfrak{B}}$ be the QF-MUD$[k]$-formula

$$(\eta_{\mathfrak{B}} \wedge \psi_{\mathfrak{B}}) \vee \neg R(x_1, \ldots, x_n) \vee \nu_{\mathfrak{B}}.$$

We define $\varphi_{\mathfrak{B}}$ to be the universal closure $\forall x_1 \ldots \forall x_n \forall \alpha_1 \ldots \forall \alpha_n \xi_{\mathfrak{B}}$ of the formula $\xi_{\mathfrak{B}}$. By Lemma 6, it suffices to prove that $\mathfrak{A} \in \mathrm{CSP}(\mathfrak{B})$ holds if and only if $\mathfrak{A}, F \models \xi_{\mathfrak{B}}$, where F is the full team on $\mathfrak{A}$ with domain $\{x_1, \ldots, x_n, \alpha_1, \ldots, \alpha_n\}$.

Assume first that $\mathfrak{A} \in \mathrm{CSP}(\mathfrak{B})$. Let T' be the set of all assignments $s \in F$ such that $s(\boldsymbol{x}) \notin R^{\mathfrak{A}}$, and let T'' be the set of all assignments $s \in F$ such that $s(\boldsymbol{\alpha}) \notin R^{\mathfrak{B}}$. Then $\mathfrak{A}, T' \models \neg R(x_1, \ldots, x_n)$ and $\mathfrak{A}, T'' \models \nu_{\mathfrak{B}}$. Furthermore, $R^{\mathfrak{A}} \times R^{\mathfrak{B}} \subseteq R_{T,\boldsymbol{x\alpha}}$ and $R_{T,\boldsymbol{\alpha}} \subseteq R^{\mathfrak{B}}$, for $T = F \setminus (T' \cup T'')$. Thus, T satisfies the conditions of Lemma 9, whence $\mathfrak{A}, T \models \eta_{\mathfrak{B}}$. Since $R_{T,\boldsymbol{\alpha}} \subseteq R^{\mathfrak{B}}$, we also have $\mathfrak{A}, T \models \psi_{\mathfrak{B}}$. Since $F = T \cup T' \cup T''$, we conclude that $\mathfrak{A}, F \models \xi_{\mathfrak{B}}$.

For the other direction, assume that $\mathfrak{A}, F \models \xi_{\mathfrak{B}}$. Then there are subteams T, T' and T'' of F such that
- $F = T \cup T' \cup T''$;
- $\mathfrak{A}, T \models \eta_{\mathfrak{B}} \wedge \psi_{\mathfrak{B}}$;
- $\mathfrak{A}, T' \models \neg R(x_1, \ldots, x_n)$;
- $\mathfrak{A}, T'' \models \nu_{\mathfrak{B}}$.

If $s \in F$ is an assignment such that $s(\boldsymbol{x}) \in R^{\mathfrak{A}}$ and $s(\boldsymbol{\alpha}) \in R^{\mathfrak{B}}$, then it is not possible that $s \in T'$ or $s \in T''$, whence necessarily $s \in T$. Thus, we see that $R^{\mathfrak{A}} \times R^{\mathfrak{B}} \subseteq R_{T,\boldsymbol{x\alpha}}$. Furthermore, since $\mathfrak{A}, T \models \psi_{\mathfrak{B}}$, we have $R_{T,\boldsymbol{\alpha}} \subseteq R^{\mathfrak{B}}$. Hence, T satisfies the conditions of Lemma 9, and consequently $\mathfrak{A} \in \mathrm{CSP}(\mathfrak{B})$. ◀

We note that the size of the sentence $\varphi_{\mathfrak{B}}$ defining $\mathrm{CSP}(\mathfrak{B})$ is polynomial in the size of $\mathfrak{B}$. In fact, it is not hard to show that the size of $\varphi_{\mathfrak{B}}$ is $O(n \cdot k^{n+1})$, where n is the arity of $R^{\mathfrak{B}}$.

We illustrate the preceding theorem by considering again the structure $\mathfrak{B} = (\{0,1\}, R^{\mathfrak{B}})$, where $R^{\mathfrak{B}} = \{(1,0,0), (0,1,0), (0,0,1)\}$. In this case, we have that

$$\varphi_{\mathfrak{B}} := \forall x_1 \forall x_2 \forall x_3 \forall \alpha_1 \forall \alpha_2 \forall \alpha_3 \big((\eta_{\mathfrak{B}} \wedge \psi_{\mathfrak{B}}) \vee \neg R(x_1, x_2, x_3) \vee \nu_{\mathfrak{B}} \big),$$

where

- $\eta_{\mathfrak{B}} := \mathrm{udep}[2](x_3;\alpha_3) \vee (\alpha_3 = \underline{0} \wedge \mathrm{udep}[2](x_2;a_2)) \vee \mathrm{udep}[2](x_1,x_2,x_3;\alpha_1,\alpha_2,\alpha_3)$
- $\psi_{\mathfrak{B}} := (\alpha_1 = \underline{1} \wedge \alpha_2 = \underline{0} \wedge \alpha_3 = \underline{0}) \vee (\alpha_1 = \underline{0} \wedge \alpha_2 = \underline{1} \wedge \alpha_3 = \underline{0}) \vee (\alpha_1 = \underline{0} \wedge \alpha_2 = \underline{0} \wedge \alpha_3 = \underline{1})$
- $\nu_{\mathfrak{B}} := (\alpha_1 = \underline{1} \wedge \alpha_2 = \underline{1} \wedge \alpha_3 = \underline{1}) \vee (\alpha_1 = \underline{0} \wedge \alpha_2 = \underline{0} \wedge \alpha_3 = \underline{0}) \vee$
  $(\alpha_1 = \underline{1} \wedge \alpha_2 = \underline{1} \wedge \alpha_3 = \underline{0}) \vee (\alpha_1 = \underline{1} \wedge \alpha_2 = \underline{0} \wedge \alpha_3 = \underline{1}) \vee (\alpha_1 = \underline{0} \wedge \alpha_2 = \underline{1} \wedge \alpha_3 = \underline{1})$.

4 From Dependence Logic to MMSNP

In this section, we prove that MMSNP is at least as expressive as $\forall$-MUD$[\omega]$. In the proof of this result, we will make use of the extension of MMSNP with k-valued variables. This logic, denoted MMSNP$[k]$, is obtained from MMSNP by adding equalities of the form $\alpha = \underline{i}$, $i \in [k]$, and universal quantification of k-valued variables. As in the case of $\forall$-MUD$[k]$, there is no need to add equalities of the form $\alpha = \beta$, since they are definable.

We will use the logic MMSNP$[k]$ only as an intermediate step in translating formulas of $\forall$-MUD$[k]$ to equivalent sentences of MMSNP. The following lemma, which has a straightforward proof, tells that the k-valued variables in any MMSNP$[k]$-sentence can be eliminated to obtain an equivalent MMSNP-sentence.

▶ **Lemma 11.** *For every sentence $\varphi \in$ MMSNP$[k]$, there is a sentence $\psi \in$ MMSNP such that, for every structure $\mathfrak{A}$, we have that $\mathfrak{A} \models \varphi$ if and only if $\mathfrak{A} \models \psi$.*

We make next the simple observation that uniform dependence atoms are expressible in existential monadic second-order logic. Indeed, the function $f : A \to [k]$ in the semantics of udep$[k]$ can be replaced with a k-tuple of unary relations: $\mathfrak{A}, T \models \mathrm{udep}[k](\boldsymbol{x};\boldsymbol{\alpha})$ if and only if there are $P_1^{\mathfrak{A}}, \ldots, P_k^{\mathfrak{A}} \subseteq A$ such that

1. $A = \bigcup_{j\in[k]} P_j^{\mathfrak{A}}$ and $P_i^{\mathfrak{A}} \cap P_j^{\mathfrak{A}} = \emptyset$ for $i \neq j$, and
2. $s(x_i) \in P_{s(\alpha_i)}^{\mathfrak{A}}$ for all $i \in [n]$ and $s \in T$.

Both of these conditions can be expressed by first-order formulas:

1. $A = \bigcup_{j\in[k]} P_j^{\mathfrak{A}}$ and $P_i^{\mathfrak{A}} \cap P_j^{\mathfrak{A}} = \emptyset$ for $i \neq j$ if and only if $(\mathfrak{A}, (P_j^{\mathfrak{A}})_{j\in[k]}) \models \forall y\, \theta$, where
$$\theta := \bigvee_{j\in[k]} P_j(y) \wedge \bigwedge_{1\le i<j\le k} \neg(P_i(y) \wedge P_j(y)).$$
2. $s(x_i) \in P_{s(\alpha_i)}^{\mathfrak{A}}$ for all $i \in [n]$ and $s \in T$ if and only if $(\mathfrak{A}, (P_j^{\mathfrak{A}})_{j\in[k]}), s \models \rho_{k,n}$ for all $s \in T$, where
$$\rho_{k,n} := \bigwedge_{i\in[n], j\in[k]} (P_j(x_i) \leftrightarrow \alpha_i = \underline{j}).$$

Thus, $\mathfrak{A}, T \models \mathrm{udep}[k](\boldsymbol{x};\boldsymbol{\alpha})$ if and only if $\mathfrak{A}, s \models \exists P_1 \ldots \exists P_k(\forall y\, \theta \wedge \rho_{k,n})$ for all $s \in T$. In the next lemma, we formulate the definition of udep$[k](\boldsymbol{x};\boldsymbol{\alpha})$ by a *sentence* of MMSNP$[k]$. To do this, we need to replace the team T with the corresponding relation $R_{T,\boldsymbol{x\alpha}}$.

▶ **Lemma 12.** *Let $\mathfrak{A}$ be a structure and T a team on $\mathfrak{A}$ such that the domain of T is $V = \{x_1, \ldots, x_n, \alpha_1, \ldots, \alpha_n\}$. Then the following statements are equivalent:*

1. $\mathfrak{A}, T \models \mathrm{udep}[k](\boldsymbol{x};\boldsymbol{\alpha})$.
2. $(\mathfrak{A}, R_{T,\boldsymbol{x\alpha}}) \models \exists P_1 \ldots \exists P_k \forall \boldsymbol{x} \forall y \forall \boldsymbol{\alpha}(R(\boldsymbol{x\alpha}) \to \theta \wedge \rho_{k,n})$.

Proof. Using the preceding observations, we get the following chain of equivalences:

$$\begin{aligned}\mathfrak{A}, T \models \mathrm{udep}[k](\boldsymbol{x};\boldsymbol{\alpha}) \iff\ & \text{there are } P_1^{\mathfrak{A}}, \ldots, P_k^{\mathfrak{A}} \subseteq A \text{ such that, for all } s \in T,\\ & (\mathfrak{A}, (P_j^{\mathfrak{A}})_{j\in[k]}), s \models \forall y\, \theta \wedge \rho_{k,n}\\ \iff\ & \text{there are } P_1^{\mathfrak{A}}, \ldots, P_k^{\mathfrak{A}} \subseteq A \text{ such that, for all } s' \in F_{V\cup\{y\}},\\ & (\mathfrak{A}, (P_j^{\mathfrak{A}})_{j\in[k]}, R_{T,\boldsymbol{x\alpha}}), s' \models R(\boldsymbol{x\alpha}) \to \theta \wedge \rho_{k,n}\\ \iff\ & (\mathfrak{A}, R_{T,\boldsymbol{x\alpha}}) \models \exists P_1 \ldots \exists P_k \forall \boldsymbol{x} \forall y \forall \boldsymbol{\alpha}(R(\boldsymbol{x\alpha}) \to \theta \wedge \rho_{k,n}).\end{aligned}$$ ◀

Our next aim is to show that the translation of uniform dependence atoms to MMSNP[k] given in Lemma 12 can be extended to arbitrary formulas of QF-MUD[k]. The number of existentially quantified unary relations in the translation of a formula ψ depends on the number of occurrences of dependence atoms in ψ. We denote this number by $\sharp_{\mathrm{dep}}(\psi)$.

▶ **Lemma 13.** *Let ψ be a formula of* QF-MUD[k] *with $\mathrm{Fr}(\psi) = \{x_1, \ldots, x_n, \alpha_1, \ldots, \alpha_m\}$ and $\sharp_{\mathrm{dep}}(\psi) = \ell$. There is a quantifier-free formula ψ^+ with $\mathrm{Fr}(\psi^+) = \mathrm{Fr}(\psi) \cup \{y_1, \ldots, y_\ell\}$ and with free unary second-order variables $P_1, \ldots, P_{k\ell}$ such that for every structure $\mathfrak{A}$ and every team T on $\mathfrak{A}$ with $\mathrm{Fr}(\psi) \subseteq \mathrm{dom}(T)$, we have that the following statements are equivalent:*
1. $\mathfrak{A}, T \models \psi$.
2. $(\mathfrak{A}, R_{T,\boldsymbol{x\alpha}}) \models \exists P_1 \ldots \exists P_{k\ell} \forall \boldsymbol{x} \forall \boldsymbol{y} \forall \boldsymbol{\alpha}(R(\boldsymbol{x\alpha}) \to \psi^+)$.

Proof. The claimed equivalence is proved by induction on the formula ψ.
- If ψ is a negated atomic formula $\neg S(\boldsymbol{x})$, then $m = \ell = 0$, and we have

$$\begin{aligned}\mathfrak{A}, T \models \psi \iff\ & \mathfrak{A}, s \models \psi \text{ for every assignment } s \in T\\ \iff\ & (\mathfrak{A}, R_{T,\boldsymbol{x}}), s \models R(\boldsymbol{x}) \to \psi \text{ for every assignment } s \in F_{\mathrm{dom}(T)}\\ \iff\ & (\mathfrak{A}, R_{T,\boldsymbol{x}}), s \models \forall \boldsymbol{x}(R(\boldsymbol{x}) \to \psi).\end{aligned}$$

  Thus, we simply let $\psi^+ := \psi$ in this case.
- If ψ is an equality $\alpha_1 = \underline{j}$, we define $\psi^+ := \psi$. Then $n = \ell = 0$, $m = 1$ and in the same way as in the previous case, we see that $\mathfrak{A}, T \models \psi \iff (\mathfrak{A}, R_{T,\alpha_1}), s \models \forall \alpha_1(R(\alpha_1) \to \psi^+)$.
- If ψ is a uniform dependence atom $\mathrm{udep}[k](\boldsymbol{x};\boldsymbol{\alpha})$, we define $\psi^+ = \theta \wedge \rho_{k,n}$; the claimed equivalence follows from Lemma 12.
- Let $\psi = \pi \vee \sigma$. Then $\ell = \sharp_{\mathrm{dep}}(\psi) = \ell' + \ell''$, where $\ell' = \sharp_{\mathrm{dep}}(\pi)$ and $\ell'' = \sharp_{\mathrm{dep}}(\sigma)$. Let σ_*^+ be the formula obtained from σ^+ by replacing each relation variable P_j with $P_{k\ell'+j}$, $j \in [k\ell'']$, and replacing each first-order variable y_j with $y_{\ell'+j}$, $j \in [\ell'']$. Put $\psi^+ := \pi^+ \vee \sigma_*^+$.
  Assume first that $\mathfrak{A}, T \models \psi$. Then there are subteams T' and T'' of T such that $T = T' \cup T''$, $\mathfrak{A}, T' \models \pi$ and $\mathfrak{A}, T'' \models \sigma$. Let $\boldsymbol{u}$ and $\boldsymbol{\beta}$ be tuples that list those of the variables in $\{x_1, \ldots, x_n, \alpha_1, \ldots, \alpha_m\}$ that occur in π, and similarly let the tuples $\boldsymbol{v}$ and $\boldsymbol{\gamma}$ list the variables occurring in σ. By induction hypothesis, we have

$$(\mathfrak{A}, R_{T',\boldsymbol{u\beta}}) \models \exists P_1 \ldots \exists P_{k\ell'} \forall \boldsymbol{u} \forall \boldsymbol{y}' \forall \boldsymbol{\beta}(R(\boldsymbol{u\beta}) \to \pi^+),$$

  where $\boldsymbol{y}' = (y_1, \ldots, y_{\ell'})$. Similarly, changing the bound variables and using the induction hypothesis, we get $(\mathfrak{A}, R_{T'',\boldsymbol{v\gamma}}) \models \exists P_{k\ell'+1} \ldots \exists P_{k\ell} \forall \boldsymbol{v} \forall \boldsymbol{y}'' \forall \boldsymbol{\gamma}(R(\boldsymbol{v\gamma}) \to \sigma_*^+)$, where $\boldsymbol{y}'' = (y_{\ell'+1}, \ldots, y_\ell)$. Thus, there are relations $P_1^{\mathfrak{A}}, \ldots, P_{k\ell}^{\mathfrak{A}} \subseteq A$ such that
  - $(\mathfrak{A}, (P_j^{\mathfrak{A}})_{j\in[k\ell]}), s \models \pi^+$ for every assignment $s \in T'[A^\ell/\boldsymbol{y}]$, and
  - $(\mathfrak{A}, (P_j^{\mathfrak{A}})_{j\in[k\ell]}), s \models \sigma_*^+$ for every assignment $s \in T''[A^\ell/\boldsymbol{y}]$.

  Since $T = T' \cup T''$, it follows that $(\mathfrak{A}, (P_j^{\mathfrak{A}})_{j\in[k\ell]}), s \models \pi^+ \vee \sigma_*^+$ for all assignments $s \in T[A^\ell/\boldsymbol{y}]$. Thus, we see that $(\mathfrak{A}, R_{T,\boldsymbol{x\alpha}}) \models \exists P_1 \ldots \exists P_{k\ell} \forall \boldsymbol{x} \forall \boldsymbol{y} \forall \boldsymbol{\alpha}(R(\boldsymbol{x\alpha}) \to (\pi^+ \vee \sigma_*^+))$.

For the other direction, assume that there are relations $P_1^{\mathfrak{A}}, \ldots, P_{k\ell}^{\mathfrak{A}} \subseteq A$ such that

$$(\mathfrak{A}, (P_j^{\mathfrak{A}})_{j\in[k\ell]}, R_{T,\boldsymbol{x\alpha}}) \models \forall \boldsymbol{x} \forall \boldsymbol{y} \forall \boldsymbol{\alpha}(R(\boldsymbol{x\alpha}) \to (\pi^+ \vee \sigma_*^+)).$$

Then $(\mathfrak{A}, (P_j^{\mathfrak{A}})_{j\in[k\ell]}), s \models \pi^+ \vee \sigma_*^+$ for every assignment $s \in T[A^\ell/\boldsymbol{y}]$. Let T' and T'' be the subteams

- $T' := \{s \in T \mid (\mathfrak{A}, (P_j^{\mathfrak{A}})_{j\in[k\ell]}), s \models \forall \boldsymbol{y}' \pi^+\}$ and
- $T'' := \{s \in T \mid (\mathfrak{A}, (P_j^{\mathfrak{A}})_{j\in[k\ell]}), s \models \forall \boldsymbol{y}'' \sigma_*^+\}$,

where $\boldsymbol{y}'$ and $\boldsymbol{y}''$ are as above. By the definition of T', we have

$$(\mathfrak{A}, (P_j^{\mathfrak{A}})_{j\in[k\ell']}, R_{T',\boldsymbol{u\beta}}) \models \forall \boldsymbol{u} \forall \boldsymbol{y}' \forall \boldsymbol{\beta}(R(\boldsymbol{u\beta}) \to \pi^+).$$

Universal quantification can be restricted here to those first-order and k-valued variables that occur in π^+. Similarly, it suffices to consider relation variables P_j occurring in π^+. From the induction hypothesis, it follows that $\mathfrak{A}, T' \models \pi$. In the same way, we see that

$$(\mathfrak{A}, (P_{k\ell'+j}^{\mathfrak{A}})_{j\in[k\ell'']}, R_{T'',\boldsymbol{v\gamma}}) \models \forall \boldsymbol{v} \forall \boldsymbol{y}'' \forall \boldsymbol{\gamma}(R(\boldsymbol{v\gamma}) \to \sigma_*^+).$$

Replacing each relation variable $P_{k\ell'+j}$ by P_j and each first-order variable $y_{\ell'+j}$ by y_j and using the induction hypothesis, we see that $\mathfrak{A}, T'' \models \sigma$. Finally, it is clear that $T = T' \cup T''$, whence we get $\mathfrak{A}, T \models \psi$.

- If $\psi = \pi \wedge \sigma$, we define $\psi^+ := \pi^+ \wedge \sigma_*^+$. This case is handled as the preceding one. ◀

We now have all the technical machinery needed to prove the main result of this section.

▶ **Theorem 14** ($\forall$-MUD[ω] is contained in MMSNP)**.** *For every sentence $\varphi \in \forall$-MUD[ω] there is a sentence $\varphi^* \in$ MMSNP such that the following statements are equivalent:*

1. $\mathfrak{A} \models \varphi$.
2. $\mathfrak{A} \models \varphi^*$.

Proof. By Lemma 11, it suffices to prove the claimed equivalence for a sentence φ^* of MMSNP[k]. Let ψ be a formula of QF-MUD[k] such that φ is (equivalent to) its universal closure $\forall x_1 \ldots \forall x_n \forall \alpha_1 \ldots \forall \alpha_m \psi$, and let F be the full team on $\mathfrak{A}$ with domain $\{x_1, \ldots, x_n, \alpha_1, \ldots, \alpha_m\}$. By applying Lemma 13, we obtain a quantifier-free formula ψ^+ such that

$$\mathfrak{A}, F \models \psi \iff (\mathfrak{A}, R_{F,\boldsymbol{x\alpha}}) \models \exists P_1 \ldots \exists P_{k\ell} \forall \boldsymbol{x} \forall \boldsymbol{y} \forall \boldsymbol{\alpha}(R(\boldsymbol{x\alpha}) \to \psi^+).$$

Since F is the full team, $(\mathfrak{A}, R_{F,\boldsymbol{x\alpha}}), s \models R(\boldsymbol{x\alpha})$ holds for all assignments s. Hence, the implication $R(\boldsymbol{x\alpha}) \to \psi^+$ can be replaced by its right-hand side ψ^+, and the relation $R_{F,\boldsymbol{x\alpha}}$ can be omitted in the equivalence above. Thus, using Lemma 6, we obtain the equivalence

$$\mathfrak{A} \models \varphi \iff \mathfrak{A} \models \exists P_1 \ldots \exists P_{k\ell} \forall \boldsymbol{x} \forall \boldsymbol{y} \forall \boldsymbol{\alpha} \psi^+.$$

From the definition of ψ^+, we see that the sentence $\exists P_1 \ldots \exists P_{k\ell} \forall \boldsymbol{x} \forall \boldsymbol{y} \forall \boldsymbol{\alpha}\, \psi^+$ is in MMSNP[k]. Thus, the claimed equivalence holds for $\varphi^* := \exists P_1 \ldots \exists P_{k\ell} \forall \boldsymbol{x} \forall \boldsymbol{y} \forall \boldsymbol{\alpha}\, \psi^+$. ◀

We note that the size of the MMSNP[k]-sentence φ^* is polynomial in the size of the $\forall$-MUD[ω]-sentence φ. In fact, it is not hard to show that the size of φ^* is $O(nk^3|\varphi|)$.

Given a sentence $\varphi \in \forall$-MUD[ω], we denote its model-checking problem "given a structure $\mathfrak{A}$, does $\mathfrak{A} \models \varphi$ hold?" by $\mathcal{MC}(\varphi)$. Using the polynomial-time equivalence of MMSNP and CSP, we obtain the following corollary to Theorems 10 and 14.

▶ **Corollary 15.** *If φ is a sentence of $\forall$-MUD[ω], then $\mathcal{MC}(\varphi)$ is polynomial-time equivalent to CSP($\mathfrak{B}$), for some structure $\mathfrak{B}$; vice versa, if $\mathfrak{B}$ is a structure, then CSP($\mathfrak{B}$) is polynomial-time equivalent to $\mathcal{MC}(\varphi)$, for some sentence φ of $\forall$-MUD[ω].*

5 Concluding Remarks

In this paper, we established a tight connection between dependence logic and constraint satisfaction. Since dependence logic has the same expressive power as existential second-order logic, it is expected that constraint satisfaction problems can be expressed in dependence logic. We believe, however, that the connection established in this paper is a priori unexpected, since we showed that a simple fragment of *universal* dependence logic captures, in a precise sense, the family of constraint satisfaction problems $\mathrm{CSP}(\mathfrak{B})$, where $\mathfrak{B}$ is a relational structure. Our results contribute to the descriptive complexity of constraint satisfaction and also shed new light on quantifier-free and universal dependence logic.

The connection between universal dependence logic and constraint satisfaction is established by using MMSNP as a bridge and also the result by Feder and Vardi [5] that MMSNP captures constraint satisfaction via polynomial-time reductions. Specifically, we showed that every constraint satisfaction problem $\mathrm{CSP}(\mathfrak{B})$, in which $\mathfrak{B}$ has only one relation, is definable by a $\forall$-MUD$[\omega]$-sentence, and every $\forall$-MUD$[\omega]$-sentence is equivalent to some MMSNP-sentence. A natural question that arises from these results is whether every MMSNP-sentence is equivalent to some $\forall$-MUD$[\omega]$-sentence or, in other words, whether MMSNP and $\forall$-MUD$[\omega]$ have the same expressive power. A related question is to identify other natural fragments of dependence logic that capture important fragments of existential second-order logic, such as *strict* existential second-order logic (i.e., the fragment of existential second-order logic in which all first-order quantifiers are universal).

Acknowledgements. A part of the research reported here was carried out while Lauri Hella was visiting the University of California Santa Cruz.

References

1 Nadia Creignou, Phokion G. Kolaitis, and Heribert Vollmer, editors. *Complexity of Constraints – An Overview of Current Research Themes [Result of a Dagstuhl Seminar]*, volume 5250 of *Lecture Notes in Computer Science*. Springer, 2008.

2 Arnaud Durand, Juha Kontinen, Nicolas de Rugy-Altherre, and Jouko Väänänen. Tractability frontier of data complexity in team semantics. In *Proceedings Sixth International Symposium on Games, Automata, Logics and Formal Verification, GandALF 2015, Genoa, Italy, 21-22nd September 2015.*, pages 73–85, 2015.

3 Paul Erdös. Graph theory and probability. *Canadian J. of Mathematics*, 11:34–38, 1959.

4 Ronald Fagin. Generalized first-order spectra and polynomial-time recognizable sets. In Richard Karp, editor, *Complexity of Computation*, number 7 in SIAM-AMS Proceedings, pages 43–73. SIAM-AMS, 1974.

5 Tomás Feder and Moshe Y. Vardi. The computational structure of monotone monadic SNP and constraint satisfaction: A study through datalog and group theory. *SIAM J. Comput.*, 28(1):57–104, 1998.

6 P. Galliani. Inclusion and exclusion dependencies in team semantics – on some logics of imperfect information. *Ann. Pure Appl. Logic*, 163(1):68–84, 2012.

7 Pietro Galliani and Lauri Hella. Inclusion logic and fixed point logic. In *Computer Science Logic 2013 (CSL 2013), CSL 2013, September 2-5, 2013, Torino, Italy*, number 23 in LIPIcs, pages 281–295. Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, 2013. doi:10.4230/LIPIcs.CSL.2013.281.

8 Erich Grädel and Jouko A. Väänänen. Dependence and independence. *Studia Logica*, 101(2):399–410, 2013. doi:10.1007/s11225-013-9479-2.

9 Johan Håstad, Andrei A. Krokhin, and Dániel Marx. The constraint satisfaction problem: Complexity and approximability (Dagstuhl Seminar 12451). *Dagstuhl Reports*, 2(11):1–19, 2012.

10 Leon Henkin. Some remarks on infinitely long formulas. In *Infinitistic Methods*. Pergamon Press, 1961.

11 Jaakko Hintikka and Gabriel Sandu. Informational independence as a semantical phenomenon. In J. E. Fenstad et al., editor, *Logic, Methodology and the Philosophy of Science VIII*, pages 571–89. North-Holland, 1989.

12 Jarmo Kontinen. Coherence and computational complexity of quantifier-free dependence logic formulas. *Studia Logica*, 101(2):267–291, 2013. `doi:10.1007/s11225-013-9481-8`.

13 Juha Kontinen and Jouko A. Väänänen. On definability in dependence logic. *Journal of Logic, Language and Information*, 18(3):317–332, 2009. `doi:10.1007/s10849-009-9082-0`.

14 Juha Kontinen and Jouko A. Väänänen. Axiomatizing first-order consequences in dependence logic. *Ann. Pure Appl. Logic*, 164(11):1101–1117, 2013. `doi:10.1016/j.apal.2013.05.006`.

15 Gábor Kun and Jaroslav Nesetril. Forbidden lifts (NP and CSP for combinatorialists). *Eur. J. Comb.*, 29(4):930–945, 2008.

16 Richard E. Ladner. On the structure of polynomial time reducibility. *J. ACM*, 22(1):155–171, 1975.

17 Jouko A. Väänänen. *Dependence Logic – A New Approach to Independence Friendly Logic*, volume 70 of *London Mathematical Society student texts*. Cambridge University Press, 2007. URL: `http://www.cambridge.org/de/knowledge/isbn/item1164246/`.