Samson Abramsky
Juha Kontinen
Jouko Väänänen
Heribert Vollmer
Editors

Dependence Logic

Theory and Applications

Birkhäuser

Birkhäuser

Samson Abramsky • Juha Kontinen
Jouko Väänänen • Heribert Vollmer
Editors

Dependence Logic

Theory and Applications

Birkhäuser

Editors
Samson Abramsky
Department of Computer Science
University of Oxford
Oxford, UK

Juha Kontinen
Department of Mathematics and Statistics
University of Helsinki
Helsinki, Finland

Jouko Väänänen
Department of Mathematics & Statistics
University of Helsinki
Helsinki, Finland

Heribert Vollmer
Institut für Theoretische Informatik
Leibniz Universität Hannover
Hannover, Germany

ISBN 978-3-319-31801-1 ISBN 978-3-319-31803-5 (eBook)
DOI 10.1007/978-3-319-31803-5

Library of Congress Control Number: 2016941197

Mathematics Subject Classification (2010): 03B15, 03B70, 03C80

Printed on acid-free paper

This book is published under the trade name Birkhäuser
The registered company is Springer International Publishing AG, CH

Preface

In February 2013, the editors of this volume organized a Dagstuhl Seminar on "Dependence Logic: Theory and Applications" at the Schloss Dagstuhl Leibniz Center for Informatics in Wadern, Germany. This event consisted of both invited and contributed talks by some of the approximately 40 participants. After the conclusion of the seminar, the organizers invited a number of speakers to write surveys that present the state-of-the-art knowledge in their area of expertise. The Dagstuhl Seminar was followed by an Academy Colloquium of the Koninklijke Nederlandse Akademie van Wetenschappen (KNAW) and a further Dagstuhl Seminar "Logics for Dependence and Independence" in June 2015. Also from these latter meetings, a few participants were invited to contribute to this volume.

All contributions were peer-reviewed by experts in the field and revised before they were included in this volume.

We thank the Directorate of Schloss Dagtuhl and the board of the KNAW for their support, the speakers of the seminars for making all three meetings a successful event, the referees and, above all, the contributors of this volume for their informative and well-written articles. We also thank Benjamin Levitt at the publisher's office for his support and guidance.

Oxford, UK — Samson Abramsky
Helsinki, Finland — Juha Kontinen
Helsinki, Finland — Jouko Väänänen
Hannover, Germany — Heribert Vollmer
June 2016

Contents

Contributors

A. Blass Mathematics Department, University of Michigan, Ann Arbor, MI, USA

I. Ciardelli ILLC, University of Amsterdam, Amsterdam, The Netherlands

J. Corander Department of Mathematics and Statistics, University of Helsinki, Helsinki, Finland

A. Durand IMJ-PRG, CNRS UMR 7586, Université Paris Diderot, Paris, France

P. Galliani School of Engineering and Informatics, University of Sussex, Falmer, Brighton, UK

E. Grädel Math. Foundations of Computer Science, RWTH Aachen University, Aachen, Germany

Å. Hirvonen Department of Mathematics and Statistics, University of Helsinki, Helsinki, Finland

W. Hodges Herons Brook, Sticklepath, Okehampton, Devon EX20 2PY, UK

J. Kontinen Department of Mathematics and Statistics, University of Helsinki, Helsinki, Finland

S. Link The University of Auckland, Auckland, New Zealand

H. Nyman Department of Mathematics and Statistics, Åbo Akademi University, Turku, Finland

E. Pacuit Department of Philosophy, University of Maryland, College Park, MD, USA

J. Pensar Department of Mathematics and Statistics, Åbo Akademi University, Turku, Finland

J. Väänänen Department of Mathematics and Statistics, University of Helsinki, Helsinki, Finland
Institute for Logic, Language and Computation, University of Amsterdam, Amsterdam, The Netherlands

H. Vollmer Institut für Theoretische Informatik, Leibniz Universität Hannover, Hannover, Germany

F. Yang Department of Philosophy and Religious Studies, Utrecht University, Utrecht, The Netherlands

Introduction

Samson Abramsky, Juha Kontinen, Jouko Väänänen, and Heribert Vollmer

Dependence logic was introduced by Jouko Väänänen in 2007. It extends first-order logic by new atomic dependence formulas (dependence atoms)

$$=\!(x_1, \dots, x_n)\,, \tag{1}$$

the meaning of which is that the value of x_n is functionally determined by the values of $x_1, \dots, x_{n-1}$. On the semantical side, dependence logic bases its semantics in the concept of a set X of assignments instead of, as is the case for usual first-order logic, a single assignment. Such sets are called *teams*. A team X is said to satisfy the dependence atom above if for any two assignments, if they agree on the variables $x_1, \dots, x_{n-1}$ then they also agree on x_n. By viewing a team X as a database over attributes $x_1, \dots, x_n$, dependence atoms correspond exactly to functional dependencies studied extensively in database theory. Dependence logic

S. Abramsky (✉)
Department of Computer Science, University of Oxford Wolfson Building, Parks Road, Oxford OX1 3QD, United Kingdom
e-mail: samson.abramsky@cs.ox.ac.uk

J. Kontinen
Department of Mathematics and Statistics, University of Helsinki, P.O. Box 68, 00014 Helsinki, Finland
e-mail: juha.kontinen@helsinki.fi

J. Väänänen
Department of Mathematics and Statistics, University of Helsinki, Helsinki, Finland

Institute for Logic, Language and Computation, University of Amsterdam, Amsterdam, The Netherlands
e-mail: jouko.vaananen@helsinki.fi

H. Vollmer
Institut für Theoretische Informatik, Leibniz Universität Hannover, Appelstr. 4, 30167 Hannover, Germany
e-mail: vollmer@thi.uni-hannover.de

S. Abramsky et al. (eds.), *Dependence Logic*, DOI 10.1007/978-3-319-31803-5_1

was introduced as a systematic extension of first-order logic by a means to explicitly talk about dependence among variables. Earlier attempts in this direction were the definition of *partially ordered quantifiers* by Henkin and the introduction of *Independence-friendly Logic* by Hintikka and Sandu.

In the past few years, the area of dependence logic has developed rapidly. One of the breakthroughs in the area was the introduction of *Independence Logic* that replaces the dependence atoms of dependence logic by independence atoms $\mathbf{x} \perp_{\mathbf{z}} \mathbf{y}$. The intuitive meaning of the atom $\mathbf{x} \perp_{\mathbf{z}} \mathbf{y}$ is that, for any fixed values of the variables $\mathbf{z}$, the variables $\mathbf{x}$ are independent of the variables $\mathbf{y}$ in the sense that knowing the value of $\mathbf{x}$ does not tell us anything about the value of $\mathbf{y}$. In databases, independence atoms correspond to embedded multivalued data dependencies. Furthermore, independence atoms and statistical conditional independence have also interesting connections.

More recently Galliani showed that independence atoms can be further analysed by the so-called inclusion $\mathbf{x} \subseteq \mathbf{y}$ and exclusion $\mathbf{x}|\mathbf{y}$ atoms. Both of these atoms have also been studied in database theory. The meaning of the atom $\mathbf{x} \subseteq \mathbf{y}$ is that all values of $\mathbf{x}$ in a team appear also as a value of $\mathbf{y}$, whereas the meaning of $\mathbf{x}|\mathbf{y}$ is that the values taken by $\mathbf{x}$ and $\mathbf{y}$ are distinct. Inclusion atoms have very interesting properties in the team semantics setting, e.g., they give rise to a variant of dependence logic that corresponds to the complexity class PTIME.

The idea of dependencies and independencies among variables has also been introduced into the context of propositional and (propositional) modal logic. This area has developed rapidly in the past few years, leading to many expressivity and complexity results.

This volume contains 11 articles discussing different aspects of logics for dependence and independence. Among these, there are articles addressing purely logical issues or computational aspects of dependence logic, but there are also several articles concerned with applications of dependence logic in various areas.

The chapter by A. Durand, J. Kontinen, and H. Vollmer gives a comprehensive survey of many propositional, modal, and first-order variants of dependence logic. It summarizes the state of the art regarding the expressive power as well as computational questions such as the complexity of the satisfiability or the model checking problems. The contribution by J. Väänänen discusses a theory of dependence developed by the German logician Kurt Grelling in an unpublished article from 1939 in the team semantics context. The chapter by P. Galliani contributes new results concerning expressive power of various variants of dependence logic with different sets of logical connectives and generalized dependence atoms. The contribution by E. Grädel studies the connections between inclusion logic and the least fixed-point logic. The chapter by W. Hodges discusses compositionality in the team semantics context. The topic of the survey of Å. Hirvonen are model-theoretic independence notions.

The notions of dependence and independence are central in many scientific areas, some of which are addressed in the remaining papers of this volume.

Inquisitive semantics is a new area aiming to give a logical account of information exchange as a process of requesting and providing information. The contribution by I. Ciardelli gives a comprehensive introduction to inquisitive logic, and also discusses the intimate connections between inquisitive logic and certain variants of dependence logic. The contribution by S. Link studies dependencies in databases by addressing the relationships between implication problems for fragments of statistical conditional independencies, embedded multivalued dependencies, and propositional logic. On the other hand, the article by H. Nyman, J. Pensar, and J. Corander reviews various Markovian models used to characterize dependencies and causality among variables in multivariate systems. The topic of the contribution by E. Pacuit and F. Yang is applications of dependence logic in social choice theory. The article defines an axiomatization of the famous Arrow's theorem using independence logic. The survey by A. Blass gives an introduction to the theory of secret-sharing pointing out connections to dependence and independence logic.

Expressivity and Complexity of Dependence Logic

Arnaud Durand, Juha Kontinen, and Heribert Vollmer

1 Introduction

In this article we review recent results on expressivity and complexity of first-order, modal, and propositional dependence logic and some of its variants such as independence and inclusion logic. Dependence logic was introduced by Jouko Väänänen in [56]. On the syntactic side, it extends usual first-order logic by the so-called dependence atoms

$$=(x_1, \ldots, x_n),$$

the meaning of which is that the value of x_n is functionally determined by the values of $x_1, \ldots, x_{n-1}$. The semantics of dependence logic is defined using sets of assignments, teams, rather than single assignments as in first-order logic. Since the introduction of dependence logic in 2007, the area of team semantics has evolved into a general framework for logics in which various notions of dependence and independence can be formalized and studied. In this paper we mainly consider

A. Durand (✉)
IMJ-PRG, CNRS UMR 7586, Université Paris Diderot, Case 7012, 75205 Paris cedex 13, France
e-mail: durand@logique.jussieu.fr

J. Kontinen
Department of Mathematics and Statistics, University of Helsinki, P.O. Box 68, 00014 Helsinki, Finland
e-mail: juha.kontinen@helsinki.fi

H. Vollmer
Institut für Theoretische Informatik, Leibniz Universität Hannover, Appelstr. 4, 30167 Hannover, Germany
e-mail: vollmer@thi.uni-hannover.de

S. Abramsky et al. (eds.), *Dependence Logic*, DOI 10.1007/978-3-319-31803-5_2

variants of dependence logic arising by replacing/supplementing dependence atoms with further dependency notions, and we also study propositional and modal variants.

In Section 2 we review the basic definitions and results on first-order dependence logic and its variants (extensions and fragments). It is divided into three subsections of which the two first ones deal with results related to *expressive power* and *definability*. In particular, results charting the expressive power of certain natural syntactic fragments of dependence logic and its variants will be discussed in Section 2.4. Section 2.5 reviews results on the complexity of *satisfiability* and *model checking* in the (first-order) dependence logic context. In Section 3 we turn to modal and propositional versions of dependence logic. After introducing the basic notions and logics, we will again first touch expressivity questions and then turn to the complexity of algorithmic problems arising in this context, mostly the complexity of satisfiability and model checking. The paper concludes in Section 4 with a list of open questions.

2 First-order Dependence Logic

2.1 Team semantics

In this section we define the basics of the team semantics as presented in the monograph [56] by Väänänen. The origins of this definition go back to a paper by Wilfrid Hodges [35], in which he gave a Tarski-style semantics for Hintikka and Sandu's *independence-friendly logic* $\mathcal{IF}$ [34]. Hodges originally used the term "trump semantics", somewhat reflecting the game-theoretic nature of the previously only known non-compositional semantics for $\mathcal{IF}$.

Definition 1. Let $\mathscr{M}$ be a structure with domain M, and V a finite set of variables. Then

- A *team* X over $\mathscr{M}$ with domain $\mathrm{Dom}(X) = V$ is a finite set of assignments $s\colon V \to M$.
- For a tuple $\mathbf{v} = (v_1, \ldots, v_n)$, where $v_i \in V$, $X(\mathbf{v}) := \{s(\mathbf{v}) : s \in X\}$ is an n-ary relation of M, where $s(\mathbf{v}) := (s(v_1), \ldots, s(v_n))$.
- For $W \subseteq V$, $X \upharpoonright W$ denotes the team obtained by restricting all assignments of X to W.
- The set of free variables of a formula ϕ is defined analogously as in first-order logic, and is denoted by $\mathrm{Fr}(\phi)$. In particular, all non-first-order atoms considered in this article (see Definition 3) are treated as atomic formulas, and hence all variable occurrences in them are considered to be free.

With the above notions defined, we are now ready to present the semantics of dependence logic. In this article we consider two variants of the semantics called the *strict* and the *lax semantics* introduced in [16]. The original semantics given in [56] is a combination of these variants (with the lax disjunction and the strict existential quantifier). For any logic, e.g., dependence logic, whose formulas have the downwards closure property of Proposition 4, the two variants of the semantics are easily seen to be equivalent. On the other hand, for inclusion and independence logic the semantics are not equivalent [16]. A serious disadvantage of the strict semantics is the failure of the locality property in the case of inclusion and independence logic (see Proposition 1).

We will first define the lax version of the team semantics for first-order formulas in negation normal form. For an assignment s, $\mathscr{M} \models_s \alpha$ below refers to satisfaction in first-order logic. We denote by $s[m/v]$ the assignment such that $s[m/v](x) = m$ if $x = v$, and $s[m/v](x) = s(x)$ otherwise.

Definition 2 (Lax Semantics). Let $\mathscr{M}$ be a structure, X a team over $\mathscr{M}$, and ϕ a formula such that $\mathrm{Fr}(\phi) \subseteq \mathrm{Dom}(X)$. Then X *satisfies* ϕ in $\mathscr{M}$, $\mathscr{M} \models_X \phi$, if

lit: For a first-order literal α, $\mathscr{M} \models_X \alpha$ if and only if for all $s \in X$, $\mathscr{M} \models_s \alpha$.
$\vee$: $\mathscr{M} \models_X \psi \vee \theta$ if and only if there are Y and Z such that $Y \cup Z = X$, $\mathscr{M} \models_Y \psi$ and $\mathscr{M} \models_Z \theta$.
$\wedge$: $\mathscr{M} \models_X \psi \wedge \theta$ if and only if $\mathscr{M} \models_X \psi$ and $\mathscr{M} \models_X \theta$.
$\exists$: $\mathscr{M} \models_X \exists v \psi$ if and only if there exists a function $F : X \to \mathscr{P}(M) \setminus \{\emptyset\}$ such that $\mathscr{M} \models_{X[F/v]} \psi$, where $X[F/v] = \{s[m/v] : s \in X, m \in F(s)\}$.
$\forall$: $\mathscr{M} \models_X \forall v \psi$ if and only if $\mathscr{M} \models_{X[M/v]} \psi$, where $X[M/v] = \{s[m/v] : s \in X, m \in M\}$.

A sentence is a formula without free variables. A sentence ϕ is *true* in $\mathscr{M}$ (abbreviated $\mathscr{M} \models \phi$) if $\mathscr{M} \models_{\{\emptyset\}} \phi$. Sentences ϕ and ϕ' are *equivalent*, $\phi \equiv \phi'$, if for all models $\mathscr{M}$, $\mathscr{M} \models \phi \Leftrightarrow \mathscr{M} \models \phi'$.

In the strict semantics, the semantic rule for disjunction is replaced by

$\mathscr{M} \models_X \psi \vee \theta$ if and only if, there are Y and Z such that $Y \cap Z = \emptyset$, $Y \cup Z = X$, $\mathscr{M} \models_Y \psi$ and $\mathscr{M} \models_Z \theta$,

and the semantic rule for existential quantifier by

$\mathscr{M} \models_X \exists v \psi$ if and only if, there exists a function $F : X \to \mathscr{P}(M) \setminus \{\emptyset\}$ such that $|F(s)| = 1$ for all $s \in X$, and $\mathscr{M} \models_{X[F/v]} \psi$.

It is worth noting that functions quantified in the strict semantics version of the existential quantifier correspond exactly to functions $F \colon X \to M$. Hence the notation $X[F/v]$ can be naturally extended to cover also functions $F \colon X \to M$.

The meaning of first-order formulas is invariant under the choice between the strict and the lax semantics. Furthermore, first-order formulas have the following *flatness property*:

Theorem 1 (Flatness). *Let $\mathcal{M}$ be a structure and X a team of $\mathcal{M}$. Then for a first-order formula ϕ the following are equivalent:*

1. $\mathcal{M} \models_X \phi$,
2. *For all $s \in X$, $\mathcal{M} \models_s \phi$.*

It is worth noting that in [56] also a general notion of flatness of a formula is defined by replacing the second item above by "For all $s \in X$, $\mathcal{M} \models_{\{s\}} \phi$".

Next we will give the semantic clauses for the non-first-order atoms and connectives considered in this paper. We begin with the new atomic formulas:

Definition 3.
- Let $\mathbf{x}$ be a tuple of variables and let y be another variable. Then $=(\mathbf{x}, y)$ is a *dependence atom*, with the following semantic rule:

  $\mathcal{M} \models_X =(\mathbf{x}, y)$ if and only if for all $s, s' \in X$, if $s(\mathbf{x}) = s'(\mathbf{x})$, then $s(y) = s'(y)$.

- Let $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$ be tuples of variables (not necessarily of the same length). Then $\mathbf{y} \perp_{\mathbf{x}} \mathbf{z}$ is a *conditional independence atom*, with the semantic rule

  $\mathcal{M} \models_X \mathbf{y} \perp_{\mathbf{x}} \mathbf{z}$ if and only if for all $s, s' \in X$ such that $s(\mathbf{x}) = s'(\mathbf{x})$, there exists a $s'' \in X$ such that $s''(\mathbf{xyz}) = s(\mathbf{xy})s'(\mathbf{z})$.

  Furthermore, we will write $\mathbf{x} \perp \mathbf{y}$ as a shorthand for $\mathbf{x} \perp_{\emptyset} \mathbf{y}$, and call it a *pure independence atom*.
- Let $\mathbf{x}$ and $\mathbf{y}$ be two tuples of variables of the same length. Then $\mathbf{x} \subseteq \mathbf{y}$ is an *inclusion atom*, with the semantic rule

  $\mathcal{M} \models_X \mathbf{x} \subseteq \mathbf{y}$ if and only if $X(\mathbf{x}) \subseteq X(\mathbf{y})$.

- Let $\mathbf{x}$ and $\mathbf{y}$ be two tuples of variables of the same length. Then $\mathbf{x} \mid \mathbf{y}$ is an *exclusion atom*, with the semantic rule

  $\mathcal{M} \models_X \mathbf{x} \mid \mathbf{y}$ if and only if $X(\mathbf{x}) \cap X(\mathbf{y}) = \emptyset$.

We denote the set of all dependence atoms by $=(\ldots)$. Analogously, all independence, inclusion and exclusion atoms are denoted by $\perp_c$, $\subseteq$, and $|$, respectively. For a collection $\mathscr{C} \subseteq \{=(\ldots), \perp_c, \subseteq, |\}$, we write $\mathcal{FO}(\mathscr{C})$ (omitting the set parenthesis of $\mathscr{C}$) for the logic obtained by adding all atoms listed in $\mathscr{C}$ to the syntax of first-order logic. Independence atoms (or independence logic) were first considered in [21], and inclusion atoms go back to [16]. In our notation, dependence logic, independence logic, and inclusion logic are denoted by $\mathcal{FO}(=(\ldots))$, $\mathcal{FO}(\perp_c)$, and $\mathcal{FO}(\subseteq)$, respectively. We also use the notation $\mathcal{D}$ as a shortcut for $\mathcal{FO}(=(\ldots))$. The fragment of independence logic containing only pure independence atoms in denoted $\mathcal{FO}(\perp)$.

Under the lax semantics, all of the above logics satisfy the following locality property [16]:

Proposition 1. *Let $\phi \in \mathcal{FO}(=(\ldots),\perp_c,\subseteq,|)$. Then for all models $\mathscr{M}$ and teams X,*

$$\mathscr{M} \models_X \phi \Leftrightarrow \mathscr{M} \models_{X\upharpoonright Fr(\phi)} \phi.$$

On the other hand, under the strict semantics Proposition 1 fails for inclusion and independence logic [16].

The aforementioned atoms are particular instances of a general notion of *generalized dependence atom* [44]. The semantics of a generalized dependence atom A_Q is determined (essentially) by a class Q of structures and teams over which the atomic formula $A_Q(\mathbf{x}_1,\ldots,\mathbf{x}_n)$ is satisfied (see [44] for details; we will consider $\mathcal{FO}$-definable generalized dependence atoms as depicted in Table 1).

Next we will define connectives and quantifiers that will also be discussed in the next section. One of the most natural extensions of dependence logic is obtained by the classical negation ($\sim$) with the usual interpretation:

$$\mathscr{M} \models_X \sim \phi \text{ iff } \mathscr{M} \not\models_X \phi.$$

This extension was introduced in [57], and the logic obtained was called *Team Logic* ($\mathcal{TL}$). The classical disjunction $\varovee$ (also sometimes referred to as intuitionistic disjunction) has also been considered especially in the modal team semantics context, see Section 3. The connective $\varovee$ has the expected interpretation

$$\mathscr{M} \models_X \phi \varovee \psi \text{ iff } \mathscr{M} \models_X \phi \text{ or } \mathscr{M} \models_X \psi.$$

In [1] two new connectives called the *intuitionistic* ($\rightarrow$) and the *linear implication* ($\multimap$) were introduced giving rise to an extension of dependence logic called BID:

$$\mathscr{M} \models_X \phi \rightarrow \psi \quad \text{iff} \quad \text{for all } Y \subseteq X, \text{ if } \mathscr{M} \models_Y \phi \text{ then } \mathscr{M} \models_Y \psi.$$
$$\mathscr{M} \models_X \phi \multimap \psi \quad \text{iff} \quad \text{for all } Y, \text{ if } \mathscr{M} \models_Y \phi \text{ then } \mathscr{M} \models_{X\cup Y} \psi.$$

Quantifiers, other than the familiar $\exists$ and $\forall$, have also been studied in the team semantics setting [12, 13]. From the complexity theoretic point of view, the following majority quantifier introduced in [7] is interesting:

$$\mathscr{M} \models_X \mathsf{M}x\phi(x) \text{ iff for at least } |M|^{|X|}/2 \text{ many functions } F\colon X \rightarrow M, \text{ we have}$$
$$\mathscr{M} \models_{X[F/x]} \phi(x).$$

2.2 Normal forms

In order to analyse the expressive power of dependence logic and to compare it with other formalisms, it is useful to obtain normal forms such as this one proved in [56].

Theorem 2. *Every dependence logic sentence is equivalent to some sentence of the form:*

$$\phi := \forall \mathbf{x} \exists \mathbf{y} \, (\bigwedge_{i \in I} {=}(\mathbf{x_i}, y_i) \wedge \theta)$$

where $I \subseteq \mathbb{N}$, $\mathbf{x_i}$ *is a subsequence of* $\mathbf{x}$, *and* y_i *is a member of* $\mathbf{y}$.

Such a result is an analogue of Skolem normal form for first-order logic. It separates clearly the functional dependencies introduced between subsets of variables from the regular part of the formula. It also makes intuitively clear that to be translated into an extension of first-order logic one would need second-order quantification to express these dependencies between variables. Refinements of such a normalization result are at the heart of various characterizations of dependence-like logics and their fragments. For example, the analogue of Theorem 2 for independence logic (with dependence atoms replaced by independence atoms) was shown in [23]. Furthermore, a prenex normal form theorem for formulas of $\mathcal{FO}({=}(\ldots), \perp_c, \subseteq)$ was shown for the strict and the lax semantics in [18] and [26], respectively.

2.3 Expressive Power

In this section we review results on the expressive power of the variants of dependence logic of the previous subsections.

As it turns out, the expressive power of sentences of dependence logic corresponds to that of existential second-order logic [56], and hence to the complexity class non-deterministic polynomial time (NP) via the well-known theorem of Fagin [14]. In the following, we will not distinguish in notation between a logic and the classes of models defined by its sentences, and we will use the equality symbol to denote that logics are equivalent for sentences, and we will use equality for logics and complexity classes in the same vein.

Theorem 3. $\mathcal{D} = \mathsf{NP} = \mathcal{ESO}$

The direction $\mathcal{ESO} \leq \mathcal{D}$ is proved by utilizing the fact that every $\mathcal{ESO}$-sentence can be transformed to the so-called Skolem normal form. On the other hand, the direction $\mathcal{D} \leq \mathcal{ESO}$ is proved by essentially simulating the team semantics of dependence logic in $\mathcal{ESO}$ with an extra relation symbol interpreting the team.

An interesting consequence of the team semantics of dependence logic is that Theorem 3 does not immediately settle the question also for open formulas. In fact, all $\mathcal{D}$-formulas have the following *Downwards Closure* property:

Theorem 4 (Downwards Closure). *Let* ϕ *be a* $\mathcal{D}$*-formula. Then for all structures* $\mathcal{M}$ *and teams* X, *if* $\mathcal{M} \models_X \phi$ *and* $Y \subseteq X$, *then* $\mathcal{M} \models_Y \phi$.

It was shown in [39] that the open formulas of dependence logic can define exactly the downward closed properties of teams expressible in $\mathcal{ESO}$ (again with an extra relation symbol for the team). Furthermore, already dependence atoms combined with disjunction give rise to NP-complete decision problems [37]. Define the formulas ϕ_1 and ϕ_2 as follows:

- $\phi_1 := \ =(x, y) \vee =(u, v)$,
- $\phi_2 := \ =(x, y) \vee =(u, v) \vee =(u, v)$.

Then the question of deciding whether a finite team X satisfies ϕ_1 is NL-complete, and for ϕ_2, already NP-complete.

As one might expect, the expressive power of dependence logic with the classical negation ($\mathcal{TL}$) increases to full second-order logic, and hence to the complexity class Polynomial Hierarchy (PH).

Theorem 5. $\mathcal{TL} = \mathcal{SO} = \mathsf{PH}$

This result is already shown in [56], but a direct translation of $\mathcal{SO}$ sentences into $\mathcal{TL}$-sentences was later given in [48]. Furthermore, in [38] it was shown that any property of teams definable in second-order logic can be expressed in team logic. It is worth noting that, for example, in general $\mathcal{TL}$-formulas are not closed downwards, e.g., the formula

$$\sim =(x) \tag{1}$$

expresses that x has at least two distinct values.

Interestingly, the two new connectives (implications) introduced in [1] preserve downwards closure when added to dependence logic. It was observed in [1] that any sentence of BID-logic can be translated into second-order logic. In fact, by the result of [62], already the intuitionistic implication alone increases the expressive power of dependence logic to full second-order logic.

Theorem 6. $\mathcal{D}(\rightarrow) = \mathcal{SO} = \mathsf{PH}$

This result utilizes the universal quantification implicit in the semantic rule of the intuitionistic implication. On the other hand, in [7, 8] the extension $\mathcal{D}(\mathsf{M})$ of dependence logic by the majority quantifier M was defined and studied. The main result of that paper is stated as follows:

Theorem 7. $\mathcal{D}(\mathsf{M}) = \mathsf{CH}$.

Above CH refers to the complexity class the counting hierarchy $\mathsf{CH} \supseteq \mathsf{PH}$. Theorems 7 and 5 imply that, for sentences, $\mathcal{D}(\mathsf{M})$ is at least as expressive as $\mathcal{TL}$ over finite structures. On the other hand, this result does not extend to open formulas since $\mathcal{D}(\mathsf{M})$-formulas have the downward closure property unlike $\mathcal{TL}$-formulas (see, e.g., formula (1)).

The aforementioned results show that dependence logic and its extensions allow us to logically characterize NP and some of its super classes. In [11] the question whether PTIME corresponds to a natural fragment of dependence logic was considered. For $\mathcal{ESO}$ (also $\mathcal{SO}$) it is known that the so-called Horn fragment $\mathcal{SO}\exists$-Horn of $\mathcal{ESO}$ captures PTIME over successor structures [19]. In [11] a fragment $\mathcal{D}^*$-Horn equivalent to $\mathcal{SO}\exists$-Horn was identified. The formulas of $\mathcal{D}^*$-Horn have the form

$$\forall \mathbf{x}\exists \mathbf{y}(\bigwedge_i =\!(\mathbf{z}_i, y_i) \wedge \bigwedge_j C_j),$$

where $\mathbf{z}_i$ is subsequence of $\mathbf{x}$, the clauses C_j (i.e. disjunctions of $\mathcal{FO}$-literals) are assumed to satisfy a certain Horn condition, and the existentially quantified variables y_i are only allowed to appear in certain identity atoms of C_j (see [11] for the exact definition). The main result of [11] shows that

Theorem 8. *Over finite successor structures, $\mathcal{D}^*$-Horn $=$ $\mathcal{SO}\exists$-Horn.*

Theorem 8 implies that

$$\mathcal{D}^*\text{-Horn} = \mathsf{PTIME}$$

over finite successor structures. In the article [11] the expressive power of open formulas of $\mathcal{D}^*$-Horn is also characterized.

All of the results discussed in this section use the original semantics of dependence logic. It is easy to check that the results hold also for both variants of the semantics. Next we will consider the expressive power of inclusion logic. It turns out that the expressive power of inclusion logic is not invariant under the choice between the strict and the lax semantics.

The expressive power of inclusion logic under the lax semantics was studied in [17]. The main result of that paper shows that

Theorem 9. *Over the lax semantics,*

$$\mathcal{FO}(\subseteq) = \mathrm{GFP}^+,$$

where GFP^+ is the so-called Positive Greatest Fixed Point Logic. It is known that over finite structures GFP^+ is equi-expressive with Least Fixed Point Logic (LFP), and furthermore for ordered finite structures LFP $=$ PTIME by the famous result of Immerman [36] and Vardi [60]. Therefore, it follows that

$$\mathcal{FO}(\subseteq) = \mathsf{PTIME}$$

over ordered finite structures. In drastic contract with Theorem 9, it was observed in [18] that, over the strict semantics, inclusion logic is equivalent to $\mathcal{ESO}$ and hence captures NP.

Theorem 10. *Over the strict semantics,*

$$\mathcal{FO}(\subseteq) = \mathcal{ESO}.$$

This result is based on a simulation of dependence atoms in a dependence logic sentence (in the $\forall^*\exists^*$-normal form) by certain inclusion logic formulas. This simulation is not possible in general but only over teams that are generated by evaluating a $\forall^*\exists^*$-block of quantifiers with the strict semantics.

2.4 Refining the correspondence with $\mathcal{ESO}$

In this part we investigate how the correspondence between existential second-order logic and $\mathcal{FO}(\mathscr{C})$ for subsets $\mathscr{C}$ of dependence-like atoms can be refined. In particular we examine what is the effect of bounding the number of variables and the so-called arity of atoms which roughly concerns the number of distinct variables involved in them.

By relating fragments of $\mathcal{FO}(\mathscr{C})$ to fragments of existential second-order logic, one may hope to obtain separation results in dependence logics through hierarchy theorems in complexity or to give evidence that such results would have non-trivial consequences in complexity theory. In either way, this provides interesting insight on the expressive power of these logics.

Let us first define the notion of arity of an atom.

Definition 1. Let $k \in \mathbb{N}$.

- A dependence atom $=(\mathbf{x}, y)$ is of arity k if the length of $\mathbf{x}$ is k.
- An independence atom $\mathbf{y} \perp_{\mathbf{x}} \mathbf{z}$ is of arity k if $\mathbf{xyz}$ contains $k+1$ distinct variables.
- An inclusion atom $\mathbf{x} \subseteq \mathbf{y}$ is of arity k if the length of $\mathbf{x}$ and $\mathbf{y}$ is k.

We now define the corresponding fragments of $\mathcal{FO}(\mathscr{C})$ and existential second-order logic.

Definition 2. Let $\mathscr{C}$ be a subset of $\{=(\ldots), \perp_c, \subseteq, |\}$. Let $k \in \mathbb{N}$. Then:

- $\mathcal{FO}(\mathscr{C})(k\text{−ary})$ is the class of sentences of $\mathcal{FO}(\mathscr{C})$ in which all atoms of $\mathscr{C}$ are of arity bounded by k.
- $\mathcal{FO}(\mathscr{C})(k\forall)$ is the class of sentences of $\mathcal{FO}(\mathscr{C})$ in which every variable is quantified exactly once and at most k universal quantifiers occur.
- For convenience, we set by $\mathcal{D}(k\text{−ary})$ the class $\mathcal{FO}(=(\ldots))(k\text{−ary})$ and by $\mathcal{D}(k\text{−ary})$ the class $\mathcal{FO}(=(\ldots))(k\forall)$.

Definition 3. Let $k \in \mathbb{N}$.

- $\mathcal{ESO}(k\text{−ary})$ is the class of $\mathcal{ESO}$-sentences

$$\exists X_1 \ldots \exists X_n \psi,$$

in which the relation symbols X_i are at most k-ary and ψ is a first-order formula.

- $\mathcal{ESO}_f(k\text{−ary})$ is the class of $\mathcal{ESO}$-sentences

$$\exists f_1 \ldots \exists f_n \psi,$$

in which the function symbols f_i are at most k-ary and ψ is a first-order formula.
- $\mathcal{ESO}_f(m\forall)$ is the class of $\mathcal{ESO}$-sentences in Skolem normal form

$$\exists f_1 \ldots \exists f_n \forall x_1 \ldots \forall x_r \psi,$$

where $r \leq m$.

Such fragments of $\mathcal{ESO}$ have been widely studied and in particular their relationship with complexity classes. Roughly speaking, controlling the number of first-order variables in existential second-order logic amounts to control the polynomial degree in non-deterministic polynomial-time computations. To be more precise, it is known (see[22]) that, for $k \geq 1$:

$$\mathcal{ESO}_f(k\text{-ary}, k\forall) = \mathcal{ESO}_f(k\forall) = \mathsf{NTIME}_{\mathrm{RAM}}(n^k).$$

where $\mathsf{NTIME}_{\mathrm{RAM}}(n^k)$ is the class of problems decidable by a non-deterministic random access machine in time $O(n^k)$. We can now relate the expressive power of these various fragments.

First, reusing of variables is a key issue in team semantics. It turns out that the following result is true (see [6]).

Proposition 2. *Any sentence of dependence logic is logically equivalent to a sentence in which at most one variable is universally quantified (possibly several times).*

For what concerns dependence logic, the correspondence with existential second-order logic for the fragments with bounded arity and bounded number of universal variables is as follows [6]:

Theorem 11. *For all integers $k \geq 1$,*

- $\mathcal{D}(k\text{−ary}) = \mathcal{ESO}_f(k\text{−ary})$,
- $\mathcal{D}(k\forall) \leq \mathcal{ESO}_f(k\forall) \leq \mathcal{D}(2k\forall)$.

Roughly speaking, dependence logic can be seen as existential second-order logic with functions (dependence atoms) but "without proper names" for these functions. Hence composition of functions, that can be done freely in existential second-order logic, can be simulated only by using intermediate variables in dependence logic. So, as long as, only the arity of dependence atoms is fixed, one can obtain an exact correspondence between the fragments (as stated in Theorem 11). By contrast an exact correspondence between the fragments $\mathcal{D}(k\forall)$ and $\mathcal{ESO}_f(k\forall)$ seems unlikely (see the second item of Theorem 11). However, it is possible to establish an exact correspondence between $\mathcal{D}(k\forall)$ and some syntactically restricted fragment of $\mathcal{ESO}_f(k\forall)$ (see [6]).

Because inclusion and independence logic do not have the downward closure property, the situation is drastically different depending on whether the lax or the strict semantics are used (see [18, 24–26]).

Theorem 12. *Let $k \geq 1$. The following holds in the lax semantics:*

- $\mathcal{FO}(\subseteq)(k\text{−ary}) < \mathcal{FO}(\subseteq)(k+1\text{−ary})$,
- $\mathcal{FO}(\subseteq)(k\text{−ary}) \leq \mathcal{ESO}_f(k\text{−ary}) = \mathcal{FO}(\perp_c)(k\text{−ary})$,
- $\mathcal{FO}(\perp)(2\forall) = \mathcal{FO}(\perp)$,
- $\mathcal{FO}(\subseteq)(1\forall) = \mathcal{FO}(\subseteq)$.

For the strict semantics, the following results are true:

- $\mathcal{FO}(\subseteq)(k\forall) = \mathcal{ESO}_f(k\forall) = \mathsf{NTIME}_{\mathrm{RAM}}(n^k)$,
- $\mathcal{FO}(\perp_c)(k\forall) \leq \mathcal{ESO}_f((k+1)\forall)$,
- $\mathcal{ESO}_f(k\forall) \leq \mathcal{FO}(\perp_c)(2k\forall)$.

The results above imply that there is an infinite expressivity hierarchy for $\mathcal{D}(k\forall)$, $\mathcal{FO}(\subseteq)(k\forall)$, and $\mathcal{FO}(\perp_c)(k\forall)$. Indeed, it is well known (by a slight adaptation of classical non-deterministic time hierarchy [5]) that, for any integer $k \geq 1$, $\mathsf{NTIME}_{\mathrm{RAM}}(n^k)$ is strictly included in $\mathsf{NTIME}_{\mathrm{RAM}}(n^{k+1})$. Hence, for example, $\mathcal{D}(k\forall)$ is strictly less expressive than $\mathcal{D}((k+1)\forall)$.

Similarly, one might ask whether there is a strict hierarchy based on arity for $\mathcal{D}$ or $\mathcal{FO}(\perp_c)$. For example, is $\mathcal{D}(k+1\text{−ary})$ strictly more expressive than $\mathcal{D}(k\text{−ary})$ for all (or some) $k \geq 1$. Such a hierarchy would imply the existence of a similar hierarchy for $\mathcal{ESO}_f(k\text{−ary})$ which is a long-standing open question (for empty signature, this is known as the *Spectrum Arity Hierarchy Conjecture* [15]).

Finally, let us examine the situation when exclusion atoms are allowed in the syntax. It turns out that none of the approach above helps to control the arity correspondence between the corresponding fragments. By introducing mainly two new concepts, namely inclusion quantifier (an adaptation of the idea of quantifier relativization applied to inclusion atoms) and term value preserving disjunction, the following result is obtained in [51]:

Theorem 13. *For all integers $k \geq 1$, $\mathcal{FO}(\subseteq, |)(k\text{−ary}) = \mathcal{ESO}(k\text{−ary})$.*

Note that in this result the correspondence is with the relational fragment of existential second-order logic: no quantification on functions is allowed.

2.5 Satisfiability and Model Checking

In this section we briefly review results on satisfiability and model checking in the first-order dependence logic context.

We begin by recalling these problems for a logic $\mathscr{L}$:

- The *Satisfiability Problem* SAT[$\mathscr{L}$] is defined as

$$\mathrm{SAT}[\mathscr{L}] := \{\phi \in \mathscr{L} \mid \text{there is a structure } \mathscr{M} \text{ such that } \mathscr{M} \models \phi\}.$$