Coherence and Complexity in Fragments of Dependence Logic

Jarmo A. Kontinen

Coherence and Complexity in Fragments of Dependence Logic

ILLC Dissertation Series DS-2010-05

INSTITUTE FOR LOGIC, LANGUAGE AND COMPUTATION

For further information about ILLC-publications, please contact

Institute for Logic, Language and Computation
Universiteit van Amsterdam
Science Park 904
1098 XH Amsterdam
phone: +31-20-525 6051
fax: +31-20-525 5206
e-mail: illc@uva.nl
homepage: http://www.illc.uva.nl/

Coherence and Complexity in Fragments of Dependence Logic

Academisch Proefschrift

ter verkrijging van de graad van doctor aan de
Universiteit van Amsterdam
op gezag van de Rector Magnificus
prof.dr. D.C. van den Boom
ten overstaan van een door het college voor
promoties ingestelde commissie, in het openbaar
te verdedigen in de Agnietenkapel
op dinsdag 22 juni 2010, te 12.00 uur

door

Jarmo Antero Kontinen

geboren te Pieksämäki, Finland.

Promotiecommissie:

Promotor:
Prof. dr. J. Väänänen

Overige leden:
Prof. dr. J.F.A.K van Benthem
Prof. dr. L. Hella
Dr. T. Janssen
Prof. dr. M. Krynicki

Faculteit der Natuurwetenschappen, Wiskunde en Informatica
Universiteit van Amsterdam
Science Park 904
1098 XH Amsterdam

Printed and bound by Ipskamp Drukkers.

ISBN: 978-90-5776-210-9

Contents

Acknowledgments

The interesting adventure of being a PhD student started for me in the Department of Mathematics and Statistics at the University of Helsinki. After few years I moved to Amsterdam and continued my research at the Institute for Logic, Language and Computation (ILLC). Doing a PhD in mathematical logic turned out to be a good experience in my life, although not always easy. There were many people who I met during this time and who in various senses contributed to the quality of my life or my research.

First of all I would like to thank to my supervisor Jouko Väänänen. The inspiration and help with scientific as well as academic matters I got from him cannot be overstated. I am thankful to Jouko for the time he devoted to discuss my research with me and for all the comments he gave on my thesis. I want to show also my gratitude to my brother Juha who also helped me a lot, particularly by being always there when I faced a mathematical problem which needed to be discussed. I would like to thank all the members of my promotion committee for agreeing to be in the committee and to participate in the defence: Johan van Benthem, Lauri Hella, Theo Janssen and Michał Krynicki. I am particularly grateful to Lauri Hella for his help and useful comments on the manuscript of my dissertation.

While doing my PhD I had to deal not only with difficult mathematical problems but also with many, equally difficult, issues connected to everyday life. I would like to thank to Tanja, Karin and Peter from ILLC for helping me in various practical matters. I am grateful to Michal Lukasiewicz for translating my abstract into Dutch. I want to express also my gratitude to Jouko Väänänen and Juliette Kennedy who were helping me in many ways while I was living in Amsterdam.

My research would not have been possible without financial support. I am grateful to Jouko for his help with organizing funding for me during my PhD-studies. My research was also partly funded by Finnish Academy of Science and Letters, *Vilho, Yrjö and Kalle Väisälä Foundation.*

I met many of my good friends while doing my studies in Helsinki. I am still looking back with warmth to the times spend with Jarno, Jarkko, Juha, Minna, Mikko, Pekka, Samuli, Simo, Tiina, Toni and Teppo. I have also had the pleasure of getting to know many nice people while staying in Amsterdam. I am happy to have met Marysia who was also at that time studying in ILLC. I am grateful to her for all the time we have shared, and for her help and support. I am grateful to Lauri Keskinen with whom we have spent numerous evenings playing go, chess, magic, pool, poker, etc. It was very nice to have another Finnish-speaking student around. I would like to thank Andy and Jakub for being great office mates, particulary I am grateful to Jakub for being always helpful and for all the discussion we had on computational complexity theory. I would also like to thank Brian, Cedric, Eelco, Daisuke, Jakub, Jonathan, Lauri, Lena, Luis, Marysia, Michal, Nina, Raul and Szymon for all the wonderful evenings spent on games, parties, climbing, discussions, etc. over these past three years.

Finally i would like to thank my parents for all their love and support.

Jarmo Kontinen
May, 2010.

Chapter 1

Introduction

We investigate the properties of *dependence logic* ($\mathcal{D}$) introduced by Väänänen in [39]. We focus mainly on finite structures. Dependence logic incorporates explicit dependence statements into first order logic (FO) by means of so-called *dependence atoms*. Dependence atoms are atomic formulas of the form

$$=(t_1, \ldots, t_n) \tag{1.1}$$

which have the intuitive meaning that the values of the terms $t_1, \ldots, t_{n-1}$ determine the value of the term t_n, or alternatively, the value of the term t_n depends only on the values of the terms $t_1, \ldots, t_{n-1}$. The dependence atoms provide, similarly to the so-called partially ordered quantifiers [19], a way to express dependence (independence) between variables that cannot be expressed in first-order logic.

The research on partially ordered (branching) quantifiers provides the historical background for dependence logic: Henkin initiated the research on partially ordered quantifiers in [19]. He considered formulas as below

$$\begin{pmatrix} \forall x_1 & \exists y_1 \\ \forall x_2 & \exists y_2 \end{pmatrix} \psi(x_1, x_2, y_1, y_2) \tag{1.2}$$

where the value of y_1 depends only on x_1 and the value of y_2 depends only on x_2. The meaning of sentences as (1.2) can be given in terms of Skolem-functions:

$$\exists f_{y_1} \exists f_{y_2} \forall x_1 \forall x_2 \phi(x_1, f_{y_1}(x_1), x_2, f_{y_2}(x_2)) \tag{1.3}$$

Which again, has the following reformulation in dependence logic:

$$\forall x_1 \exists y_1 \forall x_2 \exists y_2 (=(x_2, y_2) \wedge \psi(x_1, x_2, y_1, y_2)) \tag{1.4}$$

Ehrenfeucht was the first to show that branching quantification properly extends the expressive power of classical first order logic. He showed that the cardinality quantifier Q_0 expressing infinity; $Q_0 x\phi(x)$ is true if and only if there are infinitely many x such that $\phi(x)$, is definable in terms of branching quantifiers, see [19].

So-called independence friendly logic (IF logic), introduced by Hintikka and Sandu in [20], [21] expresses the independence between variables in terms quantifiers of $\exists y/\forall x$:

$$\forall x_1 \exists y_1 \forall x_2 \exists y_2 / \forall x_1 \ \phi(x_1, x_1, y_1, y_2) \tag{1.5}$$

In the formula (1.5), the value of y_2 has to be chosen independently of the value of x_1. Hintikka and Sandu gave semantics for IF logic-sentences in terms of semantic games. Later, Hodges gave a compositional semantics for all IF logic formulas in terms of sets of assignments, so-called trumps [22].

The branching quantifiers increase the expressive power of FO considerably as indicated by Ehrenfeucht's result. It was shown by Enderton [8] and by Walkoe [40] that first-order sentences with partially ordered quantifier prefixes are equivalent to Σ_1^1-sentences. Hodges showed that every formula of IF logic has can be represented in an equivalent form in Σ_1^1 with an extra relation symbol interpreting the trump [22]. He also raised a question which teams are definable in IF logic by means of identity only. Recently Väänänen and Kontinen showed that these are exactly the teams which are definable in Σ_1^1 with an extra predicate, occurring only negatively, for the trump [31].

The semantics for dependence logic can be given in terms of sets of assignments, so-called teams, and in terms of semantic games. In this thesis we consider only the Team-semantics in which formulas are evaluated with respect to sets of assignments. An alternative way to define semantics for dependence logic formulas is in terms of semantic games. In game theoretic semantics the satisfiability of a formula ϕ in a structure is represented as a game between two players, known as "Verifier" and "Falsifier". Falsifier doubts the verity of ϕ, while Verifier tries to show it to be true. Formula ϕ will be considered true when Verifier has a winning strategy, while it will be false whenever Falsifier has a winning strategy.

The semantic games for dependence logic formulas are not determined, i.e. it does not hold for every formula and structure that either of the players always has a winning strategy. Another important feature of these games is the amount of information available for the players during the game. A game is said to be of perfect information when the players know all the previous moves of the game, both their own and their opponents. Otherwise a game is said to be of imperfect information. The semantic game for dependence logic formulas can be given both as a game of perfect information and as a game of imperfect information (for more details see [39]).

Teams and other central notions of $\mathcal{D}$-formulas will be defined in Chapter 2 of this thesis. We will recall some properties and results concerning dependence logic given in [39], which we will need later in the proofs.

In Chapter 3, we investigate so-called *coherent* formulas of dependence logic which can be evaluated locally in teams. A k-coherent formula ϕ is true in a team

$\mathcal{X}$ if and only if it is true in every k-element sub-team of $\mathcal{X}$. This is an important feature as the teams can be very large, even infinite in infinite domains. Thus it is convenient to be able to restrict to some small finite sub-teams when evaluating the satisfiability of a formula. This becomes especially relevant when considering the computational complexity of a formula.

We will go trough the connectives and quantifiers of $\mathcal{D}$ and study the effect they have on coherence of the formula. We will observe that the satisfiability of all dependence atoms can be checked in restricting to two-element sub-teams. As for all first-order formulas we can even restrict to singleton sub-teams. We will show that the universal quantification and the conjunction preserve coherence whereas existential quantification and disjunction do not. We will also give an example of an incoherent formula. We introduce a way to interpret a given team and dependence atoms into a multigraph so that the semantics of the disjunction and the colorability of the graph match perfectly. This turns out to be very useful way to represent teams when considering dependence atoms.

Furthermore, we will show that all coherent formulas are equivalent to first-order sentences when given an additional relation interpreting the team. Coherence will be also essential notion in our classification of the computational complexity of quantifier-free formulas in Chapter 4.

There are many interesting questions in computational complexity theory, which can be approached by using logic. The field of finite model theory that studies the connection between computational complexity and logical definability is so-called *descriptive complexity theory*. One central notion studied in this field is the *model checking problem* of a logic. The model checking problem for a logic L and a class of finite structures $\mathbf{M}$ is the following: Given a structure $\mathcal{M} \in \mathbf{M}$ and a formula $\phi \in L$, determine whether or not it holds

$$\mathcal{M} \models \phi. \tag{1.6}$$

On the other hand, if we fix ϕ and let the structure $\mathcal{M}$ vary, then ϕ itself corresponds to a computational problem and the logic L to a class of problems. We now arrive at the concept of a logic L characterizing a complexity class; we say that a logic L characterizes a computational complexity class C if and only if the classes of finite structures definable in L are exactly the ones that can be recognized in C under some uniform encoding of finite structures into strings.

For most of the known computational complexity classes characterizing logics has been discovered. The characterizations usually assume the presence of an order over the domains of the structures. What is surprising is that we are not able to distinguish many of these classes from each other although we have characterizing logics for most of them. An important open question is the question if the properties computable in *polynomial time* (PTIME) coincides with the properties computable in *non-deterministic polynomial time* (NP). Another important open problem is to find a characterizing logic for polynomial time over all finite structures, i.e. not assuming the presence of an order.

Fagin's classical result [9] establishes a perfect match between Σ^1_1-formulas and languages in NP, i.e the problems definable by Σ^1_1-formulas over all finite structures are exactly the ones recognized in non-deterministic polynomial time. There is a connection between Σ^1_1-formulas and formulas defined with partially ordered quantifiers, namely everything that can be defined in Σ^1_1 can be defined also using partially ordered quantifiers and vice versa. This was first shown by Enderton [8] and Walkoe [40]. This equivalence holds also for dependence logic [39]. In other words, the properties definable in dependence logic over finite structures are exactly the ones that are computable in non-deterministic polynomial time. This makes dependence logic a very interesting logic from the point of view of descriptive complexity theory. Potentially all results on hierarchies of fragments of dependence logic can give us valuable information about NP.

We seek to characterize the computational complexity of the model checking problem for quantifier-free $\mathcal{D}$-formulas. We will discover three thresholds inside $\mathcal{D}$. Namely, when the computational complexity of the model checking can be done in logarithmic space (LOGSPACE), non-deterministic logarithmic space (NL) or NP. We will also give complete instances for NL and NP. We will also give some results on the combined complexity of the model checking of coherent formulas over all finite teams. We will mainly consider the effect of the disjunction on the combined complexity of certain classes of coherent formulas. We will give some boundaries when the combined complexity of the model checking is in NL and when it becomes complete for NP.

In chapter 5 we will look into dependence logic and 0-1 law. A logic L is said to have the 0-1 law if every sentence of L is either true in almost all finite models, or false in almost all finite models, i.e. when the size of the structure n tends to infinity, the proportion of n-element structures which satisfy ϕ over all structures of size n tends to 0 or 1. The 0-1 law is also a method for proving non-definability, which can be difficult to show otherwise on finite structures. For example, when a logic has the 0-1 law, one cannot define even cardinality with a sentence of that logic. The 0-1 law for first-order sentences was shown by Glebskii, Kogan, Liogon'kii and Talanov [15] and later independently by Fagin [10].

We will show that the set of universal and existential sentences of dependence logic have the 0-1 law. Furthermore we will show that all quantifier-free formulas of dependence logic have the 0-1. We will also point out the least fragment of $\mathcal{D}$, which is known to fail the 0-1 law.

In Chapter 6 of the thesis we will investigate whether the 0-1 law holds for the logic $\mathcal{L}^k_{\infty\omega}$ extended by a simple unary generalized quantifier over uniform distribution of finite graphs. We concentrate on a specific class of quantifiers of the form $\exists^{s/t}$, which allow us to express properties like for example " *At least half* of the people in ILLC are interested in quantifiers" or "*At most five out of eleven* of mathematicians are interested in philosophy."

We will show that a dichotomy holds for the 0-1 law of the logics $\mathcal{L}^k_{\infty\omega}(\exists^{s/t})$, i.e. when t is not of form 2^m for any $m \in \mathbb{N}$ the logic $\mathcal{L}^k_{\infty\omega}(\exists^{s/t})$ has the 0-1 law.

On the other hand, if t is of the form 2^m for some $m \in \mathbb{N}$, then $\mathcal{L}^k_{\infty\omega}(\exists^{s/t})$ has the 0-1 law if and only if $log_2 t \geq k$, i.e. if the number of variables in the sentences is less or equal to $log_2 t$.

In the last part of this thesis we will discuss a work in progress. We will discuss the problem of finding an axiomatization for fragments of dependence logic and the the connection of dependence logic to database theory. The high expressive power of dependence logic yields a consequence that the whole logic cannot be effectively axiomatized. Thus, we will look for fragments of dependence logic which are potentially axiomatizable. We will start from the dependence atoms and build up the fragment by adding more connectives.

We use so-called Armstrong's axioms for functional dependencies to axiomatize the consequence relation between dependence atoms. Functional dependencies play a important part in database designing as well as in database normalization and denormalization in data base theory. The most common model for a database is the *relational model* [5]. The model describes a database by a collection of predicates over a finite set of predicate variables. A *functional dependence* is a constraint over the values of two sets of variables, denoted with

$$X \to Y \tag{1.7}$$

If R is a relation over a set of variables U, such that $X, Y \subseteq U$, then R is said to obey the functional dependence (1.7), if for all tuples $s, s' \in R$, if the tuples agree on the variables in X, then they also agree on the variables in Y. Functional dependencies like (1.7) are expressible in dependence logic by means of the dependence atoms in the following way.

$$\bigwedge_{y \in Y} {=}(X, y) \tag{1.8}$$

Functional dependencies have a sound and complete axiomatization, so-called Armstrong's axioms [1]. The Armstrong's axioms can be adopted for the dependence atoms and with some additional rules they provide a sound and complete axiomatization of dependence atoms. We will give the Armstrong's axioms and go through the completeness proof and discuss the problem of finding axiomatization for fragments of dependence logic.

Another possible direction for future work is the study of other dependence-type properties. In natural language there are several similar concepts and expressions to "dependence", e.q. "totally determines", "function of", "mutual dependence", "liable to", etc., which one could try formalize and incorporate into logic. We will consider one such notion what we call "mutual dependence": Two variables x and y are mutually dependent if it holds that the value of x determines functionally the value of y and the value of y determines functionally the value of x. We will show that mutual dependence is axiomatizable analogous to Armstrong's axioms.

Chapter 2

Preliminaries

We will start by defining the elementary concepts like vocabulary and structures and then proceed to define the syntax and the semantics of dependence logic. After this we will recall some fundamental properties of $\mathcal{D}$ from [39], which will be used later in the proofs.

Definition 2.0.1. *A vocabulary* τ is a set of constant, relation and function symbols. Constants are denoted with c_i, relations with R_i and functions with f_i, $i \in \mathbb{N}$. The arity of the relation symbol R is denoted with $\#R$ and respectively, $\#f$ for a function symbol f. We denote variables with x_i, $i \in \mathbb{N}$.

Definition 2.0.2. Suppose τ is a vocabulary. A τ*-structure* $\mathcal{M}$ is defined to be a non-empty set M, the domain of $\mathcal{M}$, endowed with an element $c_i^{\mathcal{M}}$ of M for each $c_i \in \tau$, an $\#R$-ary relation on M for $R_i \in \tau$, and $f_i^{\mathcal{M}}$ a $\#f$-ary function on M for $f_i \in \tau$.

Definition 2.0.3. (*Terms*) Suppose τ is a vocabulary. Then all constant symbols $c_i \in \tau$ and variables are τ-terms. If $t_1, \ldots, t_n$ are τ-terms, then $ft_1 \ldots t_n$ is a τ-term for each $f \in \tau$ of arity n.

The syntax of $\mathcal{D}$ extends FO with the dependence atoms. We assume the negation normal form, .ie., negation is assumed to appear only in front of atomic formulas.

Definition 2.0.4. (*Formulas*) Suppose τ is a vocabulary. If $t_1, \ldots, t_n$ are τ-terms and R a n-ary relation symbol in τ. Then

$$\begin{aligned}
\approx t_i t_j,\\
=(t_1,\ldots,t_n),\\
Rt_1\ldots t_n\\
\neg\approx t_i t_j,\\
\neg=(t_1,\ldots,t_n),\\
\neg Rt_1\ldots t_n
\end{aligned}$$

are τ-formulas of $\mathcal{D}$. If ϕ and ψ are τ-formulas of $\mathcal{D}$, then

$$\begin{aligned}
(\phi\vee\psi),\\
(\phi\wedge\psi),\\
\forall x\phi,\\
\exists x\phi,
\end{aligned}$$

are τ-formulas of $\mathcal{D}$. We define $\top$ as the dependence atom $=(\ )$ and $\perp$ as the negated dependence atom $\neg=(\ )$. We denote the set of all τ-formulas of $\mathcal{D}$ with $\mathcal{D}(\tau)$ and the set of free variables of ϕ by $Fr(\phi)$.

The semantics first order logic formulas was defined by Tarski [36] in terms of assignments. Semantics for $\mathcal{D}$ can be given in terms of sets of assignments, so-called *teams* and in terms of semantic games [39]. We adopt the team-semantics which is in analogue to *Trump-semantics* defined by Hodges for IF logic [22].

Definition 2.0.5. (*Assignments* and *Teams*) Let $V=\{x_i \mid i\in n\}$, $n\in\mathbb{N}$, be a set of variables and $\mathcal{M}$ a structure with domain M. Then, *an assignment s with domain V and range M* is a function $s: V\to M$. *A team $\mathcal{X}$ with a domain V, and range M* is any set of assignments with domain V and range M. We denote the domain of the team $\mathcal{M}$ with $\mathrm{dom}(\mathcal{X})$ and the range of a team $\mathcal{X}$ with $\mathrm{range}(\mathcal{X})$. We use the following notation when the team is given as a relation:

$$Rel(\mathcal{X})=\{(s(x_0),\ldots,s(x_{n-1}))\mid s\in\mathcal{X}\}.$$

Definition 2.0.6. Suppose $s: V\to M$ is an assignment and $W\subseteq V$. Then

$$s\lceil W=\{(v,a)\in s\mid v\in W\}.$$

Suppose $\mathcal{X}$ is a team on domain V. Then

$$\mathcal{X}\lceil W=\{s\lceil W\mid s\in\mathcal{X}\}.$$

Definition 2.0.7. Suppose $\mathcal{M}$ and $\mathcal{N}$ are structures, $f : \mathcal{M} \to \mathcal{N}$ a function and $\mathcal{X}$ is a team with range M. Then

$$f[\mathcal{X}] = \{f \circ s \mid s \in \mathcal{X}\}.$$

The following two operations of teams, *duplication* and *supplementation*, correspond to universal quantification and respectively to existential quantification [39].

Definition 2.0.8. (*Supplementation*) Suppose $\mathcal{X}$ is a team of domain V and range M and $F : \mathcal{X} \to M$ a function. Let $\mathcal{X}(F, x_n)$ denote the *supplement team*

$$\{s(F(s)/x_n) \mid s \in \mathcal{X}\},$$

where $s(F(s)/x_n)$ is the assignment obtained by replacing $(x_n, s(x_n))$ in s with $(x_n, F(s))$.

Definition 2.0.9. (*Duplication*) Suppose $\mathcal{X}$ is a team of domain V and range M. Let $\mathcal{X}(M, x_n)$ denote the *duplicated team*

$$\{s(a/x_n) \mid a \in M,\ s \in \mathcal{X}\},$$

where $s(a/x_n)$ is the assignment obtained by replacing $(x_n, s(x_n))$ in s with (x_n, a).

Definition 2.0.10. (*Semantics*) Suppose τ is a vocabulary, $\mathcal{X}$ a team of domain V and range M, $\mathcal{M}$ a τ-structure and ϕ and θ formulas of $\mathcal{D}(\tau)$. The semantics of $\mathcal{D}$-formulas are defined in the following way:

1. $=(\,)$ is assumed universally true.
2. $\mathcal{M} \models_{\mathcal{X}} \ =(t)$, iff for all $s, s' \in \mathcal{X}$ it holds $s(t) = s'(t)$. We will call such terms constant dependencies.
3. $\mathcal{M} \models_{\mathcal{X}} \ =(t_1, \ldots, t_n)$, $n > 1$, iff for all $s, s' \in \mathcal{X}$ it holds that, if $s(t_i) = s'(t_i)$ for $i \leq n-1$, then $s(t_n) = s'(t_n)$.
4. $\mathcal{M} \models_{\mathcal{X}} \neg =(t_1, \ldots, t_n)$ iff $\mathcal{X} = \emptyset$.
5. $\mathcal{M} \models_{\mathcal{X}} \ \approx t_1 t_2$, iff for every $s \in \mathcal{X}$, $s(t_1) = s(t_2)$.
6. $\mathcal{M} \models_{\mathcal{X}} \neg \approx t_1 t_2$, iff for every $s \in \mathcal{X}$, $s(t_1) \neq s(t_2)$.
7. $\mathcal{M} \models_{\mathcal{X}} R(t_1, \ldots, t_n)$, iff for every $s \in \mathcal{X}$, $(s(t_1), \ldots, s(t_n)) \in R^{\mathcal{M}}$.
8. $\mathcal{M} \models_{\mathcal{X}} \neg R(t_1, \ldots, t_n)$, iff for every $s \in \mathcal{X}$, $(s(t_1), \ldots, s(t_n)) \notin R^{\mathcal{M}}$.
9. $\mathcal{M} \models_{\mathcal{X}} \phi \wedge \theta$, iff $\mathcal{M} \models_{\mathcal{X}} \phi$ and $\mathcal{M} \models_{\mathcal{X}} \theta$.

10. $\mathcal{M} \models_{\mathcal{X}} \phi \vee \theta$, iff there exists $\mathcal{Y}$ and $\mathcal{Z}$, such that $\mathcal{Y} \cup \mathcal{Z} = \mathcal{X}$, $\mathcal{M} \models_{\mathcal{Y}} \phi$ and $\mathcal{M} \models_{\mathcal{Z}} \theta$.
11. $\mathcal{M} \models_{\mathcal{X}} \exists x\phi(x)$, iff there is $F : V \to M$, such that $\mathcal{M} \models_{\mathcal{X}(F,x)} \phi(x)$.
12. $\mathcal{M} \models_{\mathcal{X}} \forall x\phi(x)$, iff $\mathcal{M} \models_{\mathcal{X}(M,x)} \phi(x)$.

Note that it follows from the definition that the empty team satisfies all formulas of $\mathcal{D}$. Sentences are evaluated in terms of the team $\{\emptyset\}$ containing just the empty assignment.

Theorem 2.0.11. *[39] For all sentences $\phi \in \mathcal{D}$, structures $\mathcal{M}$ and teams $\mathcal{X}$ the following holds:*

$$\mathcal{M} \models_{\mathcal{X}} \phi \Leftrightarrow \mathcal{M} \models_{\{\emptyset\}} \phi.$$

2.1 Properties of dependence logic

The properties introduced in this section are given in [39].

Definition 2.1.1. (*Logical consequence* and *equivalence*) Let $\phi \in \mathcal{D}$ and $\theta \in \mathcal{D}$. Formula θ is a *logical consequence* of ϕ,

$$\phi \Rightarrow \theta,$$

if for all structures $\mathcal{M}$ and teams $\mathcal{X}$ with $dom(\mathcal{X}) \supseteq Fr(\phi) \cup Fr(\theta)$, $\mathcal{M} \models \phi$ implies $\mathcal{M} \models \theta$. The formulas θ and ϕ are *logically equivalent* if $\phi \Rightarrow \theta$ and $\theta \Rightarrow \phi$.

The satisfiability of a $\mathcal{D}$-formula depends only on the interpretation of the free variables of the formula. It was shown by Nurmi in his dissertation [28] that the proof needs the Axiom of Choice.

Theorem 2.1.2. *[39] Let V be a set of variables, such that $Fr(\phi) \subseteq V$. Then*

$$\mathcal{M} \models_{\mathcal{X}} \phi \Leftrightarrow \mathcal{M} \models_{\mathcal{X}\lceil V} \phi.$$

The other basic fact is that isomorphic structures satisfy the same formulas of dependence logic.

Theorem 2.1.3. *[39] Suppose $\mathcal{M}$ and $\mathcal{N}$ are isomorphic and $f : \mathcal{M} \cong \mathcal{N}$. Then for all $\phi \in \mathcal{D}$ the following holds:*

$$\mathcal{M} \models_{\mathcal{X}} \phi \Leftrightarrow \mathcal{N} \models_{f[\mathcal{X}]} \phi.$$

Theorem 2.1.4. *[39](*Downwards closure*) Suppose $\phi \in \mathcal{D}$ and $\mathcal{X}$ and $\mathcal{Y}$ are teams, such that $\mathcal{Y} \subseteq \mathcal{X}$. Then the following holds:*

$$\mathcal{M} \models_{\mathcal{X}} \phi \Rightarrow \mathcal{M} \models_{\mathcal{Y}} \phi.$$

There is a correspondence between dependence logic formulas and existential second-order formulas, which bases on the result of Hodges [22].

Theorem 2.1.5. *[39] For every formula $\phi(x_1, \ldots, x_k) \in \mathcal{D}(\tau)$, there is a sentence $\theta(R) \in \Sigma_1^1(\tau \cup \{R\})$, where R is k-ary, such that for all τ-structures $\mathcal{M}$ and teams $\mathcal{X}$ with domain $\{x_1 \ldots, x_k\}$ the following are equivalent:*

1. $\mathcal{M} \models_{\mathcal{X}} \phi(x_1, \ldots, x_k)$.
2. $(\mathcal{M}, Rel(\mathcal{X})) \models \theta(R)$.

The converse also holds:

Theorem 2.1.6. *[31] For every formula $\phi \in \Sigma_1^1(\tau)$ there is a sentence $\phi^* \in \mathcal{D}(\tau)$, such that for all τ-structures $\mathcal{M}$ the following holds:*

$$\mathcal{M} \models \phi \Leftrightarrow \mathcal{M} \models \phi^*.$$

The translation from dependence logic to existential second-order logic yields some important model theoretical results, e.q. Compactness theorem and Löwenheim-Skolem theorem for dependence logic, see [39].

Chapter 3

Coherence

In this chapter we will investigate the notion of coherence in dependence logic. The intuition behind coherence is that a coherent formula can be verified locally in a team. A k-coherent formula ϕ is true in a team $\mathcal{X}$ if and only if it is true in every k-element sub-team of $\mathcal{X}$. Väänänen defined so-called *flatness* of a formula, which in terms of coherence is exactly the 1-coherence of a formula. It is shown in [39], that any formula of dependence logic which is equivalent to a first-order formula is flat, i.e. 1-coherent. An important point about coherence is that it allows us to come down from potentially very large teams into finite constant size sub-teams to evaluate the satisfiability of the formula. A well-know fact in finite model theory is that every finite structure (finite team) can be characterized in first-order logic up to isomorphism. For a fixed finite cardinality and finite vocabulary there are only finite number of different isomorphism types of structures. This yields that for every coherent $\phi \in \mathcal{D}$ there is an equivalent first-order sentence with an extra predicate interpreting the team.

The initial motivation to investigate coherence was it's implications on the computational complexity of the model checking of the formula. It seemed that every coherent formula could be verified in polynomial time, however, the FO-definability of coherent formulas yields a consequence that the computational complexity of the model checking of coherent formulas can be checked in logarithmic space. We will come back to this in the following chapter.

We will start by going through the atomic formulas, connectives and quantifiers of $\mathcal{D}$ and study the effect they have on coherence. We will observe that the satisfiability of first-order formulas can be done in restricting to singleton sub-teams whereas dependence atoms can be checked in restricting to two-element sub-teams. Furthermore, we will show that conjunction and universal quantification preserve coherence. We will observe that disjunction and existential quantifier are more complex than the other connectives when it comes to coherence. We will show that with both disjunction and existential quantifier one can fail coherence. We will give an example for both cases. We will also introduce a way to associate a multigraph to a team and a disjunction of dependence atoms in such a way that the satisfiability of the disjunction in the team corresponds exactly to the colorability of the multigraph.

Definition 3.0.7. Suppose $\phi(x_1 \ldots, x_n)$ is a quantifier-free $\mathcal{D}$-formula. Then ϕ is k-*coherent* if and only if for all structures $\mathcal{M}$ and teams $\mathcal{X}$ of range $dom(\mathcal{M})$, such that $Fr(\phi) \subseteq Dom(\mathcal{X})$ the following are equivalent:

1. $\mathcal{M} \models_{\mathcal{X}} \phi$.
2. For all k-element sub-teams $\mathcal{Y} \subseteq \mathcal{X}$ it holds that $\mathcal{M} \models_{\mathcal{Y}} \phi$.

Notice that the direction $1 \Rightarrow 2$ follows from downwards closure 2.1.4. Recall that the truth value of a sentence in a given structure depends only on the structure and not the interpretation: All sentences can be evaluated just with respect to the team with the empty assignment $\{\emptyset\}$. Since $\{\emptyset\}$ has only one proper sub-team, namely the empty team, all sentences are trivially coherent. Thus, the notion of coherence in its general form does not give any new information about sentences. It is convenient to assume the following normal form for formulas of dependence logic [31]. Note that we do not consider quantifier-free formulas to be in the normal form.

Theorem 3.0.8. *[31] Suppose $\phi \in \mathcal{D}$. Then ϕ is equivalent to a formula $\phi^* \in \mathcal{D}$ of the following form:*

$$\phi^* := \forall x_1 \ldots \forall x_k \exists x_{k+1} \ldots \exists x_m \theta(x_1, \ldots, x_m),$$

where θ a quantifier free formula[1].

We are not concerned on the form of the quantifier-free part of the formula in the last definition. We just need that the quantifiers are in front of the formula and that universal quantifiers precede the existential ones. This way we ensure that we are dealing with constant size sub-teams. We define coherence for formulas in the normal form in the following way:

Definition 3.0.9. Suppose $\phi := \forall x_1 \ldots \forall x_n \exists x_{n+1} \ldots \exists x_m \theta(x_1, \ldots, x_m) \in \mathcal{D}(\tau)$, where θ is quantifier-free $\mathcal{D}$-formula. We say that ϕ is k-*coherent* iff for all τ-structures $\mathcal{M}$ and teams $\mathcal{X}$, such that the $range(\mathcal{X}) = dom(\mathcal{M})$ and that $Fr(\phi) \subseteq \mathcal{X}$ the following are equivalent:

1. $\mathcal{M} \models_{\mathcal{X}} \forall x_1 \ldots \forall x_n \exists x_{n+1} \ldots \exists x_m \theta(x_1, \ldots, x_m)$.
2. $\mathcal{M} \models_{\mathcal{Y}} \exists x_{n+1} \ldots \exists x_m \theta(x_1, \ldots, x_m)$
   for all k-assignment subsets $\mathcal{Y} \subseteq \mathcal{X}((M, x_1)(M, x_2) \ldots (M, x_n))$.

Thus, we define coherence in terms of the teams where all the universally quantified variables are duplicated. In the case of sentences, coherence is defined with respect to a single team where all the universally quantified variables are duplicated.

[1] θ is as a conjunction of dependence atoms and a first-order formula

3.1 Connectives and coherence

We will first show the coherence of all atomic formulas and that conjunction and universal quantification preserve coherence. After this, we will show that disjunction and existential quantification do not preserve coherence.

Proposition 3.1.1. *First-order atomic formulas and negated atomic first-order formulas are* 1*-coherent.*

Proof. Let us recall the semantics of the atomic and negated atomic formulas.

- $\mathcal{M} \models_{\mathcal{X}} \approx t_1 t_2$, iff for every $s \in \mathcal{X}$, $s(t_1) = s(t_2)$.
- $\mathcal{M} \models_{\mathcal{X}} \neg \approx t_1 t_2$, iff for every $s \in \mathcal{X}$, $s(t_1) \neq s(t_2)$.
- $\mathcal{M} \models_{\mathcal{X}} R(t_1, \ldots, t_n)$, iff for every $s \in \mathcal{X}$, $(s(t_1), \ldots, s(t_n)) \in R^{\mathcal{M}}$.
- $\mathcal{M} \models_{\mathcal{X}} \neg R(t_1, \ldots, t_n)$, iff for every $s \in \mathcal{X}$, $(s(t_1), \ldots, s(t_n)) \notin R^{\mathcal{M}}$.

Clearly, in the case of all atomic and negated atomic formulas we have to check that each singleton subset $\{s\}$ of the team satisfies the formula. □

Proposition 3.1.2. *All dependence atoms are* 2*-coherent.*

Proof. Then semantics of the dependence atom was defined in the following way:

- $\mathcal{M} \models_{\mathcal{X}} =\!(t)$, iff for all $s, s' \in \mathcal{X}$ it holds $s(t) = s'(t)$.
- $\mathcal{M} \models_{\mathcal{X}} =\!(t_1, \ldots, t_n)$, $n > 1$, iff for all $s, s' \in \mathcal{X}$ it holds that, if $s(t_i) = s'(t_i)$ for $i \leq n-1$, then $s(t_n) = s'(t_n)$.

It is easy to see that it is enough to just check all the 2-element subsets $\{s, s'\}$ of the team when evaluating the satisfiability of the formula in a team. □

Next we will show that a conjunction of two coherent formulas stays coherent.

Proposition 3.1.3. *Suppose ϕ and ψ are quantifier-free formulas, such that ϕ is k-coherent and ψ is l-coherent and $l \leq k$. Then $\phi \wedge \psi$ is k-coherent.*

Proof. Suppose $\mathcal{X} \models \phi \wedge \psi$. Then, by downwards closure, all subsets of $\mathcal{X}$ satisfy $\phi \wedge \psi$, especially the k-element subsets.

The other direction: Suppose all k-element subsets $Y \subseteq \mathcal{X}$ satisfy $\phi \wedge \psi$. Then, by 2.0.10 it holds that $\mathcal{Y}$ satisfy ϕ and ψ for all $\mathcal{Y}$. Then, by the coherence of ϕ it follows that $\mathcal{X}$ satisfies ϕ. By downward closure and the fact that all l-element subsets of $\mathcal{X}$ are contained in some k-element subset, we conclude that all l-element subsets of $\mathcal{X}$ satisfy ψ. Then by coherence of ψ, also $\mathcal{X}$ satisfies ψ. Thus $\mathcal{X}$ satisfies $\phi \wedge \psi$. □

Also, the universal quantification preserves coherence.

Proposition 3.1.4. *Suppose ϕ is a k-coherent formula. Then $\forall x\phi$ is k-coherent.*

Proof. Notice that since ϕ is coherent it holds that $\forall x\phi$ is in the normal form. Suppose ϕ is the formula $\forall x_1 \ldots \forall x_m \exists y_1 \ldots \exists y_n \theta(\bar{x}, \bar{y}, \bar{z}, x)$. Since ϕ is k-coherent, the following holds for all teams $\mathcal{X}$ on domain $\{\bar{z}, x\}$:

$$\mathcal{X} \models \phi \Leftrightarrow \mathcal{X}((M, x_1) \ldots (M, x_n)) \models \exists y_1 \ldots \exists y_n \theta(\bar{x}, \bar{y}, \bar{z}, x).$$

Thus the equivalence above holds for all the teams of the form $\mathcal{X}(M, x)$, which is exactly the coherence condition for $\forall x\phi$. □

As we showed above, by universally quantifying over the variable x in ϕ we reduce the number of the teams with respect to we determine the coherence of ϕ. When x is universally quantified, we only consider teams where x is duplicated. When x is free, we consider all teams with x (and the other free variables of ϕ) in the domain. These teams naturally include the ones where x is duplicated.

Given a k-coherent formula we say that k is the coherence-level of the formula. Note that the lower the coherence level is the more "easy" it is to check the satisfiability of the formula. As we have established, combining atomic formulas with conjunction do not increase the coherence-level of the formula. When we look at the semantics of disjunction it is not clear if it preserves coherence.

- $\mathcal{M} \models_{\mathcal{X}} \phi \vee \theta$, iff there exists $\mathcal{Y}$ and $\mathcal{Z}$, such that $\mathcal{Y} \cup \mathcal{Z} = \mathcal{X}$, $\mathcal{M} \models_{\mathcal{Y}} \phi$ and $\mathcal{M} \models_{\mathcal{Z}} \theta$

Indeed, later we will show that even with one disjunction one can fail coherence. First we will consider the cases where disjunction preserves coherence.

Proposition 3.1.5. *Suppose ϕ and ψ are quantifier-free $\mathcal{D}$-formulas, such that ϕ is 1-coherent and ψ is k-coherent for some $k \in \mathbb{N}$. Then $\phi \vee \psi$ is k-coherent.*

Proof. Suppose it holds that $\mathcal{X} \models \phi \vee \psi$, then by downwards closure 2.1.4 $\mathcal{X}_k \models \phi \vee \psi$ holds for all k-element subsets $\mathcal{X}_k \subseteq \mathcal{X}$.

The other direction: Suppose that $\mathcal{X}_k \models \phi \vee \psi$ holds for all k-element subsets $\mathcal{X}_k \subseteq \mathcal{X}$. Now the division of $\mathcal{X}$ into $\mathcal{Y}$ and $\mathcal{Z}$, such that $\mathcal{Y} \models \phi$ and $\mathcal{Z} \models \psi$ is obtained in the following way:

- $s \in \mathcal{Y}$ iff $\{s\} \models \phi$ and $s \in \mathcal{Z}$ otherwise.

Clearly, it holds that $\mathcal{Y} \models \phi$. Let us show that $\mathcal{Z} \models \psi$. By k-coherence of ψ we have to check that for all k-element subsets $\mathcal{Z}_k \subseteq \mathcal{Z}$, it holds that $\mathcal{Z}_k \models \psi$. Suppose $\mathcal{Z}_k \subseteq \mathcal{Z}$, such that $\mathcal{Z}_k$ fails ψ. Since all the singletons $s \in \mathcal{Z}_k$ fail ϕ it holds that $\mathcal{Z}_k \not\models \phi \vee \psi$, which is a contradiction with the assumption. Thus all the k-element subsets of $\mathcal{Z}$ satisfy ψ. Which means by k-coherence of ψ that $\mathcal{Z} \models \psi$. Thus $\mathcal{Y} \cup \mathcal{Z} \models \phi \vee \psi$ holds. □

As established in the previous proposition, combining a k-coherent formula with a 1-coherent formula does not increase the coherence level. Thus we have to have both disjuncts at least 2-coherent. Namely we have to consider disjunctions over dependence atoms.

We denote the disjunction of size k over a single dependence atom $=(x_1,\dots,x_n)$, by $\bigvee_k =(x_1,\dots,x_n)$. We will next show that disjunctions over the same dependence atom increases the coherence-level, i.e. the coherence-level is increased by 1 for each disjunct.

Proposition 3.1.6. *$\bigvee_k =(x_1,\dots,x_n)$ is $k+1$-coherent for $k \in \mathbb{N}$.*

Proof. Suppose $\mathcal{X}$ is a team of type $\bigvee_k =(x_1,\dots,x_n)$. Then, by downwards closure property all $k+1$-element subsets of $\mathcal{X}$ satisfy $\bigvee_k =(x_1,\dots,x_n)$.

Other direction: Suppose all $k+1$-element subsets of $\mathcal{X}$ satisfy $\bigvee_k =(x_1,\dots,x_n)$. Let $S(a_1,\dots,a_{n-1})$ be defined in the following way for each $(a_1,\dots,a_{n-1}) \in M^{n-1}$:

$$S(a_1,\dots,a_{n-1}) = \{s \in \mathcal{X} \mid (s(x_1),\dots,s(x_{n-1})) = (a_1,\dots,a_{n-1})\}.$$

Let $|S(a_1,\dots,a_{n-1})|^*$ be the number of different values of x_n under the assignments in $S(a_1,\dots,a_{n-1})$. We will show that the following are equivalent:

1. $\mathcal{X} \models \bigvee_k =(x_1,\dots,x_n)$.
2. $|S(\bar{a})|^* \leq k+1$ for each $\bar{a} \in M^{n-1}$.

Suppose (2) holds. Then each $S(a_1,\dots,a_{n-1})$ can be divided into $k+1$ sets $S(a_1,\dots,a_{n-1})^i$, $1 \leq i \leq k+1$, such that x_n is constant in each $S(a_1,\dots,a_{n-1})^i$. Now the following partition of $\mathcal{X}$ into sets $\mathcal{X}_i$, $1 \leq i \leq k+1$, is what we are looking for:

$$\mathcal{X}_i = \bigcup_{\bar{a} \in M^{n-1}} S(a_1,\dots,a_{n-1})^i.$$

Next we will show that $\mathcal{X}_i \models =(x_1,\dots,x_n)$ for each $\mathcal{X}_i$, $1 \leq i \leq k+1$.

Suppose $s, s' \in \mathcal{X}_i$, such that s and s' are from the same $S(a_1,\dots,a_{n-1})^i$ for some $(a_1,\dots,a_{n-1}) \in M^{n-1}$. Now s and s' will agree on x_n since x_n is constant in each $S(a_1,\dots,a_{n-1})^i$. Thus $\{s,s'\} \models =(x_1,\dots,x_n)$ holds. Suppose s and s' are from different sets, say s from $S(a_1,\dots,a_{n-1})$ and s' from $S(a'_1,\dots,a'_{n-1})$. Then s and s' will disagree on the sequence $(x_1,\dots,x_{n-1})$. Thus $\{s,s'\} \models =(x_1,\dots,x_n)$ holds. Now $\mathcal{X}_i \models =(x_1,\dots,x_n)$ holds for each $\mathcal{X}_i$, $1 \leq i \leq k+1$ by 2-coherence of dependence atoms.

Other direction: Suppose (2) does not hold. Then, there exists $(a_1,\dots,a_{n-1}) \in M^n$, such that $|S(a_1,\dots,a_{n-1})| > k+1$. By *pigeon hole principle*[2] it is not possible to divide $S(a_1,\dots,a_n)$ into $k+1$ subsets $S(a_1,\dots,a_{n-1})^i$, so that in each

[2]Formally it states that there does not exist an injective function on finite sets whose codomain is smaller than its domain.

set x_n would be assigned a constant value. Since all the tuples in $S(a_1, \ldots, a_n)$ agree on sequence $x_1, \ldots, x_{n-1}$ it follows that $=(x_1, \ldots, x_n)$ will be failed in some subset independent of the division of $S(a_1, \ldots, a_n)$. Thus $\mathcal{X}$ does not satisfy $\bigvee_k =(x_1, \ldots, x_n)$.

The original assumption was that each $k+1$-element sub-team of $\mathcal{X}$ satisfies ϕ. Thus it holds that there are no such $k+1$-element subsets in $\mathcal{X}$ where the assignments agree on the first $n-1$ terms and all disagree on the last term. Thus for each tuple $(a_1, \ldots, a_{n-1}) \in M^{n-1}$ it holds that $|S(a_1, \ldots, a_{n-1})|^* \leq k+1$. Thus the claim follows with the above established equivalence.

$\square$

3.1.1 Incoherence

In this section we will show that disjunction and existential quantification do not preserve coherence. We will introduce a way to interpret a given team and disjunction of dependence atoms as a multigraph. Given a team $\mathcal{X}$ and a disjunction of dependence atoms $=(X_i, y_i)$, $i \in I$, denoted by $\bigvee_{i \in I} =(X_i, y_i)$, we interpret the team as a multigraph in such a way that the $|I|$-colorability of the multigraph corresponds to $\mathcal{X}$ satisfying $\bigvee_{i \in I} =(X_i, y_i)$. Each assignment translates into a vertex in the graph. Each dependence atom $=(X_i, y_i)$, induces edges between the vertices in such a way that if two assignments fail the dependence atom, then the corresponding vertices share the corresponding edge E_i.

Definition 3.1.7. Suppose $\mathcal{X} = \{s_1, \ldots, s_n\}$ is a team of domain V and range M and $\phi \in \mathcal{D}$ is of the form $\bigvee_{i \in I} =(X_i, y_i)$. For each $\mathcal{X}$ and ϕ we construct a multi graph $\mathcal{G}_X^\phi = (V, \{E_i | \, i \in I\})$ in the following way:

1. $V = \{v_j \mid s_j \in \mathcal{X}\}$.
2. For each $i \in I$, if $\{s_j, s_l\} \not\models =(X_i, y_i)$, then $(v_j, v_l) \in E_i$.

The k-colorability of a multigraph is defined as an existence of a coloring function $\sigma : dom(\mathcal{G}_X^\phi) \to |I|$, such that if two nodes share an edge E_i then they cannot be colored both with the same color i. The existence of such a coloring function on a graph matches exactly with the semantical condition of the disjunction in Team-semantics under the interpretation 3.1.7.

Proposition 3.1.8. *Suppose $\mathcal{G}_X^\phi$ is a multigraph defined as in 3.1.7 for a team $\mathcal{X}$ and formula $\phi =: \bigvee_{i \in I} =(X_i, y_i)$. Then the following two conditions are equivalent:*

1. *There exists a function $\sigma : V \to I$, such that if $\sigma(v_i) = \sigma(v_j) = m$, then $(v_i, v_j) \notin E_m$.*
2. *$\mathcal{X} \models \bigvee_{i \in I} =(X_i, y_i)$.*

Proof. Suppose $\sigma : V \to I$ is a function, such that if $\sigma(v_i) = \sigma(v_j) = m$, $m \in I$, then $(v_i, v_j) \notin E_m$. Let X_i, $1 \leq i \leq m$, be defined the following way:

$$X_i = \{s_n \mid s_n \in X \wedge \sigma(v_n) = i\}.$$

Since σ is defined on the domain $\mathcal{G}_X^{\phi}$, it holds that $\mathcal{X} = \bigcup_{i \in m} \mathcal{X}_i$. We will show next that $\mathcal{X}_i \models =\!(X_i, y_i)$ holds for each $\mathcal{X}_i$ $1 \leq i \leq m$:

Suppose $s_l, s_k \in \mathcal{X}_i$. Then, the corresponding vertices v_l and v_k are assigned the value i under σ. Then, by assumption on σ, it follows that $(v_l, v_k) \notin E_i$. Thus by 3.1.7, it follows that $\{s_l, s_k\} \models =\!(X_i, y_i)$. Further, it follows from the 2-coherence of the dependence atoms that $X_i \models =\!(X_i, y_i)$. Thus $\mathcal{X} \models \bigvee_{i \in I} =(X_i, y_i)$ holds.

The other direction: Suppose $\mathcal{X} \models \bigvee_{i \in I} =(X_i, y_i)$ holds. Then, there is a partition of $\mathcal{X}$ into sets $\mathcal{X}_i$, such that $\mathcal{X}_i \models =(X_i, y_i)$ for each $i \in I$, and $\mathcal{X} = \bigcup_{i \in I} \mathcal{X}_i$. Then, let σ be defined the following way:

- $\sigma(v_n) = i$, if $s_n \in \mathcal{X}_i$.

Clearly, σ is well defined and it holds that, if $\sigma(v_i) = \sigma(v_j) = m$, then $(v_i, v_j) \notin E_m$. □

Next lemma will show that disjunction does not preserve coherence. An important detail to notice is whether the disjuncts share some variables or not. As we will show later, disjunctions of the same dependence formula stay coherent.

Theorem 3.1.9. $=\!(x, y) \vee =\!(z, v)$ *is not k-coherent for any $k \in \mathbb{N}$.*

Proof. We will actually show that a stronger claim holds, namely that $=\!(x, y) \vee =\!(z, v)$ is not $f(n)$-coherent for any function $f : \mathbb{N} \to \mathbb{N}$, such that $f(n) < n$, for all n. Here the meaning of $f(n)$-coherence is, that a formula ϕ is $f(n)$-coherent, if for all teams $\mathcal{X}$, such that $|\mathcal{X}| = n$, it holds that $\mathcal{X} \models \phi \Leftrightarrow \mathcal{Y} \models \phi$ for every $\mathcal{Y} \subseteq \mathcal{X}$, such that $|\mathcal{Y}| = f(n)$.

We will construct a team $\mathcal{X}$ for every $k \in \mathbb{N}$ so that every proper subset of the team satisfies $=\!(x, y) \vee =\!(z, v)$, but the whole team fails to satisfy $=\!(x, y) \vee =\!(z, v)$. We represent the team as a multigraph as in 3.1.7. Each of the vertices correspond to an assignment of the team. Suppose $s_v, s_w \in \mathcal{X}$. There are two type of edges we assign between vertices in the following way.

- If $\{s_v, s_w\} \not\models =\!(x, y)$, then we assign a smooth edge between the vertices v and w.

- If $\{s_v, s_w\} \not\models =\!(z, v)$, then we assign a wavy edge between the corresponding vertices v and w.

We will use "black" color to denote the vertices that do not allow wavy edge and "white" color to denote the vertices that do not allow smooth edges. A coloring of the multigraph will be a partition of the universe into two sets, black and white vertices, such that the black vertices do not share any smooth edges and the white vertices do not share any wavy edges. The graph in figure 3.1 is such that every proper subgraph is 2-colorable, but the whole multigraph is not.

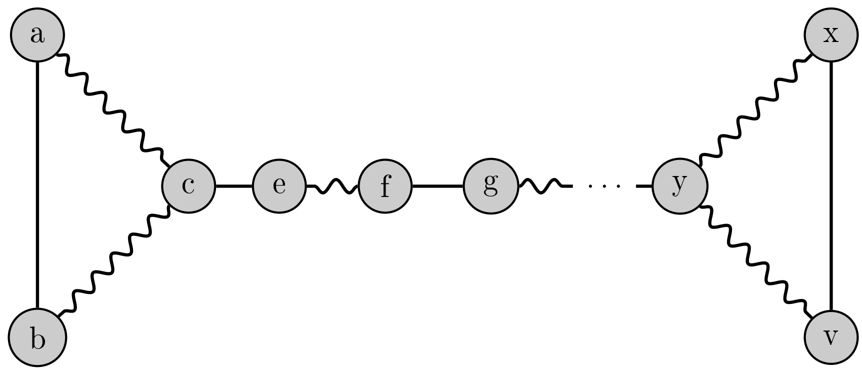

Figure 3.1: Graph $\mathcal{G}_{\mathcal{X}}$

$\mathcal{G}_{\mathcal{X}}$ is not 2-colorable: Suppose both the nodes c and y are colored black. Then vertices a, b, x, v should be colored white as they all share a wavy edge either with c or y. But since there is a smooth edge between a and b as well as between v and x and the fact that white color did not allow smooth edges, this cannot be a proper coloring. Thus the only way to properly color the triangles is to color both c and y white. The colors of a, b, x, v can be chosen black or white as long as v, x and a, b are not both white. The two triangles $\{a, b, c\}$ and $\{x, y, z\}$ are connected with a path (of even length). The path is such that the edge alternates between smooth and wavy, which forces the proper coloring also to alternate between black and white for the nodes on the path. Since the length of the path is even, there cannot be a coloring for the whole graph as the color of c totally determines the coloring of the whole path, in the same way as the color of y. They both force different colors on the path, thus making the proper coloring impossible. Thus the whole graph is not 2-colorable.

Every proper subgraph of $\mathcal{G}_{\mathcal{X}}$ is 2-colorable: We will show that if we remove a vertex from either of the triangles, then the coloring of the vertex c (or y), which is connected to the path, can be chosen either black or white. Suppose a is removed. Then we can choose so that c is colored black and b is with white. The vertex y has to be still colored with white. Now, since c and y are colored with different colors and the path connecting them is even, it holds that the whole graph can be colored. The cases where we remove any other vertex from the two triangles are analogous to this one.

On the other hand, suppose one of the vertices from the path connecting the

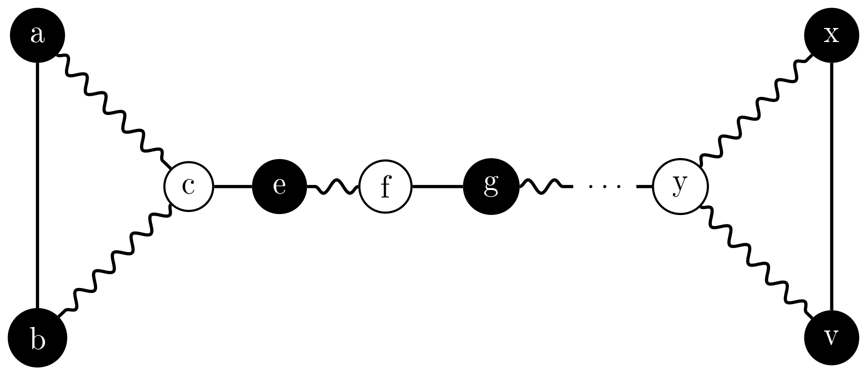

Figure 3.2: Coloring of the graph $\mathcal{G}_{\mathcal{X}}$

two triangles is removed. Then we have two components of the graph that are not connected by edges. The coloring of the whole graph reduces to the coloring of the two subgraphs for which there is a trivial coloring induced by the coloring of two nodes c and y. The team, which corresponds to the graph $\mathcal{G}_{\mathcal{X}}$ is the following table 3.1.

assignment	x	y	z	v
s_a	0	0	0	0
s_b	0	2	0	0
s_c	1	2	0	1
s_e	1	1	1	2
s_f	2	3	1	3
s_g	2	4	2	4
.				
.				
.				
s_v	4	7	3	5
s_x	4	6	3	5
s_y	3	5	3	4

Table 3.1:

As one can observe the values of the whole path are not explicitly given in the picture. If two vertices share a smooth edge, the corresponding assignments in the team in table 3.1 are assign the same value for x and different one for y. Similarly, if two vertices share an wavy edge the corresponding assignments assign the same value for z and different one for v. When we choose the values for the assignments that correspond to a vertex in the the path, we use always new values for the variables if possible. This way we ensure that there will be no unintended edges between the vertices in the triangle and the vertices in the path, just the ones that appear in the picture.

For example, for the assignment s_e the value of x is assigned the same as $s_c(x)$ and $s_e(y)$ is assigned different to $s_c(y)$, but $s_e(z)$ and $s_e(v)$ can be chosen new values. With the next vertex on the path, which is f, we can already assign the new values for x and y. We have to take care that the values of $s_f(z)$ and $s_f(v)$ are assigned so that the dependence $=(z, v)$ is failed. At this point of the path the values, which the assignment s_a, s_b and s_c assign to variables x, y, z, v are no more assigned when we go left in the path. Thus, the ranges of the variables under the assignments corresponding to the vertices of the triangles are disjoint with ranges of variables under the assignment that correspond to the vertices on the path (excluding the endpoints of the path).

Let us show that the team in table 3.1 indeed translates into the graph in figure 3.1. Recall that smooth edges are generated when two assignments fail the dependence $=(x, y)$ and wavy edges when the $=(z, v)$ is failed;

Smooth edges from vertex a: $s_a(x) = 0$. It holds that $s_b(x) = 0$ and since $s_b(y) \neq s_a(y)$ there is a smooth edge (a, b). All the other assignments assign other value than 0 for x, thus there cannot be smooth edges from a to other vertices.

Wavy edges from vertex a: $s_a(z) = s_c(z) = 0$ and $s_a(v) = 0 \neq 2 = s_z(v)$, thus there is a wavy edge (a, c). Indeed $s_b(z) = 0$ but $s_b(v) \neq s_a(v)$, thus there is no wavy edge (a, b). All the other assignments assign a value different to 0 for z, thus there are no wavy edges between a and other vertices.

Smooth edges from vertex b: The only smooth edge from b is the one that is shared with a. All other assignments assign a value different to 0 for x, thus there are no other smooth edges from b.

Wavy edges from vertex b: $s_b(z) = s_c(z) = 0$ and $s_b(v) = 0 \neq 2 = s_z(v)$, thus there is a wavy edge (b, c). Again, s_b and s_a agree on z but disagree on v, thus there is no wavy edge between them. All the other assignments assign a value different to 0 for z. Thus there are no other wavy edges from b.

Smooth edges from vertex c: Only assignment that assigns x as 1 is s_e. Also they disagree on y. Thus there is smooth edge (c, e). As we earlier noted, after the next vertex on the path, the values that are assigned by the assignments that correspond to nodes that appear in the triangle do not appear in the ranges of the assignments that correspond to the vertices that come later in the path. Thus all the other assignments assign a value different to 1 for x. Thus there are no other smooth edges from the node c.

Wavy edges from vertex c: The edges (c, a) and (c, b) have been already established. Again, the value that s_c assigns for z does not appear as a value for z under assignment corresponding the nodes which appear later in the path. Thus there are no other wavy edges from z.

The other triangle $\{x, y, z\}$ is isomorphic to that of $\{a, b, c, \}$. It can be checked analogous that exactly the edges that appear in the graph will be generated under the translation 3.1.7.

Now we have given a construction of collection of graphs like in 3.1,which are not 2-colorable, but for which hold that every proper sub-graph is. Now by 3.1.8 the following are equivalent:

1. $\mathcal{G}_{\mathcal{X}}$ is 2-colorable.

2. $\mathcal{X} \models =(x,y) \vee =(z,v)$.

Thus the whole team $\mathcal{X}$ as in table 3.1 does not satisfy $=(x,y) \vee =(z,v)$, but every proper sub-team of $\mathcal{X}$ satisfies $=(x,y) \vee =(z,v)$. By increasing the length of the path which connects the two triangles, we get the same counter example for different cardinalities. □

It is not obvious that there are teams that interpret the graph in figure 3.1 as there is no 1-1 correspondence between graphs and teams in terms of the translation defined in 3.1.7. For example, no cycle larger than 5 corresponds to any team.

Next we will show that existential quantification does not preserve coherence. We will show this by simulating disjunction with the existential quantifier.

Theorem 3.1.10. *The formula* $\exists x \exists y \exists z (\not\approx xy \wedge =(x) \wedge =(y) \wedge (\not\approx xz \vee =(u,v)) \wedge (\not\approx yz \vee =(w,t))$ *is not* k*-coherent for any* $k \in \mathbb{N}$.

Proof.

supplemented part of $\mathcal{X}$				original team $\mathcal{X}$			
assignment	x	y	z	v	u	w	t
s_0	$F_x(s_0)$	$F_y(s_0)$	$F_z(s_0)$	a_{i_0}	b_{i_0}	c_{i_0}	d_{i_0}
s_1	$F_x(s_1)$	$F_y(s_1)$	$F_z(s_1)$	a_{i_1}	b_{i_1}	c_{i_1}	d_{i_1}
s_2	$F_x(s_2)$	$F_y(s_2)$	$F_z(s_2)$	a_{i_2}	b_{i_2}	c_{i_2}	d_{i_2}
s_3	$F_x(s_3)$	$F_y(s_3)$	$F_z(s_3)$	a_{i_3}	b_{i_3}	c_{i_3}	d_{i_3}
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
s_m	$F_x(s_m)$	$F_y(s_m)$	$F_z(s_m)$	a_{i_m}	b_{i_m}	c_{i_m}	d_{i_m}

Table 3.2:

We will show that for all structures $\mathcal{M}$, $dom(\mathcal{M}) \geq 2$, and teams $\mathcal{X}$ the following are equivalent:

1. $\mathcal{M} \models_{\mathcal{X}} =(u,v) \vee =(w,t)$.

2. $\mathcal{M} \models_{\mathcal{X}} \exists x \exists y \exists z (\not\approx xy \wedge =(x) \wedge =(y) \wedge (\not\approx xz \vee =(u,v)) \wedge (\not\approx zy \vee =(w,t))$.

Suppose $\mathcal{M} \models_{\mathcal{X}} =(u,v) \vee =(w,t)$ holds. Then by 2.0.10, there are $\mathcal{Y}$ and $\mathcal{Z}$, such that $\mathcal{X} = \mathcal{Y} \cup \mathcal{Z}$ and $\mathcal{Z} \models =(u,v)$ and $\mathcal{Y} \models =(w,t)$. Let $a_1, a_2 \in M$ such that $a_1 \neq a_2$. We will define the choice functions for z, x and z in the following way:

- $F_x(s) = a_1$,
- $F_y(s) = a_2$,
- $F_z(s) = a_1$, if $s \in Y$,
- $F_z(s) = a_2$, if $s \in Z$.

Notice that the partition of $\mathcal{X} = \mathcal{Z} \cup \mathcal{Y}$ induces a partition on the supplemented team $\mathcal{X}(F_x, x)(F_y, y)(F_z, z)$. We denote this partition by $\mathcal{Z}' \cup \mathcal{Y}'$. One can observe that the following claims hold in the team $\mathcal{X}(F_x, x)(F_y, y)(F_z, z)$

1. $F_x(s) \neq F_y(s)$ for all s, thus $\mathcal{X}(F_x, x)(F_y, y)(F_z, z) \models \not\approx xy$
2. F_x is a constant function, thus $\mathcal{X}(F_x, x)(F_y, y)(F_z, z) \models =(x)$
3. F_y is a constant function, thus $\mathcal{X}(F_x, x)(F_y/y)(F_z, z) \models =(y)$

Now it holds that $\mathcal{Z}' \models \not\approx xz$ and $\mathcal{Y}' \models =(u,v)$, which implies that $\mathcal{Z}' \cup \mathcal{Y}' \models \not\approx xz \vee =(u,v)$ holds. Furthermore it holds that $\mathcal{Y}' \models \not\approx yz$ and $\mathcal{Z}' \models =(w,t)$, which implies that $\mathcal{Z}' \cup \mathcal{Y}' \models \not\approx yz \vee =(w,t)$ holds.

Thus it holds that

$$\mathcal{M} \models_{\mathcal{X}((F_x,x)(F_y,y)(F_z,z))} x \neq y \wedge =(x) \wedge =(y) \wedge (\not\approx zx \vee =(u,v)) \wedge (\not\approx yz \vee =(w,t)).$$

Which implies that the following hods:

$$\mathcal{M} \models_{\mathcal{X}} \exists x \exists y \exists z (\not\approx xy \wedge =(x) \wedge =(y) \wedge (\not\approx xz \vee =(u,v)) \wedge (z \neq y \vee =(w,t)).$$

The other direction: Suppose it holds that

$$\mathcal{M} \models_{\mathcal{X}} \exists x \exists y \exists z (\not\approx xy \wedge =(x) \wedge =(y) \wedge (\not\approx xz \vee =(u,v)) \wedge (\not\approx yz \vee =(w,t)).$$

Then there are choice functions for x, y and z denoted with F_x, F_y and F_z respectively, such that the following holds:

$$\mathcal{M} \models_{\mathcal{X}((F_x,x)(F_y,y)(F_z,z))} \not\approx xy \wedge =(x) \wedge =(y) \wedge (\not\approx xz \vee =(u,v)) \wedge (\not\approx yz \vee =(w,t)).$$

Thus it holds that:

1. $\not\approx xy \wedge =(x) \wedge =(y)$ implies that F_x and F_y are two different constant functions.
2. $\not\approx xz \vee =(u,v)$ implies that the set of assignments which agree on z and x satisfy $=(u,v)$.

3. $\not\approx yz \vee =(w,t)$ implies that the set of assignments that agree on z and y satisfy $=(w,t)$.

We will define the partition of $\mathcal{X}$ into two sets in the following way:

- $\mathcal{Z} = \{s \in \mathcal{X} \mid F_z(s) \neq F_x(s)\}$.
- $\mathcal{Y} = \{s \in \mathcal{X} \mid F_z(s) \neq F_y(s)\}$.

Clearly, now it holds that $\mathcal{Z} \models =(u,v)$ and $\mathcal{Y} \models =(w,t)$.

One can observe that the disjunctions $(\not\approx yz \vee =(w,t))$ and $(\not\approx xz \vee =(u,v))$ are 2-coherent by 3.1.5. Thus the whole quantifier-free formula $(\not\approx xy \wedge =(x) \wedge =(y) \wedge (\not\approx xz \vee =(u,v)) \wedge (\not\approx yz \vee =(w,t))$ is 2-coherent since conjunction preserves coherence by 3.1.3. The whole existentially quantified formula, however, is not coherent, since it is equivalent to $=(u,v) \vee =(w,t)$, which is incoherent by 3.1.9. □

Corollary 3.1.11. *The set of coherent formulas is not closed under existential quantification nor disjunction.*

Proof. By 3.1.9 and 3.1.10. □

As shown above, disjunctions of dependence atoms are not coherent in general.

3.2 Interpreting coherent formulas in terms of first-order sentences

In this subsection we will show that every coherent formula is equivalent to a first-order sentence when an additional relation symbol R is given interpreting the team $\mathcal{X}$.

Definition 3.2.1. Suppose $\mathcal{X}$ is a team of domain $\{x_1, \ldots, x_m\}$. Then let the relation $Rel(X)$ be the set of m-tuples defined in the following way:

$$Rel(\mathcal{X}) = \{(s(x_1), \ldots, s(x_m)) \mid \{s \in \mathcal{X}\}.$$

A well-know fact in finite model theory is that every finite structure can be characterized in first-order logic up to isomorphism. For a fixed finite cardinality and finite vocabulary there are only finite number of different isomorphism types of structures. Now the satisfiability of a k-coherent formula can be reduced to satisfiability in k-element sub-teams, which there are only finite many different ones (up to isomorphism). Thus we can just list the ones which satisfy ϕ and say that all the sub-teams should be one of these types.

Definition 3.2.2. Suppose τ is a finite vocabulary. *A k-τ-type* $t^k_\tau(x_1,\ldots,x_k)$ is a maximal consistent set of τ-atomic-, negated τ-atomic-, identity- and negated identity formulas over $\{x_1,\ldots,x_k\}$.

Suppose $\mathcal{M}$ is a τ-structure and $\bar{a}$ is a k-tuple $(a_1,\ldots,a_k)\in M^k$. We say that the tuple $(a_1,\ldots,a_k)$ realizes the k-τ-type $t^k_\tau(x_1,\ldots,x_k)$ in $\mathcal{M}$ if it holds

$$(M,\bar{a})\models\bigwedge_{\phi\in t^k_\tau}\phi.$$

We denote the k-τ-type realized by the tuple $\bar{a}$ in $\mathcal{M}$ by $t^{\mathcal{M}}_{\bar{a}}$.

Definition 3.2.3. Suppose $Rel(\mathcal{X})\subseteq M^m$ is a m-ary relation of size k. Let π be an ordering of the tuples of $Rel(\mathcal{X})$. Then $\bar{a}^\pi_x$ is the concatenation of the tuples in $Rel(\mathcal{X})$ in the order determined by π.

On the other hand, given a km-sequence $(a_1,\ldots,a_{km})$, $Rel(a_1,\ldots,a_{km})$ is the relation obtained by cutting the km-tuple into m-tuples

$$(a_1,\ldots,a_m),(a_{m+1},\ldots,a_{2m}),\ldots,(a_{m(k-1)+1},\ldots,a_{km}).$$

Given a team $\mathcal{X}$ with range M and domain V, let $Img(\mathcal{X})$ be the set of all elements $a\in M$, such that $s(v)=a$, for some $v\in V$ and for some $s\in\mathcal{X}$. Given a τ-structure $\mathcal{M}$ and a set $S\subseteq\mathrm{dom}(\mathcal{M})$ we let $\mathcal{M}\lceil S$ be the sub-structure with domain S endowed with relation $R^{\mathcal{M}}\cap S^n$ for each n-ary $R\in\tau$, $f^{\mathcal{M}}\cap S^n$ for each n-ary $f\in\tau$, and $c^{\mathcal{M}}$, such that $c^{\mathcal{M}}\in S$, for each $c\in\tau$.

Lemma 3.2.4. *Suppose τ is a finite vocabulary and $\mathcal{M}$ and $\mathcal{N}$ are τ-structures. Suppose $\mathcal{X}$ and $\mathcal{Y}$ are teams of k assignments with domain $\{x_1,\ldots,x_m\}$, such that $range(\mathcal{X})=M$ and $range(\mathcal{Y})=N$. Let $Img(\mathcal{X})=A$ and $Img(\mathcal{Y})=B$. Then the following conditions are equivalent:*

1. $(\mathcal{M}\lceil A,Rel(\mathcal{X}))\cong(\mathcal{N}\lceil B,Rel(\mathcal{Y}))$.
2. *There is an order π_1 of the tuples of $Rel(\mathcal{X})$ and an order π_2 of the tuples of $Rel(\mathcal{Y})$, such that $t^{M\lceil A}_{\bar{x}^{\pi_1}}=t^{N\lceil B}_{\bar{y}^{\pi_2}}$.*

Proof. Suppose $(\mathcal{M}\lceil A,Rel(\mathcal{X}))\cong(N\lceil B,Rel(\mathcal{Y}))$ holds and let f be the isomorphism between the two structures. Let $(a_1,\ldots,a_{km})$ be the sequence obtained by concatenating the tuples of $Rel(\mathcal{X})$ in the order π_1. Let $t_{\bar{x}}$ be the km-τ-type realized by $(a_1,\ldots,a_{km})$ in $\mathcal{M}$. Since f is an isomorphism, it holds that the image of the tuple $(a_1,\ldots,a_{km})$ under f, $(f(a_1),\ldots,f(a_{km}))$ realizes the same type $t_{\bar{x}}$ in $\mathcal{N}$ as $(a_1,\ldots,a_{km})$ in $\mathcal{M}$. Thus f induces an order π_2 over tuples in $Rel(\mathcal{Y})$, such that $t^{M\lceil A}_{\bar{x}^{\pi_1}}=t^{N\lceil B}_{\bar{y}^{\pi_2}}$. Thus the second condition is satisfied.

Suppose there is an order π_1 of the tuples of $Rel(\mathcal{X})$ and an order π_2 of the tuples of $Rel(\mathcal{Y})$, such that $t^{\mathcal{M}\lceil A}_{\bar{x}^{\pi_1}}=t^{\mathcal{N}\lceil Y}_{\bar{y}^{\pi_2}}$. Clearly, the concatenation of the tuples in the order of the π_1 and π_2 induce an order over the domains $M\lceil A$ and $N\lceil B$.

Now, let $f : M\lceil A \to N\lceil B$, $f(a_i^{\mathcal{M}\lceil\mathcal{A}}) = a_i^{\mathcal{N}\lceil\mathcal{B}}$, where $a_i^{\mathcal{M}}$ is the i:th element of $M\lceil A$ in the order induced by π_1 and respectively $a_i^{\mathcal{N}}$ of $N\lceil B$ in the order induced by π_2. Since the sequences $\bar{x}^{\pi_1}$ and $\bar{y}^{\pi_2}$ realize the same type, it holds that $f : (\mathcal{M}\lceil A, Rel(\mathcal{X})) \to (\mathcal{N}\lceil B, Rel(\mathcal{Y}))$ is an isomorphism. □

Theorem 3.2.5. *Suppose τ is a finite vocabulary, R an relation symbol interpreted as $Rel(\mathcal{X})$, and $\phi(\bar{z}) = \forall\bar{x}\exists\bar{y}\theta(\bar{x},\bar{y},\bar{z})$ is k-coherent $\mathcal{D}$-formula for some $k \in \mathbb{N}$. Then there is a sentence $\phi^* \in FO(\tau \cup \{R\})$, such that the following conditions are equivalent:*

1. $(M, Rel(\mathcal{X})) \models \phi^*$
2. $\mathcal{M} \models_{\mathcal{X}} \phi(\bar{x},\bar{y},\bar{z})$

Proof. Suppose $\mathcal{M}$ is a τ-structure and $\mathcal{X}$ a team with domain $\bar{z}$ and range M and that $\mathcal{M} \models_{\mathcal{X}} \phi$ holds. Then it holds that

$$\mathcal{M} \models_{\mathcal{X}(M,\bar{x})} \exists\bar{y}\theta(\bar{x},\bar{y},\bar{z}) \tag{3.1}$$

By k-coherence of ϕ, (3.1) is equivalent to:

$$\mathcal{M} \models_{\mathcal{Y}} \exists\bar{y}\theta(\bar{x},\bar{y},\bar{z}) \tag{3.2}$$

for all k-element subsets $\mathcal{Y} \subseteq \mathcal{X}(M,\bar{x})$. Further, (3.2) is equivalent to the fact that every $\mathcal{Y}$ can be supplemented to satisfy $\theta(\bar{x},\bar{y},\bar{z})$. We denote the supplemented team by $\mathcal{Y}(F_{\bar{y}},\bar{y})$. Each $\mathcal{Y}(F_{\bar{y}},\bar{y})$ is a set of k distinct assignments. Note that the domain of the team is finite and of fixed size. Thus, each $\mathcal{Y}(F_{\bar{y}},\bar{y})$ can be characterized in FO up to isomorphism.

Let us denote the number of variables in the formula ϕ with $\#\phi$. Since τ is finite, there are only finitely many different $(\#\phi \cdot k)$-τ-types. Now, let T^{ϕ} be the set of all $(\#\phi \cdot k)$-τ-types t, such that there is a team $\mathcal{X}$ for which $\mathcal{X} \models \theta$ holds and that the sequence $(a_1, \ldots, a_{km})$, obtained by concatenating the tuples of $Rel(\mathcal{X})$ in some order, realizes the type t. Now ϕ^* is the following sentence:

$$\phi^* =: \forall\bar{s}_1 \ldots \forall\bar{s}_k((\bigwedge_{i\in k} \bar{s}_i \in R \bigwedge_{i\neq j} \bar{s}_i \neq \bar{s}_j) \to$$
$$\forall\bar{x}_1 \ldots \forall\bar{x}_k \ldots \exists\bar{y}_1 \ldots \exists\bar{y}_k \bigvee_{t_\tau \in T^{\phi}} t_\tau(\bar{s}_1, \bar{x}_1, \bar{y}_1, \ldots, \bar{s}_k, \bar{x}_k, \bar{y}_k,)),$$

where $\forall\bar{s}_i$ is a shorthand for $\forall z_{i_1} \ldots \forall z_{i_m}$, $\forall\bar{x}_i$ for $\forall x_{i_1} \ldots \forall x_{i_n}$ and $\exists\bar{y}_i$ for $\exists y_{i_1} \ldots \exists y_{i_l}$, where n is the number of universally quantified variables and l the number of existentially quantified variables.

Let us show that that the following conditions are equivalent.

1. $(M, Rel(\mathcal{X})) \models \phi^*$.

2. $\mathcal{M} \models_{\mathcal{X}} \phi(\bar{x}, \bar{y}, \bar{z})$.

Suppose it holds that $(M, Rel(\mathcal{X})) \models \phi^*$ and that $\mathcal{X}_k = \{s_1, \ldots s_k\} \subseteq \mathcal{X}(M, \bar{x})$ is a k-element team. Let $\bar{a}_i = (s_i(z_1), \ldots, s_i(z_m), s_i(x_1), \ldots, s_i(x_n))$ for each assignment $s_i \in \mathcal{X}_k$. It follows that for every k-element sub-team $\{s_1, \ldots s_k\} \subseteq \mathcal{X}(M, \bar{x})$ the corresponding $(m+n)$-tuples $\bar{a}_1, \ldots, \bar{a}_k$ satisfy:

$$\mathcal{M} \models \exists \bar{y}_1 \ldots \exists \bar{y}_k \bigvee_{t_\tau \in T^\phi} t_\tau(\bar{a}_1, \bar{y}_1, \ldots, \bar{a}_k, \bar{y}_k,).$$

Furthermore, there exists tuples $\bar{a}'_1, \ldots, \bar{a}'_k \in M^l$, such that

$$\mathcal{M} \models \bigvee_{t_\tau \in T^\phi} t_\tau(\bar{a}_1, \bar{a}'_1, \ldots, \bar{a}_k, \bar{a}'_k).$$

We denote the function $s_i \mapsto \bar{a}'_i$, $i \leq k$, by $F_{\bar{y}}$. Since the concatenation of the tuples $(\bar{a}_1, \bar{a}'_1), \ldots, (\bar{a}_k, \bar{a}'_k)$ realizes a type t in T^ϕ, it holds by 3.2.4 that $\mathcal{X}_k(F_{\bar{y}}, \bar{y})$ is isomorphic to a team which satisfies $\theta(\bar{x}, \bar{y}, \bar{z})$. Thus it holds $\mathcal{X}_k(F_{\bar{y}}, \bar{y}) \models \theta(\bar{x}, \bar{y}, \bar{z})$. Since this holds for every k-element sub-team $\mathcal{X}_k \subseteq \mathcal{X}(M, \bar{x})$, it follows that the coherence condition for the $\mathcal{D}$-formula $\phi(\bar{x}, \bar{y}, \bar{z})$ is satisfied. Thus $\mathcal{M} \models_{\mathcal{X}} \phi(\bar{x}, \bar{y}, \bar{z})$ holds.

Other direction. Suppose $\mathcal{M} \models_{\mathcal{X}} \phi(\bar{x}, \bar{y}, \bar{z})$ holds. Then by k-coherence of ϕ it follows that

$$\mathcal{M} \models_{\mathcal{X}(M,\bar{x})_k} \exists \bar{y} \theta(\bar{x}, \bar{y}, \bar{z})$$

for every k-element sub-team $\mathcal{X}(M, \bar{x})_k$ of the duplicated team $\mathcal{X}(M, \bar{x})$.

Suppose $\{\bar{a}_1, \ldots, \bar{a}_k\}$ are k distinct m-tuples of $Rel(\mathcal{X})$. Let us denote the extension of tuple $\bar{a}_i$ with n-tuple $\bar{b}_i \in M^n$ with $\bar{a}'_i$, i.e. $\bar{a}'_i = (\bar{a}_i, \bar{b}_i)$. Now each such set of k distinct $(m+n)$-tuples $\bar{a}'_1, \ldots, \bar{a}'_k$ correspond to a k-element sub-team of $\mathcal{X}(M, \bar{x})_k$. Since $\mathcal{M} \models_{\mathcal{X}} \phi(\bar{x}, \bar{y}, \bar{y})$ holds, there exists a choice function $F_{\bar{y}}$ for the variables $\bar{y}$, such that $\mathcal{X}(M, \bar{x})_k(F_{\bar{y}}, \bar{y}) \models \theta(\bar{x}, \bar{y}, \bar{z})$. It follows that the concatenation of the tuples $(\bar{a}'_1, F_{\bar{y}}(s), \ldots \bar{a}'_1, F_{\bar{y}}(s))$ realizes a type in T^ϕ. Since $F_{\bar{y}}$ exists for each k-element sub-team $\mathcal{X}(M/\bar{x})_k$ it holds that

$$(\mathcal{M}, Rel(\mathcal{X})) \models \forall \bar{s}_1 \ldots \forall \bar{s}_k ((\bigwedge_{i \in k} \bar{s}_i \in R \bigwedge_{i \neq j} \bar{s}_i \neq \bar{s}_j) \rightarrow$$

$$\forall \bar{x}_1 \ldots \forall \bar{x}_k \ldots \exists \bar{y}_1 \ldots \exists \bar{y}_k (\bigvee_{t_\tau \in T^\phi} t_\tau(\bar{s}_1, \bar{x}_1, \bar{y}_1, \ldots, \bar{s}_k, \bar{x}_k, \bar{y}_k,))).$$

$\square$

Chapter 4

Descriptive complexity

Descriptive complexity theory is a field of *finite model theory* which studies the connection between logical definability and computational complexity. Two central problems considered in this field are the *model checking* and the *satisfiability testing*. The *satisfiability testing* for a logic L and a class of finite structures $\mathbf{M}$ is the following problem: Given a sentence $\phi \in L$, determine whether or not there is a structure in $\mathbf{M}$ which satisfies ϕ. The *model checking problem* for a logic L and a set of finite structures $\mathbf{M}$ is to determine whether $\mathcal{M} \models \phi$ holds or not for a given formula $\phi \in L$ and a finite structure $\mathcal{M} \in \mathbf{M}$. Closely related problem to the model checking problem is the *query evaluation problem*: Given a formula $\phi(x_1, \ldots x_k)$ and a structure $\mathcal{M}$, it is to calculate the relation defined by the formula $\phi(x_1, \ldots, x_k)$, i.e. the set of tuples $\bar{a} \in M^k$ for which hold $(M, \bar{a}) \models \phi(x_1, \ldots, x_k)$. The query evaluation reduces to a polynomially many model checking problems.

The complexities of these problems are measured as a function of the size of the input. The taxonomy for measuring the computational complexity of different query languages was developed by Vardi in [37]. One usually differentiates between three complexities: *Data complexity*; The formula is fixed and the structure is given as an input. *Expression complexity*; The structure is fixed and the formula is given as a input. *Combined complexity*; Both the structure and the formula are given as input. Usually, combined complexity is exponentially higher than the data complexity. The computational complexity of the query evaluation for various logics is presented in the table 4.1 [38].

Definition 4.0.6. Suppose $\mathbf{M}$ is a class of structures and Φ is a class formulas. The *model checking problem for* $\mathbf{M}$ *and* Φ, denoted with $MC(\mathbf{M}, \Phi)$, is to determine whether $\mathcal{M} \models \phi$ holds for given $\mathcal{M} \in \mathbf{M}$ and $\phi \in \Phi$.

When we fix the formula ϕ and the class of structures $\mathbf{M}$ is clear from the context, we denote the model checking MC($\mathbf{M}, \{\phi\}$) with MC(ϕ).

We will consider both data- and combined complexity of the model checking problem of $\mathcal{D}$-formulas over finite teams. We will give a characterization of the

logic	data complexity	expression complexity	combined complexity
FO	AC^0	PSPACE-complete	PSPACE-complete
FP	PTIME-complete	EXPTIME-complete	EXPTIME-complete
Σ_1^1	NP-complete	NEXPTIME-complete	NEXPTIME-complete
PFP	PSPACE-complete	EXPSPACE-complete	EXPSPACE-complete

Table 4.1: Data- and combined complexity for different logics.

data complexity for quantifier-free formulas in terms of coherence and number of conjunctions in the formula. We will observe that disjunction increases the computational complexity of the model checking for dependence logic formulas. In case of classical disjunction, the effect is linear, as you just have to check if one of the disjuncts is satisfied. In dependence logic, however, we have a weaker form of disjunction and it is more complex than the classical one. We will show that there is a leap in computational complexity between disjunctions of size one and two when considering the combined complexity of the model checking problem for dependence atoms.

4.1 Computational complexity theory

Computational complexity theory focuses on classifying problems according to their inherent difficulty. One measures the difficulty of a given problem by the amount of resources it take to solve it, such as the running *time*, or the memory *space* used by the algorithm. The complexity is measured as a function of the size of the input. A problem is regarded as inherently difficult if all the algorithms solving the problem require a large amount of resources. The following notation is commonly adopted:

Definition 4.1.1. Suppose f and g are functions from $\mathbb{N}$ to $\mathbb{N}$. We write

$$f \in \mathcal{O}(g(n)),$$

if there are positive constants c and k, such that $0 \leq f(n) \leq cg(n)$ for all $n > k$. We write $f \in o(g(n))$ if

$$\lim_{n \to \infty} \frac{f(n)}{g(n)} = 0.$$

Definition 4.1.2. A function is called *time-constructible*, if there exists a Turing machine M which, given a string consisting of n ones, stops after exactly $f(n)$ steps. Analogously, f is *space-constructible*, if there is a Turing machine M that stops after using exactly $f(n)$ cells.

Notice that the definitions for time-and space constructible functions may slightly vary depending on the source. Most of the commonly used functions such as polynomial-functions are both time- and space-constructible.

We will denote the class of languages recognized by a deterministic Turing machine working in time $f(n)$, where n is the length of the input by TIME($f(n)$) and languages recognized by a deterministic Turing machine using at most $f(n)$ memory cells, where n is the length of the input by SPACE($f(n)$). And respectively NTIME($f(n)$) and NSPACE($f(n)$) for the non-deterministic versions of these classes.

Definition 4.1.3.

$$\begin{aligned}
LOGSPACE &= \bigcup_{k\in\mathbb{N}} SPACE(k\cdot\log(n)),\\
NL &= \bigcup_{k\in\mathbb{N}} NSPACE(k\cdot\log(n)),\\
PTIME &= \bigcup_{k\in\mathbb{N}} TIME(n^k),\\
NP &= \bigcup_{k\in\mathbb{N}} NTIME(n^k),\\
PSPACE &= \bigcup_{k\in\mathbb{N}} SPACE(n^k),\\
EXPTIME &= \bigcup_{k\in\mathbb{N}} TIME(2^{n^k}),\\
NEXPTIME &= \bigcup_{k\in\mathbb{N}} NTIME(2^{n^k}),\\
EXPSPACE &= \bigcup_{k\in\mathbb{N}} SPACE(2^{n^k}).
\end{aligned}$$

It is known that

$$\begin{aligned}
&LOGSPACE \subseteq NL \subseteq PTIME \subseteq NP \subseteq PSPACE \subseteq\\
&\quad EXPTIME \subseteq NEXPTIME \subseteq EXPSPACE.
\end{aligned}$$

The strictness of this hierarchy is open for most of the cases. The strictness of some of the containments follow from the Hierarchy theorems for time and space see [33] and [13]:

Theorem 4.1.4. *[18](Deterministic Time hierarchy theorem) Suppose $f(n)$ is a time-constructible function. Then $TIME(f(n)) \subset TIME(f^2(n))$.*

Theorem 4.1.5. *[6](Nondeterministic Time Hierarchy Theorem) Suppose $g(n)$ is a time-constructible function, and $f(n+1) = o(g(n))$.*
Then NTIME($f(n)$) $\subsetneq$NTIME($g(n)$).

Theorem 4.1.6. *(Space Hierarchy Theorem) Suppose $f(n)$ is a space-constructible function, such that $f(n) > \log n$ and f(n)= o(g(n)). Then $SPACE(f(n)) \subsetneq SPACE(g(n))$.*

Savitch's theorem establishes a relation between deterministic and non-deterministic space.

Theorem 4.1.7. *[34](Savitch's theorem) Suppose $f(n)$ is a function, such that $f(n) > \log n$. Then $NSPACE(f(n)) \subseteq SPACE(f^2(n))$.*

The Hierarchy theorems for time and space and Savitch's Theorem give the following known inequalities:

Corollary 4.1.8.

$$\begin{aligned} NL &\subsetneq PSPACE, \\ PSPACE &\subsetneq EXPSPACE, \\ PTIME &\subsetneq EXPTIME, \\ NP &\subsetneq NEXPTIME. \end{aligned}$$

The Hierarchy theorems for time and space give us the last three inequalities. Savitch's Theorem shows that NL $\subseteq SPACE(log2n)$, while the Space Hierarchy Theorem shows that SPACE($\log 2n$) $\subsetneq$ SPACE(n). Thus giving us that NL $\subsetneq$PSPACE.

Measuring computational complexity using Turing machines as computational model one has to encode finite structures into strings. For this, one has to assume some ordering of the structure. The encoding in it self is not relevant as long as certain conditions are met. For example, the size of the encoding should be at most polynomial to the size of the structure.

We denote all the finite τ-structures with $\mathbf{M}^\tau$. Let us fix an encoding Bin of finite τ-structures into binary words. We denote the encoding of a structure $\mathcal{M}$ with $Bin(\mathcal{M})$.

Definition 4.1.9. Let $\phi \in L(\tau)$-formula. Then let

$$\mathcal{K}(\phi) = \{\mathcal{M} \mid \mathcal{M} \in \mathbf{M}^\tau \wedge \mathcal{M} \models \phi\}.$$

Definition 4.1.10. Suppose $\mathcal{K}$ is a class of τ-structures. Then $\mathcal{L}(\mathcal{K})$ is the language defined by $\mathcal{K}$ under the encoding Bin;

$$\mathcal{L}(\mathcal{K}) = \{Bin(\mathcal{M}) \mid \mathcal{M} \in \mathcal{K}\}.$$

Also each language can be seen as a class of structures in the following way.

Definition 4.1.11. Suppose $\mathcal{L}$ is a language over some finite vocabulary τ. A word $w = v_0 \dots v_n \in \mathcal{L}$ can be seen as a structure $\mathcal{W}(w)$ over vocabulary $\leq \cup \{P_v \mid v \in \Sigma\}$ with universe $\{v_0, \dots, v_n\}$ and with the natural interpretation for $\leq$ and $P_a = \{i \mid v_i = a\}$. Now let $\mathcal{W}(\mathcal{L})$ be defined the following way for each language $\mathcal{L}$:

$$\mathcal{W}(\mathcal{L}) = \{\mathcal{W}(w) \mid w \in \mathcal{L}\}.$$

Thus, mathematical structures can be encoded into languages which can be then given as inputs for Turing machines. On the other hand, languages can be turned into classes of mathematical structures, which can be then characterized using logic. When there is a perfect match between a definability in logic L and computability in a complexity class C, we say that the logic L characterizes the complexity class C:

Definition 4.1.12. Let L be a logic and C a complexity class. We say that the logic L characterizes a complexity class C, if for any vocabulary τ and any class of τ-structures $\mathcal{K}$ the following two conditions are equivalent:

1. $\mathcal{K}$ is definable in L,
2. $\mathcal{L}(\mathcal{K}) \in C$.

We denote it by $L \equiv \mathcal{C}$ and $\equiv_{\leq}$ for equivalence over ordered structures.

Fagin initiated the field of descriptive complexity theory in [9]. He showed that Σ_1^1 characterizes the non-deterministic polynomial time. Another classical result is by Immerman and Vardi, that polynomial time is characterized by first order logic extended by a least fixed point operator (FO(LFP)) over ordered structures [23], [37]. It is considered an important open problem to find a characterizing logic for PTIME over all finite structures.

Theorem 4.1.13. *([9])* Fagin's theorem:

$$\Sigma_1^1 \equiv NP.$$

Theorem 4.1.14. *([23], [37])* Immerman-Vardi theorem:

$$FO(LFP) \equiv_{\leq} PTIME.$$

Many of the main classes are given logical characterizations, e.q. first-order logic extended by deterministic transitive closure operator ($FO(DTC)$) characterizes LOGSPACE over ordered structures and first-order logic extended by the transitive closure operator ($FO(TC)$) characterizes NL over ordered structures. Second-order logic extended with second order transitive closure operator ($SO(TC^2)$) characterizes PSPACE over all finite structures.

A problem $\mathcal{L}$ is said to be complete for a class C if every other problem in C can be reduced to $\mathcal{L}$. The theory of complete problems was initiated by Cook in

[7]. He showed that determining the satisfiability of a given boolean first order formula is complete for NP. Soon after, Karp showed the NP-completeness of 21 famous combinatoric- and graph-theoretic problems in [25]. For more NP-complete problems see [12].

Definition 4.1.15. A function $f : A \to B$ is *C-computable* if there is a Turing machine M working in C, which for each input $a \in A$ returns the value $f(a)$.

Definition 4.1.16. Suppose $\mathcal{L}_1 \subseteq A$ and $\mathcal{L}_2 \subseteq B$. $\mathcal{L}_1$ is *C-reducible* to $\mathcal{L}_2$, if there is a C-computable function $f : A \to B$, such that for all $\alpha \in A$

$$\alpha \in \mathcal{L}_1 \iff f(\alpha) \in \mathcal{L}_2.$$

We denote this by $\mathcal{L}_1 \leq_C \mathcal{L}_2$.

Definition 4.1.17. Let $\mathcal{L}$ be a language, $\leq_r$ is a reducibility relation and C is a complexity class. We say that $\mathcal{L}$ is *complete for C with respect to r-reductions* if the following two conditions are met:

1. $\mathcal{L} \in C$,
2. for all $\mathcal{L}' \in C$, $\mathcal{L}' \leq_r \mathcal{L}$.

NP-complete problems are usually considered with respect to polynomial time reductions.

Definition 4.1.18. *Boolean satisfiability problem* (SAT) is a decision problem to determine wether a given propositional first order formula is satisfiable. The variables are boolean and may occur positively or negatively in the formula. The formulas are assumed to be in the conjunctive normal form. The problem is to determine, whether there is an assignment, which evaluates the given formula true. There are several variations of SAT from which we consider the following two:

- 2-SAT: At most 2 disjuncts in each clause.
- 3-SAT: At most 3 disjuncts in each clause.

2-SAT is shown to be complete for NL(for proof see) [33] and 3-SAT complete for NP [25]. Another well-know NP-complete problem is the 3-colorability of a graph [12].

Definition 4.1.19. *k-colorability of a graph* (k-COL) is a decision problem to determine wether the vertices of a given graph can be colored with k colors in such a way that, if two vertices share an edge, then they are colored with different colors. In other words, if there is a function $\xi : V \to \{0, \dots 1, k-1\}$, such that if $(v, w) \in E$, then $\xi(v) \neq \xi(w)$.

We will use reductions to 2-SAT, 3-SAT and k-COL to show completeness of certain model checking problems of $\mathcal{D}$-formulas.

4.2 Computational complexity of dependence logic formulas

We are interested in finding a connection between the complexity of the model checking problem and the syntactic form of ϕ. The model checking for all $\mathcal{D}$-formulas is in NP. We will show that coherent formulas can be verified in logarithmic space.

Recall, that the formulas of dependence logic are verified in terms of sets of assignments. A formula $\phi \in \mathcal{D}$ defines a collection of pairs $(\mathcal{M}, \mathcal{X})$, where $\mathcal{M}$ is a τ-structure and $\mathcal{X}$ is a team with range M, such that $Fr(\phi) = dom(\mathcal{X})$. Thus every formula of dependence logic can be seen as a collection of $\tau \cup \{R\}$-structures, where R is a $|Fr(\phi)|$-ary relation symbol interpreting the team $\mathcal{X}$. When we consider sentences, the team is always the same team containing just the empty assignment, $\{\emptyset\}$, and it can be naturally left out, but for quantifier-free formulas, which we are mainly focused in, we have to take the team into consideration. When given a structure $\mathcal{M}$ and a team $\mathcal{X}$, the pair $(\mathcal{M}, \mathcal{X})$ is coded as a structure $(\mathcal{M}, Rel(\mathcal{X}))$.

Theorem 4.2.1. $\mathcal{D} \equiv NP$.

Proof. Suppose $\phi \in \mathcal{D}(\tau)$ defines the class $\mathcal{K}(\phi)$ of pairs $(\mathcal{M}, \mathcal{X})$, where $\mathcal{M}$ is a finite τ-structure and $\mathcal{X}$ is a team with range M and domain $Fr(\phi)$. By Theorem 2.1.5 $\mathcal{K}(\phi)$ is definable in Σ_1^1 as a collection of $\tau \cup \{R\}$-structures $\mathcal{K}(\phi^*)$, where R is interpreted as $Rel(\mathcal{X})$.

Furthermore by Theorem 4.1.13, it holds that $((\mathcal{M}, Rel(\mathcal{X}))) \in \mathcal{K}(\phi^*)$ can be decided in NP. Thus $(\mathcal{M}, \mathcal{X}) \in \mathcal{K}(\phi)$ can be decided in NP.

Other direction: Suppose $\mathcal{K} \in NP$. Then by Theorem 4.1.13 $\mathcal{K}$ is definable in Σ_1^1 . By Theorem 2.1.6, $\mathcal{K}$ is also definable in $\mathcal{D}$. □

There are several related results on the computational complexity of fragments of Σ_1^1 and for partially ordered quantifiers: Gottlob, Kolaitis and Schwentick characterize Σ_1^1-formulas with respect to their quantifier prefixes over directed, undirected and undirected graphs with self-loops in [14]. Grädel considers certain fragments of SO, which collapse to their existential fragments, which in the presence of a successor relation provide characterizations for LOGSPACE, NL and PTIME [17].

Blass and Gurevich observe the connection between NP-computability and definability with Henkin quantifiers [2]. When the prefixes of the FO formulas are linearly ordered, it is just the classical quantification of FO and the formula can be verified in LOGSPACE. They show that all non-linear Henkin quantifiers can express NP-complete problems as long as the existentially quantified variables range over the whole universe. They impose constraints on the existentially quantified variables and show that

$$\begin{pmatrix} \forall x_{11} & \ldots & \forall x_{1k} & \exists y_1 \\ \forall x_{21} & \ldots & \forall x_{2k} & \exists y_2 \end{pmatrix} \tag{4.1}$$

can express NP-complete problems as long as the variables y_1 and y_2 range over at least a three element set. When y_1 and y_2 are boolean variables they can be verified in NL. Furthermore, they show that the following two quantifiers are enough to express NP-complete problems:

$$\begin{pmatrix} \forall x_1 & \exists \alpha_1 \\ \forall x_2 & \exists \alpha_2 \\ \forall x_3 & \exists \alpha_3 \end{pmatrix} \tag{4.2}$$

$$\begin{pmatrix} \forall x_1 & \exists \gamma_1 \\ \forall x_2 & \exists \gamma_2 \end{pmatrix} \tag{4.3}$$

where, α_i, $i \leq 3$, are boolean and γ_i, $i \leq 2$ range over three element domain.

One method for obtaining results for fragments of dependence logic would have been to map the fragments of dependence logic into fragments of Σ_1^1 or into fragments of first-order logic defined with Henkin quantifiers, for which the computational complexity is known. We did not find this approach fruitful since at least the straightforward translations of formulas did not seem to give any non-trivial results.

It is known that the data complexity of the model checking of first order formulas can be done in LOGSPACE(see [16] for proof).

Theorem 4.2.2. *Suppose $\phi \in FO(\tau)$. Then $MC(\phi) \in$LOGSPACE.*

Theorem 4.2.3. *Suppose $\phi \in \mathcal{D}(\tau)$ is a k-coherent formula for some $k \in \mathbb{N}$. Then $MC(\phi) \in$LOGSPACE.*

Proof. Suppose $\phi \in \mathcal{D}(\tau)$ is a k-coherent formula and that it defines a class $\mathcal{K}(\phi)$ of pairs $(\mathcal{M}, \mathcal{X})$, where $\mathcal{M}$ is a τ-structure and $\mathcal{X}$ is a team of range M and domain $Fr(\phi)$. Then by 3.2.5 there is FO($\tau \cup \{R\}$)-sentence ϕ^*, where R is $|Fr(\phi)|$-ary relation symbol interpreted as $Rel(\mathcal{X})$, such that

$$(\mathcal{M}, \mathcal{X}) \in \mathcal{K}(\phi) \Leftrightarrow (\mathcal{M}, Rel(\mathcal{X})) \in \mathcal{K}(\phi^*) \tag{4.4}$$

By 4.2.2, it holds that $(\mathcal{M}, Rel(\mathcal{X})) \in \mathcal{K}(\phi^*)$ can be decided LOGSPACE. Thus $(\mathcal{M}, \mathcal{X}) \in \mathcal{K}(\phi)$ can be decided in LOGSPACE. □

4.3 Quantifier-free formulas

We will give characterization for the data complexity of quantifier-free formulas in terms of numbers of disjunctions in the formula and coherence. We will point out three thresholds, namely when the complexity of the model checking is in LOGSPACE, NL or in NP. We will also give complete instances for NL and NP.

Suppose $\phi(x_1, \ldots, x_k) \in \mathcal{D}$ is a quantifier free formula. We will show that the following claims hold:

1. If ϕ is coherent, then $MC(\phi) \in LOGSPACE$ by Theorem 4.2.3.

2. If $\phi =: \theta \vee \psi$, where θ and ψ are 2-coherent formulas , then $MC(\phi) \in NL$ and contains NL-complete instances.

3. Formulas of form $\theta \vee \psi$, where θ is 2-coherent and ψ a 3-coherent formula contain NP-complete instances.

First, we will show that the model checking problem for disjunctions of 2-coherent formulas can be reduced to 2-satisfiability problem.

Theorem 4.3.1. *Suppose ϕ and ψ are 2-coherent $\mathcal{D}$-formulas. Then*

$$MC(\phi \vee \psi) \leq_{LOGSPACE} 2 - SAT.$$

Proof. Suppose we are given a team $\mathcal{X} = \{s_1, \ldots, x_k\}$. We will go through all the two-element subsets $\{s_i, s_j\} \subseteq \mathcal{X}$, and construct an instance of 2-SAT in the following way:

- If $\{s_i, s_j\} \not\models \phi$, then $(x_i \vee x_j) \in C$.
- If $\{s_i, s_j\} \not\models \psi$, then $(\neg x_i \vee \neg x_j) \in C$.

We let $\Theta_X = \bigwedge_{\phi \in C} \phi$. Clearly, Θ_X is a proper instance of 2-SAT. We will next show that there is an assignment S, which satisfies Θ_X if and only if $\mathcal{X} \models \phi \vee \psi$ holds: Suppose there is an assignment $S : Var(\Theta_X) \to \{0, 1\}$, which evaluates Θ_X true. Let us define the partition of $\mathcal{X}$ in the following way:

- $\mathcal{Z} = \{s_i \in \mathcal{X} \mid S(x_i) = 1\}$.
- $\mathcal{Y} = \mathcal{X} \setminus \mathcal{Z}$.

Clearly it holds that $\mathcal{X} = \mathcal{Z} \cup \mathcal{Y}$. Let us show that $\mathcal{Z} \models \psi$ and $\mathcal{Y} \models \phi$ hold:

Suppose $s_i, s_j \in \mathcal{Z}$. Since S satisfies Θ_X, $(\neg x_i \vee \neg x_j)$ cannot be a clause in $C_{\mathcal{X}}$. By the construction above, it follows that $\{s_i, s_j\} \models \psi$ holds. Now, by 2-coherence of ψ it follows that $\mathcal{Z} \models \psi$.

Suppose $s_i, s_j \in \mathcal{Y}$. Since S was assumed to satisfy $C_{\mathcal{X}}$, $(x_i \vee x_j)$ cannot be a clause in $C_{\mathcal{X}}$. It follows by the construction above that $\{s_i, s_j\} \models \phi$ holds. Again, from 2-coherence of ϕ it follows that $\mathcal{Y} \models \phi$ holds.

The other direction: Suppose $\mathcal{X} \models \phi \vee \psi$ holds. Then, by Definition 2.0.10 it holds that there is a division of $\mathcal{X}$ into two sets $\mathcal{Z}$ and $\mathcal{Y}$, such that $\mathcal{X} = \mathcal{Z} \cup \mathcal{Y}$, $\mathcal{Z} \cap \mathcal{Y} = \emptyset$, $\mathcal{Y} \models \phi$ and $\mathcal{Z} \models \psi$. Let S be defined the following way:

- $S(x_i) = 1$, if $s_i \in \mathcal{Z}$.
- $S(x_i) = 0$, if $s_i \in \mathcal{Y}$.

Clearly it holds that $S : Var(\Theta_X) \to \{0,1\}$ is a function. Let us show that S satisfies Θ_X: Suppose $\theta \in C$ of form $(x_i \vee x_j)$. Then $\{s_i, s_j\}$ fails ϕ by the construction of $\Theta_{\mathcal{X}}$. Then s_i and s_j cannot be both in $\mathcal{Y}$, since $\mathcal{Y}$ was supposed to satisfy ϕ. Thus, either s_i or s_j must be in $\mathcal{Z}$. Then, it holds that $S(x_i) = 1$ or $S(x_j) = 1$, which implies that $S(x_i \vee x_j) = 1$.

Suppose θ is $(\neg x_i \vee \neg x_j)$. Then, by the construction of $\Theta_{\mathcal{X}}$, it holds that $\{s_i, s_j\}$ fails ψ. Then, s_i and s_j cannot be both in $\mathcal{Z}$, since $\mathcal{Z}$ was supposed to satisfy ψ. Thus either s_i or s_j must be in $\mathcal{Y}$. Then, it holds that $S(x_i) = 0$ or $S(x_j) = 0$, which implies that $S(\neg x_i \vee \neg x_j) = 1$.

Last, the complexity of this reduction is in LOGSPACE: We go trough the 2-element subsets of the team $\mathcal{X}$ and check if they fail ϕ or ψ. Since ϕ and ψ were coherent, the model checking for the sub-teams can be done in LOGSPACE by Theorem 4.2.3. □

Corollary 4.3.2. *Suppose ϕ and ψ are 2-coherent $\mathcal{D}$-formulas. Then*

$$MC(\phi \vee \psi) \in NL.$$

Next we will show that the set of formulas of form $\phi \vee \psi$, where ϕ and ψ are 2-coherent contain NL-complete instances.

4.3.1 A complete instance for non-deterministic logarithmic space

We will reduce the problem 2-SAT to the model checking problem of the formula $=\!(x,y) \vee =\!(z,v)$.

Theorem 4.3.3. $2-SAT \leq_{LOGSPACE} MC(=\!(x,y) \vee =\!(z,v))$.

Proof. Suppose $\theta(p_0, \ldots, p_{m-1})$ is an instance of 2-SAT of the form $\bigwedge_{i \in I} E_i$, where each conjunct $E_i = (A_{i_1} \vee A_{i_2})$, $i \in I$, where A_{i_j}, $j \leq 1$, are positive or negative boolean variables.

We will construct a team $\mathcal{X}$, such that the following are equivalent:

1. $\mathcal{X} \models =\!(x,y) \vee =\!(z,v)$.
2. $\theta(p_0, \ldots, p_{m-1})$ is satisfiable.

For each conjunct E_i, $i \in I$, we create a team $\mathcal{X}_{E_i}$ where we code the information required to satisfy E_i. Now, E_i will be satisfied if one of the disjuncts will be true. Thus it has two conditions for being satisfied. We will code these conditions into the team we construct in the following way:

We will have a variable z denote the clause E_i, x denote the variables of the clause, y the truth value of the corresponding variable and v that makes sure we choose at least one of the assignments form each $\mathcal{X}_{E_i}$ into the sub-set of $\mathcal{X}$ which eventually codes the assignment which evaluates θ true. Each disjunct A_{ij} gives a rise to one assignment. Now $\mathcal{X}$ is the union $\bigcup_{i \in I} \mathcal{X}_{Ei}$.

For example, the team $\mathcal{X}_{E_i}$ for a clause $(p_k \vee p_j)$ is the one in Table 4.2. The team for the whole instance of 2−SAT:

$$(A_{01} \vee A_{02}) \wedge (A_{11} \vee A_{12}) \wedge \ldots \wedge (A_{I_1} \vee A_{I_2})$$

is the one in Table 4.3, where $t(A_i) = 1$ if A_i is a positive variable and $t(A_i) = 0$, if A_i is a negated variable.

z	x	y	v
i	p_k	1	1
i	p_j	1	2

Table 4.2: Team for $(p_k \vee p_j)$.

z	x	y	v
0	A_{01}	$t(A_{01})$	1
0	A_{02}	$t(A_{02})$	2
1	A_{11}	$t(A_{11})$	1
1	A_{12}	$t(A_{02})$	2
2	A_{21}	$t(A_{21})$	1
2	A_{22}	$t(A_{22})$	2
.	.	.	.
.	.	.	.
.	.	.	.
n	A_{I1}	$t(A_{I1})$	1
n	A_{I2}	$t(A_{I2})$	2

Table 4.3: Team $\bigcup_{i \in I} \mathcal{X}_{Ei}$.

Suppose $\theta(p_0, \ldots, p_{m-1})$ is satisfiable. Then there exists an assignment $F : \{p_0, \ldots, p_{m-1}\} \to \{0, 1\}$, such that F evaluates $\theta(p_0, \ldots, p_{m-1})$ true. We define the partition of the team $\mathcal{X}$ into two sets in the following way: For each $s \in \mathcal{X}$, $s \in \mathcal{X}_1$ if the following condition holds:

$$(s(x) = p_i) \to F(p_i) = s(y). \tag{4.5}$$

Otherwise $s \in \mathcal{X}_2$.

Condition (4.5) guarantees that the tuples that agree with the assignment F are chosen to $\mathcal{X}_1$. Since F evaluates $\bigwedge_{i\in I} E_i$ to true, it evaluates every conjunct E_i true. As the satisfying conditions of each E_i are coded into $\mathcal{X}_{E_i}$, the condition (4.5) is satisfied by at least one of the assignments in each $\mathcal{X}_{E_i}$. Thus there will be at most one tuple from each $\mathcal{X}_{E_i}$ in $\mathcal{X}_2$. Thus $\mathcal{X}_2$ trivially satisfies $=(z,v)$ since all tuples in $\mathcal{X}_2$ disagree on z. Next we will show that $\mathcal{X}_1$ satisfies $=(x,y)$: Let $s,s' \in \mathcal{X}_1$, such that $s(x) = s'(x) = p_i$. Then by (4.5) it follows that $s(y) = F(p_i) = s'(y)$ holds. Thus $\mathcal{X}_1 \models\ =(x,y)$.

The other direction: Suppose $\mathcal{X} \models =(x,y)\vee =(z,v)$. Then there is a partition of $\mathcal{X}$ into $\mathcal{X}_1$ and $\mathcal{X}_2$, such that $\mathcal{X}_1 \models =(x,y)$ and $\mathcal{X}_2 \models =(z,v)$. We will define the assignment $F : \{p_0,\ldots,p_m\} \to \{0,1\}$ in the following way:

- If $\exists s \in \mathcal{X}_1$, such that $s(x) = p_i$, then $F(p_i) = s(y)$.
- If $\forall s \in \mathcal{X}_1$ it holds $s(x) \neq p_i$, then $F(p_i) = 1$.[1]

Let us show that $F : \{p_0,\ldots,p_{m-1}\} \to \{0,1\}$ is a function, which evaluates $\theta(p_0,\ldots,p_{m-1})$ true:

1. Clearly, $Dom(F) = \{p_0,\ldots,p_{m-1}\}$ and $Range(F) = \{0,1\}$.
2. F is a function: Let $p_i \in \{p_0,\ldots,p_m\}$. Suppose there exists $s,s' \in \mathcal{X}_1$, such that $s(x) = s'(x) = p_i$ holds. Since $\mathcal{X}_1 \models =(x,y)$ holds, it follows that $s(y) = s'(y)$ holds. Suppose there are no $s \in \mathcal{X}_1$, such that $s(x) = p_i$. Then by definition of F it holds that $F(p_i) = 1$.
3. F evaluates Θ_X true: Note that z is constant and v is assigned different value by each tuple in each $\mathcal{X}_{E_i}$. Thus $\mathcal{X}_1$ contains at least one of the tuples from each $\mathcal{X}_{E_i}$. Let $s_0 \in \mathcal{X}_{E_i}$, such that $s_0 \in \mathcal{X}_1$. Since each tuple codes a satisfying condition of E_i it follows that F evaluates one of the disjuncts in E_i true. Thus $S(E_i) = 1$.

Each conjunct of θ gives rise to a constant size team of two assignments with domain $\{x,y,z,v\}$. Thus the team $\mathcal{X}$ can be constructed in LOGSPACE when given θ.

□

The problem 2−SAT is complete for NL [33]. Thus we have the following corollary.

Corollary 4.3.4. *$MC(=(x,y)\vee =(z,v)$ is complete for NL.*

[1] If for all the assignments $s \in X_1$ holds $s(x) \neq p_i$, then the value of p_i is not relevant to the satisfiability of Θ. Thus the value of p_i can be chosen 0 or 1.

4.3.2 A complete instance for non-deterministic polynomial time

We will show that the set of formulas of form $\phi \vee \psi$, where ϕ is 2-coherent and ψ is 3-coherent formula contains NP-complete instances. We will show that 3-SAT can be reduced to the model checking problem of the formula $=(x,y) \vee =(z,u) \vee = (z,v)$.

Recall that an instance $\theta \in$ 3-SAT is a first-order formula in conjunctive normal form, where each conjunct has at most three variables: $\bigwedge_{i \in I} E_i$, where I is finite. Each E_i is of form $(A_{i_0} \vee A_{i_2} \vee A_{i_3})$, where A_i is either a positive or a negated boolean variable. θ is accepted if there is an assignment, which evaluates θ true. The reduction is analogous with the reduction given in Theorem 4.3.3.

Theorem 4.3.5. $3 - SAT \leq_{LOGSPACE} MC(=(x,y) \vee =(z,v) \vee =(z,v))$.

Proof. Suppose $\theta(p_0, \ldots, p_{m-1})$ is an instance of 3-SAT with conjuncts E_i, $i \in I$. We will construct a team $\mathcal{X}$, such that the following are equivalent:

- $\mathcal{X} \models =(x,y) \vee =(z,v) \vee =(z,v)$.
- $\theta(p_0, \ldots, p_{m-1})$ is satisfiable.

For each conjunct E_i, $i \in I$, we create a team $\mathcal{X}_{E_i}$ where we code all the satisfying conditions of the clause E_i. Let $\mathcal{X} = \bigcup_{i \in I} \mathcal{X}_{E_i}$. For example, a clause $E_i = (p_l \vee \neg p_j \vee \neg p_k)$ will be satisfied if $p_l = 1$ or $p_j = 0$ or $p_k = 0$. The team for $(p_l \vee \neg p_j \vee \neg p_k)$ is the one in Table 4.4.

z	x	y	v
1	p_l	1	0
1	p_j	0	1
1	p_k	0	2

Table 4.4: A team for $(p_l \vee \neg p_j \vee \neg p_k)$.

Suppose $\theta(p_0, \ldots, p_{m-1})$ is satisfiable. Then there exists an assignment $F : \{p_0, \ldots, p_{m-1}\} \to \{0,1\}$, such that F evaluates $\theta(p_0, \ldots, p_{m-1})$ true. We define $\mathcal{X}_1$ in the following way: For all $s \in \mathcal{X}$, $s \in \mathcal{X}_1$ if

$$s(x) = p_) \to F(p_i) = s(y) \tag{4.6}$$

Since F evaluates $\bigwedge_{i \in n} E_i$ true, it evaluates every conjunct E_i true. Furthermore, since we coded all the satisfying conditions of E_i into $\mathcal{X}_{E_i}$, it holds that at least one assignment from each $\mathcal{X}_{E_i}$ satisfies the condition (4.6). Thus $\mathcal{X}_1$ contains at least one assignment from each $\mathcal{X}_{E_i}$. Thus the two "leftover"-assignment form each $\mathcal{X}_{E_i}$ can be easily divided into $\mathcal{X}_2$ and $\mathcal{X}_3$ in such a way that $=(z,v)$ holds in both of them. We just place one of the assignments into $\mathcal{X}_2$ and one into $\mathcal{X}_3$.

Let us show that $\mathcal{X}_1 \models =(x,y)$: Suppose $s, s' \in \mathcal{X}_1$, such that $s(x) = s'(x) = p_i$. Then, by (4.6), it follows that $s(y) = s'(y) = F(p_i)$. Thus $\mathcal{X}_1 \models =(x,y)$.

The other direction: Suppose $\mathcal{X} \models =(x,y) \vee =(z,v) \vee =(z,v)$ holds. Then by the truth definition of the disjunction, it follows that $\mathcal{X}$ can be partitioned into three sets $\mathcal{X}_1$, $\mathcal{X}_2$ and $\mathcal{X}_3$, such that $\mathcal{X}_1 \models =(x,y)$, $\mathcal{X}_2 \models =(z,v)$ and $\mathcal{X}_3 \models =(z,v)$ hold. Let F be defined in the following way for each variable p_i.

- If $\exists s \in \mathcal{X}_1$, such that $s(x) = p_i$, then $F(p_i) = s(y)$.
- If $\forall s \in \mathcal{X}_1$ it holds $s(x) \neq p_i$, then $F(p_i) = 1$.

Let us show that $F : \{p_0, \dots, p_m\} \to \{0,1\}$ is a function, which evaluates $\theta(p_0, \dots, p_{m-1})$ true:

1. Clearly, F is well defined and the domain of F is $\{p_0, \dots, p_{m-1}\}$ and the range is $\{0,1\}$.
2. F is a function: Let $p_i \in \{p_0, \dots, p_m\}$. Suppose there exists $s, s' \in \mathcal{X}_1$, such that $s(x) = s'(x) = p_i$ holds. Since $\mathcal{X}_1 \models =(x,y)$ holds, it follows that $s(y) = s'(y) = F(p_i)$ holds. If there exists no $s \in \mathcal{X}_1$, such that $s(x) = p_i$, then it holds by the definition of F, that $F(p_i) = 1$.
3. We will show that S evaluates each E_i, $i \in I$, true: Note that z is constant and v gets different value by each tuple in each $\mathcal{X}_{E_i}$. Thus $\mathcal{X}_1$ must contain at least one of the tuples from each X_{E_i}. Since each tuple in $\mathcal{X}_{E_i}$ codes a satisfying condition for E_i it means that F agrees with one of the satisfying conditions for E_i. Thus F satisfies E_i.

Each conjunct of theta gives rise to a constant size team of three assignments with domain $\{x,y,z,v\}$. Thus given θ, $\mathcal{X}$ can be constructed in LOGSPACE. □

By Theorem 4.2.1 it holds that $MC(=(x,y) \vee =(z,v) \vee =(z,v)) \in$NP. It is well-know that 3-SAT is complete for NP [25]. Thus we have the following corollary.

Corollary 4.3.6. *$MC(=(x,y) \vee =(z,v) \vee =(z,v))$ is complete for NP.*

4.4 Combined complexity of dependence logic formulas over finite teams

We will study the combined complexity of classes of 2-coherent formulas over all finite teams and the effect of disjunction on these classes. We will show that when we allow disjunction over two formulas the combined complexity of the model checking for these classes is in NL and that it becomes NP-complete when we allow disjunctions over three formulas. When we measure the combined complexity for a given set of formulas and set of structures, both the structure and the formula are coded as input.

Definition 4.4.1. Suppose $\mathbf{M}$ is a class of finite structures and Φ is a class of formulas. Let $D_i(\mathbf{M}, \Phi)$ be the decision problem with input $(\mathcal{X}, \phi_0, \ldots, \phi_{i-1})$, where $\mathcal{X} \in \mathbf{M}$ and $\phi_j \in \Phi$ for all $j < i$. Problem is to determine whether $\mathcal{X} \models \bigvee_{j \in i} \phi_j$.

Theorem 4.4.2. *Suppose C is a complexity class, such that $LOGSPACE \subseteq C$, $\mathbf{M}$ is a set of finite teams closed under sub-teams and Φ is a class of 2-coherent formulas, such that $MC(\mathbf{M}, \Phi) \in C$. Then*

$$D_2(\mathbf{M}, \Phi) \leq_C 2SAT.$$

Proof. Suppose $\mathcal{X}$ is a team, ϕ_0 and $\phi_1 \in \Phi$. Let Cl be the set of clauses defined the following way: For each 2-element subset $\{s_i, s_j\} \subseteq \mathcal{X}$;

- If $\{s_i, s_j\} \not\models \phi_0$ holds, then $(x_i \vee x_j) \in Cl$.
- If $\{s_i, s_j\} \not\models \phi_1$ holds, then $(\neg x_i \vee \neg x_j) \in Cl$.

Let $C_{\mathcal{X}} = \bigwedge_{\phi \in Cl} \phi$. Clearly, $C_{\mathcal{X}}$ is a proper instance of $2SAT$. Next we will show that $\mathcal{X} \models \phi_0 \vee \phi_1$ holds if and only if $C_{\mathcal{X}}$ is satisfiable.

Suppose there is an assignment $S : Var(C_{\mathcal{X}}) \to \{0, 1\}$, which evaluates $C_{\mathcal{X}}$ true. Let $\mathcal{X}_0$ and $\mathcal{X}_1$ be defined in the following way:

$$\mathcal{X}_0 = \{s_j \mid S(x_j) = 0\},$$

$$\mathcal{X}_1 = \mathcal{X} \setminus \mathcal{X}_0.$$

Clearly, it holds that $\mathcal{X} = \mathcal{X}_0 \cup \mathcal{X}_1$. Suppose s_i, $s_j \in \mathcal{X}_0$. Since S was assumed to satisfy $C_{\mathcal{X}}$, it holds that $(x_i \vee x_j)$ cannot be a clause in $C_{\mathcal{X}}$. Thus $\{s_i, s_j\} \models \phi_0$ holds by the construction of $C_{\mathcal{X}}$. By 2-coherence of ϕ_0, it follows that $\mathcal{X}_0 \models \phi_0$ holds. Suppose s_i, $s_j \in \mathcal{X}_1$. Again, since S was assumed to satisfy $C_{\mathcal{X}}$, it holds that $(\neg x_i \vee \neg x_j)$ cannot be a clause in $C_{\mathcal{X}}$. Thus $\{s_i, s_j\} \models \phi_1$ holds by the construction of $C_{\mathcal{X}}$. Again, by 2-coherence of ϕ_1, it follows that $\mathcal{X}_1 \models \phi_1$ holds. Thus $\mathcal{X} \models \phi_0 \vee \phi_1$.

The other direction: Suppose $\mathcal{X} \models \phi_0 \vee \phi_1$. Then by Definition 2.0.10, there are $\mathcal{X}_0$ and $\mathcal{X}_1$, such that $\mathcal{X} = \mathcal{X}_0 \cup \mathcal{X}_1$ and $\mathcal{X}_i \models \phi_i$, for $i \in \{0, 1\}$. Let $S : Var(C_{\mathcal{X}}) \to \{0, 1\}$ be defined in the following way:

- $S(x_i) = 0$, if $s_i \in \mathcal{X}_0$.
- $S(x_i) = 1$, if $s_i \in \mathcal{X}_1$.

Clearly, S is well defined. Let us show that S satisfies $C_{\mathcal{X}}$. Suppose $(x_i \vee x_j)$ is a clause in $C_{\mathcal{X}}$. Then $\{s_i, s_j\}$ fails ϕ_0 by the construction of $C_{\mathcal{X}}$. Thus s_i and s_j cannot be both in $\mathcal{X}_0$. Thus x_i or x_j must be in $\mathcal{X}_1$. Thus $S(x_i) = 1$ or $S(x_j) = 1$. In both cases, it holds that $S(x_i \vee x_j) = 1$.

Suppose $(\neg x_i \vee \neg x_j)$ is a clause in $C_{\mathcal{X}}$. Then by the construction of $C_{\mathcal{X}}$, it follows that $\{s_i, s_j\}$ fails ϕ_1. Then s_i and s_j cannot be both in $\mathcal{X}_1$. Thus x_i or x_j must be in $\mathcal{X}_0$. Thus $S(x_i) = 0$ or $S(x_j) = 0$. In both cases, it holds that $S(\neg x_i \vee \neg x_j) = 1$.

Last, the complexity of the reduction is in C: We go trough the 2-element subsets of the team $\mathcal{X}$, which can be done in LOGSPACE. For each subset $\{s, s'\}$ we check if they fail ϕ_0 or ϕ_1. Since we assumed that $\mathbf{M}$ was closed under sub-teams, especially the 2-element sub-teams and that $MC(\mathbf{M}, \Phi) \in$C, $\{s, s'\} \models \phi_i$ can be decide in C for each 2-element sub-team and for each ϕ_i, $i \leq 2$. □

The following corollary states that if the combined complexity of the model checking for a pair $MC(\mathbf{M}, \Phi)$ is already as high as NL, then allowing one disjunction over the formulas of Φ does not increase the combined complexity of the model checking.

Corollary 4.4.3. *Suppose* $\mathbf{M}$ *is a set of teams closed under sub-teams,* Φ *is a class of* 2*-coherent formulas,* C *is a complexity class, such that* $NL \subseteq C$ *and* $MC(\mathbf{M}, \Phi) \in C$*. Then* $D_2(\mathbf{M}, \Phi) \in C$*.*

4.4.1 Dependence atoms

We will consider the combined complexity of the model checking for dependence atoms over finite teams. We will show that the combined complexity of disjunctions of conjunctions of dependence atoms becomes NP-complete for disjunctions larger than 2. Note, that we consider disjunctions of a single dependence atom. Thus all the formulas considered in this section are coherent. It is essential to the reduction that we have unbounded number of variables in use.

Definition 4.4.4. Let M_n^k be the set of all teams with domain $\{x_1, \ldots, x_k\}$ and range $\{1, \ldots, n\}$. Let

$$\mathcal{M} = \bigcup_{k,n \in N} M_n^k.$$

Let $\mathcal{T}$ be the set of all finite conjunctions of dependence atoms over variables $\{x_i \mid i \in N\}$.

Theorem 4.4.5. *$MC(\mathcal{M}, \mathcal{T}) \in LOGSPACE$.*

Proof. Suppose we are given a team $\mathcal{X}$ and some finite conjunction of dependence atoms $\bigwedge_{i \in m} =\!(X_i, y_i)$. We use $log(n)$ many memory space as a counter to go through all the conjuncts of the formula. We will check for each $i \in m$ whether

$\mathcal{X} \models =(X_i, y_i)$. If this holds, we increase the counter to $i+1$ and check again $\mathcal{X} \models =(X_{i+1}, y_{i+1})$. We proceed this way until some of the conjuncts fail or the counter reaches m. If the counter reaches m, then all the conjuncts are satisfied and the machine accepts the input. If some conjunct is failed, the machine halts, and does not accept the input.

Since the size of the conjunction m is always smaller than the input n (the formula is part of the input), $log(n)$ memory space is enough to go through all the conjuncts. The model checking for a single dependence atom can be checked in LOGSPACE by Theorem 4.2.3. The algorithm checking for a dependence in a relation is the same for all dependence atoms, we just check from different places (for different variables) with different dependence atoms. Thus, $MC(\mathcal{M}, \mathcal{T})$ is in LOGSPACE. □

Next we restrict the problem $D_k(\mathcal{M}, \mathcal{T})$ so that we only consider disjunction over a single formula. We denote this problem with $D^*_k(\mathcal{M}, \mathcal{T})$. Thus the the problem is to decide if $\mathcal{X} \models \phi \vee \phi$ for given $\mathcal{X} \in \mathcal{M}$ and $\phi \in T$. We will show that $k - COL$ can be reduced to $D^*_k(\mathcal{M}, \mathcal{T})$

Theorem 4.4.6. $k - COL \leq_{LOGSPACE} D^*_k(\mathcal{M}, \mathcal{T})$.

Proof. Given a graph $\mathcal{G} = (V, E)$ we construct a team $\mathcal{X}_G$ and a conjunction of dependence atoms ϕ in the following way:

- For each $v \in V$ we add an assignment $s_v \in \mathcal{X}_G$.
- Suppose there is some order of the vertices of E. For the n:th pair $(v_i, v_j) \in E$ we add a new conjunct $=(x_{i_n}, x_{j_n})$ to ϕ, and extend all the assignments in $\mathcal{X}_G$ with variables x_{i_n} and x_{j_n}. We let $s_v(x_{i_n}) = s_w(x_{i_n})$ and $s_v(x_{j_n}) \neq s_w(x_{j_n})$, thus we assign values in a way that $\{s_v, s_w\} \not\models =(x_{i_n}, x_{j_n})$. The values for $s_v(x)$ and $s_w(x)$ for all other variables x are set to 0.
- Every time we process an edge (v, w) of the relation E, we use new variables and assign new values (excluding 0), which have not been used before in the construction.

We will show that the following two conditions are equivalent:

1. $\mathcal{G}$ is k-colorable.
2. $\mathcal{X}_G \models \bigvee_k \phi$.

Suppose $\mathcal{G}$ is k-colorable. Then, there is a function $\xi : V \to \{1, \ldots, k\}$, such that if $(v, w) \in E$, then $\xi(v) \neq \xi(w)$. Let the partition of $\mathcal{X}$ into k subsets $\mathcal{X}_i$, $i \leq k$ be defined in the following way:

$$\mathcal{X}_i = \{s_v \mid \xi(v) = i\}.$$

One can observe that $\mathcal{X}_G = \bigcup_i \mathcal{X}_i$ holds. Next we will show that $\mathcal{X}_i \models \phi$ holds for all $\mathcal{X}_i$, $i \leq k$.

Suppose $s_v, s_w \in \mathcal{X}_i$ and $=(x, y)$ is a conjunct in ϕ. Suppose $=(x, y)$ was generated because $(v, w) \in E$. Then $s_v(x) = s_w(x)$ and $s_v(y) \neq s_w(y)$, which means that $\{s_v, s_w\} \not\models \phi$, which is a contradiction with the assumption. Thus $=(x, y)$ was not generated because $(v, w) \in E$. Then, by the construction of $\mathcal{X}_G$, it holds that $s_v(x) = s_v(y) = s_w(x) = s_w(y) = 0$. Then it holds that $\{s_i, s_j\} \models =(x, y)$.

The other direction: Suppose $\mathcal{X}_G \models \bigvee_k \phi$ holds. Then by the definition of disjunction it follows that there are sets $\mathcal{X}_i$, $i \leq k$, such that $\mathcal{X}_G = \bigcup_i \mathcal{X}_i$ and $\mathcal{X}_i \models \phi$ for all i, $i \leq k$. Let ξ be defined in the following way:

- $\xi(v) = i$, if $s_v \in X_i$.

Clearly, $\xi : Var(X) \to \mathbb{N}$ is well defined. Suppose $(v, w) \in E$. Then, by the construction of $\mathcal{X}_G$ there is a conjunct in $=(x, y)$ in ϕ, such that $\{s_v, s_w\} \not\models =(x, y)$. Then s_v and s_w cannot be in the same set $\mathcal{X}_i$, thus $\xi(v) \neq \xi(w)$.

□

We also show the other direction, i.e. $D_k^*(\mathcal{M}, \mathcal{T})$ can be reduced to $k - COL$.

Theorem 4.4.7. $D_k^*(\mathcal{M}, \mathcal{T}) \leq_{LOGSPACE} k - COL$.

Proof. Given a team $\mathcal{X}$ and a k-disjunction of conjunctions of dependence atoms $\bigvee_k \bigwedge_{i \in m} =(X_i, y_i)$ we create a graph $\mathcal{G}_{\mathcal{X}} = (V, E)$ in the following way:

- For each $s_i \in \mathcal{X}$ we add $v_i \in V$.
- If $\{s_i s_j\}$ fails one of the dependence atoms $=(X_n, y_n)$, we add the pairs (v_i, v_j) and (v_j, v_i) into E.

Now, the following are equivalent:

1. $\mathcal{G}_{\mathcal{X}}$ is k-colorable.
2. $\mathcal{X} \models \bigvee_{i \in k} \phi_i$.

The proof is analogous to the proof of Theorem 4.4.6. □

Corollary 4.4.8. *$D_k^*(\mathcal{M}, \mathcal{T})$ is NP-complete for $k > 2$.*

Chapter 5

The 0-1 law and fragments of dependence logic

We consider only finite relational vocabularies τ. We fix the atomic probability to 1/2, i.e. for all k-tuples $(a_1, \dots a_k) \in M^k$ and all k-arity relations $R_i \in \tau$, $(a_1, \dots a_k) \in R_i^M$ with probability 1/2. This generates a uniform distribution of structures.

Let $\mathbf{M}_n$ be the class of all τ-structures over domain $\{0, \dots, n-1\}$. We assign probability $\mu_n(\phi)$ for each sentence of $\mathcal{D}$ in the following way:

$$\mu_n(\phi) \;=\; \frac{|\{\mathcal{M} \in \mathbf{M}_n \;:\; \mathcal{M} \models \phi\;\}|}{|\mathbf{M}_n|} \tag{5.1}$$

We are interested in the limit behavior of the probability $\mu_n(\phi)$ as the size of the structure n tends to infinity. We write $\mu(\phi)$ for the limit $\lim_{n\to\infty} \mu_n(\phi)$.

Definition 5.0.9. A logic L has the 0-1 law if for all sentences $\phi \in L$, $\mu(\phi)$ exists and is either 0 or 1.

The reason for to consider just relational vocabularies is the well-known fact that both the constant- and the function symbols fail the 0-1 law as we can see in the following example.

Example 5.0.10. Let $c \in \tau$ a constant-symbol, $f \in \tau$ a function-symbol and $R \in \tau$ a k-arity relation symbol. Then

$$\mu((c, \dots, c) \in R) = 1/2.$$

$$\mu_n(\exists x(f(x) = x)) = (1 - 1/n)^n \to_{n\to\infty} 1/e.$$

0-1 law is a result about the expressive power of the logic. One cannot express divisibility properties e.g. "the domain of the structure is divisible by n," $n \in \mathbb{N}$, in a logic with the 0-1 law. It was first shown by Glebskii, Kogan, Liogon'kii and Talanov [15] and later independently by Fagin [9] that FO has the 0-1 law. 0-1 law has been widely studied over different probability distributions for various logics and their fragments.

5.1 The 0-1 law for universal and existential sentences

Definition 5.1.1. We consider $\mathcal{D}$-formulas only in the normal form (see Definition 3.0.8). We denote a set of formulas with a quantifier prefix of the form $\forall x_1 \ldots \forall x_k \exists y_1 \ldots \exists y_m$ with $\forall^k \exists^m \mathcal{D}$. Thus $\forall^* \mathcal{D}$ denotes the set of $\mathcal{D}$-sentences with a prefix of just universal quantifiers and $\exists^* \mathcal{D}$ set of $\mathcal{D}$-sentences with a prefix of just existential quantifiers.

The whole dependence logic cannot have the 0-1 law. For example, one can express even cardinality with the following sentence:

$$\begin{aligned}&\forall x_0 \forall x_1 \exists x_2 \exists x_3 (=(x_0, x_2) \wedge =(x_1, x_3) \wedge \not\approx x_0 x_2 \\ &\quad \wedge (\approx x_0 x_1 \to \approx x_2 x_3) \wedge (\approx x_1 x_2 \to \approx x_0 x_3))\end{aligned}$$

This gives us the least known fragment of $\mathcal{D}$, which fails the 0-1-law.

Proposition 5.1.2. 0-1-*law does not hold for* $\forall\forall\exists\exists\mathcal{D}$.

To be able to quantify over two functions makes $\forall\forall\exists\exists\mathcal{D}$ a very strong fragment of $\mathcal{D}$. One can express NP-complete problems in $\forall\forall\exists\exists\mathcal{D}$, e.g. 3-colorability of a graph can be expressed with the following sentence:

$$\begin{aligned}\forall x_1 \forall x_2 \exists x_3 \exists x_4 (=(x_1, x_3) \wedge =(x_2, x_4) \quad &\wedge \quad (=(x_3) \vee =(x_3) \vee =(x_3)) \wedge \\ (\approx x_1 x_2 \to \approx x_3 x_4) \quad &\wedge \quad (E x_1 x_2 \to \not\approx x_3 x_4))\end{aligned}$$

We will show that both $\forall^* \mathcal{D}$ and $\exists^* \mathcal{D}$ have the 0-1 law.

Proposition 5.1.3. 0-1-*law holds for* $\exists^* \mathcal{D}$.

Proof. Recall that $\mathcal{D}$-sentences ϕ are evaluated with respect to the team with the empty assignment. When checking the truth value of the formula in a given structure, we first supplement the team $\{\emptyset\}$ for each existentially bound variable. After supplementing we have singleton team. Singleton teams satisfy trivially all dependence atoms, thus all dependence atoms can be replaced with $\top$ in ϕ. After replacing all dependence atoms we are left with an FO-formula ϕ^f, so-called *flattening of* ϕ (see [39] for more details), which is equivalent to the original formula ϕ. □

We will show that all formulas in the normal form without existential quantifiers have a definition in a fragment of existential second order logic, which has the 0-1 law.

Definition 5.1.4. Bernays-Schönfinkel class is the collection of first order sentences of form $\exists^* \forall^* \phi$, where ϕ is a quantifier-free first order formula.

Definition 5.1.5. Σ_1^1(B-S) is the class of existential second-order formulas of the form $\exists S_1 \ldots \exists S_k \psi(x_1, \ldots, x_k)$, where $\psi(x_1, \ldots, x_k)$ is a Bernays-Schönfinkel-formula.

The following theorem is from [27].

Theorem 5.1.6. *Σ_1^1(B-S) has the 0-1 law.*

We consider the following fragment of Σ_1^1(B-S):

Definition 5.1.7. $\Sigma_1^1(\forall)$ is the class of existential second-order formulas of the form $\exists S_1 \ldots \exists S_k \forall x_1 \ldots \forall x_k \psi(x_0, \ldots, x_k)$, where $\psi(x_1, \ldots, x_k)$ is a quantifier-free first-order formula.

Lemma 5.1.8. *$\Sigma_1^1(\forall)$ is closed under the following operations:*

1. *conjunction*
2. *existential second-order quantification*
3. *first-order universal quantification*

Proof. 1. *Conjunction*: Suppose ϕ and ψ in $\Sigma_1^1(\forall)$. Let us denote ϕ' and ψ' for the sentences obtained by replacing all the variables in ϕ and ψ so that the same variables do not occur in both formulas.

$$\phi' =: \exists R_1^\phi \ldots \exists R_l^\phi \forall x_1^\phi \ldots \forall x_k^\phi \theta_0(\bar{R}^\phi, \bar{x}^\phi),$$

$$\psi' =: \exists R_1^\psi \ldots \exists R_j^\psi \forall x_1^\psi \ldots \forall x_m^\psi \theta_1(\bar{R}^\psi, \bar{x}^\psi).$$

Then $\phi \wedge \psi$ is equivalent to

$$\exists R_1^\phi \ldots \exists R_l^\phi \exists R_1^\psi \ldots \exists R_j^\psi \forall x_1^\phi \ldots \forall x_k^\phi \forall x_1^\psi \ldots \forall x_m^\psi (\theta_0' \wedge \theta_1')(\bar{R}^\psi, \bar{R}^\phi, \bar{x}^\phi \bar{x}^\psi).$$

2. *Existential second order quantification*: Suppose $\phi(S) \in \Sigma_1^1(\forall)$ is of the following form:

$$\phi(S) =: \exists R_1 \ldots \exists R_l \forall x_1 \ldots \forall x_k \theta(S, \bar{R}, \bar{x}).$$

Then clearly $\exists S \exists R_1 \ldots \exists R_k \forall x_1 \ldots \forall x_k \theta(S, \bar{R}, \bar{x}) \in \Sigma_1^1(\forall)$.

3. *Universal first order quantification*: Suppose $\phi(x) \in \Sigma_1^1(\forall)$ is of the following form:

$$\phi =: \exists R_1 \ldots \exists R_l \forall x_1 \ldots \forall x_k \theta(\bar{R}, \bar{x}, x).$$

Let ϕ' be the formula obtained by replacing everywhere $R_i t_1 \ldots t_n$ with $R_i' x t_1 \ldots t_n$ where R_i' is of arity $\# R_i{+}1$ for all $i \leq l$. Now the following are equivalent;

- $\mathcal{M} \models_S \exists R_1' \ldots \exists R_l' \forall x_1 \ldots \forall x_k \theta'(\bar{R}', \bar{x}, x)$.
- For all $a \in M$, $\mathcal{M} \models_{s(a/x)} \exists R_1 \ldots \exists R_l \forall x_1 \ldots \forall x_k \theta(\bar{R}, \bar{x}, x)$.

□

We use the following equivalence of first order formulas:

1. $\forall x(\phi) \wedge \psi$ is equivalent to $\forall x(\phi \wedge \psi)$, where x cannot appear free in ψ.

In the next theorem we use the translation from $\mathcal{D}$ to Σ_1^1 given in [39].

Theorem 5.1.9. *Suppose $\phi \in \forall^*\mathcal{D}$, then there is a sentence $\phi^* \in \Sigma_1^1(\forall)$, such that for all structures $\mathcal{M}$ and for all teams $\mathcal{X} \supseteq Fr(\phi)$ the two conditions are equivalent.*

$$\mathcal{M} \models_{\mathcal{X}} \phi.$$

$$(\mathcal{M}, Rel(X)) \models \phi^*.$$

Proof. We use the translation from $\mathcal{D}$ to Σ_1^1 given in [39] to construct ϕ^*. . We use induction on the structure of the formula to show that $\phi^* \in \Sigma_1^1(\forall)$.

- Suppose $\phi(x_{i_1}, \ldots, x_{i_k})$ is a *positive or negated identity or relational atomic formula*. Then $\phi^*(S)$ is the following formula:

  $$\forall x_{i_1} \ldots \forall x_{i_k}(Sx_{i_1}, \ldots, x_{i_k} \to \phi(x_{i_1}, \ldots, x_{i_k}).$$

  Clearly it holds that $\phi^*(S) \in \Sigma_1^1(\forall)$.

- *Dependence atom*: Suppose ϕ is $=(x_{i_1}, \ldots, x_{i_k}))$. Then $\phi^*(S)$ is the following formula:

  $$\forall x_{i_1} \ldots \forall x_{i_k} \forall x_{i_1}' \ldots \forall x_{i_k}'((Sx_{i_1}, \ldots, x_{i_k} \wedge Sx_{i_1}', \ldots, x_{i_k}' \\ \bigwedge_{j=1}^{k-1} x_{i_j} = x_{i_j}') \to x_{i_k} = x_{i_k}')$$

- *Negated dependence atom*: Suppose $\phi(x_{i_1}, \ldots, x_{i_k})$ is $\neg = (x_{i_1}, \ldots, x_{i_k})$. Then $\phi^*(S)$ is the following formula:

  $$\forall x_{i_1} \ldots \forall x_{i_k} \neg S x_{i_1}, \ldots, x_{i_k}.$$

  Again one can observe that in both cases $\phi^*(S) \in \Sigma_1^1(\forall)$.

- *Disjunction*: Suppose $\phi(x_{i_1}, \ldots, x_{i_k})$ is a $\psi(x_{i_1}, \ldots, x_{i_n}) \vee \theta(x_{i_1}, \ldots, x_{i_m})$, where $\{x_{i_1}, \ldots, x_{i_k}\} = \{x_{i_1}, \ldots, x_{i_n}\} \cup \{x_{i_1}, \ldots, x_{i_m}\}$. Then $\phi^*(S)$ is the following formula:

  $$\exists R \exists T(\psi^*(R) \wedge \theta^*(T) \wedge \forall x_{i_1} \ldots \forall x_{i_k}(Sx_{i_1}, \ldots, x_{i_k} \to \\ (Rx_{i_1}, \ldots, x_{i_k} \vee Tx_{i_1}, \ldots, x_{i_k}))).$$

By induction hypothesis, it holds that $\psi^*(R)$ and $\theta^*(T)$ are both in $\Sigma^1_1(\forall)$. Now, since $\Sigma^1_1(\forall)$ is closed under conjunction and existential second order quantification by 5.1.8, and using the rule for universal quantifier (1), we can observe that $\phi^*(S) \in \Sigma^1_1(\forall)$.

- *Universal quantification*: Suppose $\phi(x_{i_1}, \ldots, x_{i_k})$ is the formula $\forall x_{i_{k+1}} \psi(x_{i_1}, \ldots, x_{i_{k+1}})$. Then $\phi^*(S)$ is the following formula:

$$\exists R(\psi^*(R) \wedge \forall x_{i_1} \ldots \forall x_{i_k}(Sx_{i_1}, \ldots, x_{i_k} \to \forall x_{i_{k+1}} Rx_{i_1}, \ldots, x_{i_{k+1}})).$$

  By induction hypothesis $\psi^*(R) \in \Sigma^1_1(\forall)$. Again by 5.1.8 and rule (1) it follows that $\phi^*(S) \in \Sigma^1_1(\forall)$.

We used here the translation from $\mathcal{D}$ to Σ^1_1 given in [39] to obtain the definition $\phi^*(S)$ in Σ^1_1 for each $\phi \in \forall^*\mathcal{D}$. The equivalence of ϕ and $\phi^*(S)$ is shown by direct induction on the structure of the formula (more details see [39]). $\square$

Proposition 5.1.10. $\forall^*\mathcal{D}$ *has the* 0-1 *law.*

Proof. Suppose $\phi \in \forall^*\mathcal{D}$. Then, there is a sentences $\phi^* \in \Sigma^1_1(\forall)$, such that for all structures $\mathcal{M}$ and for all teams $\mathcal{X}$ the two conditions are equivalent.

$$\mathcal{M} \models_{\mathcal{X}} \phi.$$

$$(\mathcal{M}, Rel(X)) \models \phi^*.$$

Thus by Theorem 5.1.6 it holds that $\mu(\phi) = \mu(\phi^*) \in \{0, 1\}$. $\square$

5.2 The 0-1 law for quantifier-free formulas

We will consider the 0-1 law for quantifier-free formulas over finite teams. Usually 0-1-law is studied only for sentences as in previous subsection. Here however, we consider quantifier-free formulas instead of sentences and teams instead of structures.

Let $\mathbf{K}^n_k$ be the set of all pairs $(\mathcal{M}, \mathcal{X})$, where $\mathcal{M}$ is a τ-structure of the domain $\{1, \ldots, n\}$ and $\mathcal{X}$ is a team with domain $\{x_1, \ldots x_k\}$ and range $\{1, \ldots, n\}$. Let $\mathcal{D}^k(\tau)$ be the set of all τ-formulas with variables $\{x_1, \ldots x_k\}$.

Definition 5.2.1. Suppose $\phi \in \mathcal{D}^k(\tau)$. Then the probability $\mu_n(\phi)$ is defined in the following way:

$$\mu_n(\phi) \;=\; \frac{|\{(\mathcal{M}, \mathcal{X}) \in \mathbf{K}^n_k \;:\; \mathcal{M} \models_{\mathcal{X}} \phi \;\}|}{|\mathbf{K}^n_k|} \tag{5.2}$$

We write $\mu(\phi)$ for the limit $\lim_{n\to\infty} \mu_n(\phi)$.

Theorem 5.2.2. *Suppose ϕ is a quantifier-free $\mathcal{D}$-formula. Then $\mu(\phi) \in \{0, 1\}$.*

Proof. It follows from Theorem 5.1.9 that for each quantifier free $\phi \in \mathcal{D}(\tau)^k$ there is a sentence $\phi^* \in \Sigma_1^1(\forall)$ in vocabulary $(\tau \cup \{R\})$, where R is interpreted as $\mathrm{Rel}(\mathcal{X})$, such that the following are equivalent:

$$\mathcal{M} \models_{\mathcal{X}} \phi.$$

$$(\mathcal{M}, Rel(\mathcal{X})) \models \phi^*.$$

Let us denote the set of $(\tau \cup \{R\})$-structures with domain $\{1, \dots, n\}$, where $\#R = k$ by ${\mathbf{K}^*}_k^n$. It holds that

$$\mu(\phi) = \frac{|\{(\mathcal{M}, \mathcal{X}) \in \mathbf{K}_k^n \ : \ \mathcal{M} \models_{\mathcal{X}} \phi \ \}|}{|\mathbf{K}_k^n|}$$

$$= \frac{|\{(\mathcal{M}, Rel(\mathcal{X})) \in {\mathbf{K}^*}_k^n \ : \ (\mathcal{M}, Rel(\mathcal{X})) \models \phi^* \ \}|}{|{\mathbf{K}^*}_k^n|} = \mu(\phi^*).$$

Now it holds that $\mu(\phi) = \mu(\phi^*) \in \{0, 1\}$ by Theorem 5.1.6. $\square$

Chapter 6

Generalized quantifiers and the 0-1 law

We investigate whether the 0-1 law holds for the extensions of logic $\mathcal{L}^{\omega}_{\infty\omega}$ and it's fragments $\mathcal{L}^{k}_{\infty\omega}$ by a simple unary generalized quantifier over uniform distribution of finite graphs. We concentrate on a specific class of quantifiers, which allow us to express things like " *At least half* of the people in ILLC are interested in quantifiers" or "*At most five out of eleven* of mathematicians are interested in philosophy." More precisely, we will consider simple unary quantifiers of the following form:

$$\exists^{s/t} = \{(M, P^M) \ : \ |P^M| \geq s/t \cdot |M|\} \tag{6.1}$$

Knyazev studied the probabilities of first-order sentences defined with quantifiers of form (6.1) in [26]. He showed that the sentences defined with quantifiers of form $\exists^{s/t}$, where $m \neq 2^m$ has the 0-1 law. Kaila studied in his thesis in a more general setting the 0-1 law and the convergence law for $\mathcal{L}^{k}_{\infty\omega}(Q)$ [24]. We extend these results with a complete characterization for the 0-1 law of $\mathcal{L}^{k}_{\infty\omega}(\exists^{s/t})$. We establish a connection between the form of the quantifier and the number of variables allowed in the formulas. We will show that a dichotomy holds for the 0-1 law of $\mathcal{L}^{k}_{\infty\omega}(\exists^{s/t})$, i.e. we will show that if t is not of the form 2^m for any $m \in \mathbb{N}$, then the logic $\mathcal{L}^{k}_{\infty\omega}(\exists^{s/t})$ has the 0-1 law. On the other hand, if t is of the form 2^m for some $m \in \mathbb{N}$, then $\mathcal{L}^{k}_{\infty\omega}(\exists^{s/t})$ has the 0-1 law if and only if $log_2 t$ is larger or equal to k, i.e. if we allow less than $log_2 t$ distinct variables in the formulas.

The first 0-1 law result is by Glebskii, Kogan, Liogon'kii and Talanov [15], which is that the set of first order sentences has the 0-1 law. This was soon after shown independently also by Fagin [10]. Fagin's proof for the 0-1 law for FO relies on properties called *Extension axioms*. We will use a stronger version of these axioms, so-called *Strong extension axioms*, which were introduced by Shelah in [35]. Another method we use are so-called *pebble games*, which can be used to show that given two structures are elementarily equivalent up to a certain degree, or alternatively that they are not. We will show how the strong extension axioms provide the second player a winning strategy in the monotone $(k, \{\exists^{s/t}\})$-pebble game.

6.1 Preliminaries

We consider only finite graphs. We study the symmetric case, where the atomic probability is 1/2, which means there is an edge between two vertex with probability 1/2. This leads to a uniform distribution of the graphs.

Definition 6.1.1. Let $[n] = \{1, \ldots, n\}$ and $\mathbf{G}_n$ the class of all graphs with $[n]$ as universe:

$$\mathbf{G}_n = \{\mathcal{G} \mid V^{\mathcal{G}} = [n]\}.$$

Let probability $\mu_n : L \to [0,1]$ be defined as follows for all formulas ϕ.

$$\mu_n(\phi) \;=\; \frac{|\{\mathcal{G} \in \mathbf{G}_n \;:\; \mathcal{G} \models \phi\,\}|}{|\mathbf{G}_n|}.$$

We say that ϕ is true in random graph of cardinality n with probability $\mu_n(\phi)$. We are interested in the asymptotic behavior of $\mu_n(\phi)$ as n grows. We let

$$\lim_{n\to\infty} \mu_n(\phi) = \mu(\phi),$$

if the limit exists. We say that ϕ is satisfied by almost all graphs if $\mu(\phi) = 1$.

Definition 6.1.2. A logic $\mathcal{L}$ is said to have the 0-1 *law*, if for all sentences $\phi \in \mathcal{L}$, $\mu(\phi)$ is defined and is either 0 or 1.

Lindström gave the following formalization for generalized quantifiers in [32]:

Definition 6.1.3. Let $(r_1, \ldots, r_n)$ be a tuple of natural numbers. A *Lindström quantifier* of type $(r_1, \ldots, r_n)$ is a collection Q of structures of relational vocabulary $\tau_s = (P_1, \ldots, P_n)$ such that P_i is of arty r_i for $1 \le i \le r$, and Q is closed under isomorphisms. The arity of a quantifier Q is $ar(Q) = max\{ar(P_1), \ldots, ar(P_n)\}$. Q is called *simple* if $n = 1$ and *unary* if $ar(Q) = 1$.

Definition 6.1.4. A simple unary generalized quantifier Q is *monotone* (increasing), if for all structures $(M, P^M) \in Q$ and for all subsets $X \subseteq M$ such that $P^M \subseteq X$, then also $(M, X) \in Q$.

We focus on monotone simple unary quantifiers of the following form:

Definition 6.1.5. Suppose $\alpha \in \mathbb{R}$, $0 < \alpha < 1$. Let $\exists^{\ge\alpha}$ is defined in the following way:

$$\exists^{\ge\alpha} = \{\ (M, P^M) \;:\; |P^M| \ge \alpha \cdot |M|\ \} \tag{6.2}$$

we consider quantifiers $\exists^{\ge\alpha}$, where α is a rational s/t in a reduced form[1].

[1] s and t have no common factors.

The logic $\mathcal{L}^{\omega}_{\infty\omega}$ is an extension of FO, which allows infinite disjunctions and conjunctions. Each formula of $\mathcal{L}^{\omega}_{\infty\omega}$ can use only finite number of distinct variables. We write $\mathcal{L}^{k}_{\infty\omega}$ for the set $\mathcal{L}^{\omega}_{\infty\omega}$-formulas with at most k variables. The 0-1 law for $\mathcal{L}^{\omega}_{\infty\omega}$ was shown in [30]. We extend $\mathcal{L}^{\omega}_{\infty\omega}$ by a single quantifier $\exists^{\geq s/t}$. We denote the obtained logic by $\mathcal{L}^{\omega}_{\infty\omega}(\exists^{\geq s/t})$ and it's k-variable fragment with $\mathcal{L}^{k}_{\infty\omega}(\exists^{\geq s/t)}$. The semantics of the logic $\mathcal{L}^{k}_{\infty\omega}(\exists^{\geq s/t)}$ extend the semantics of FO for the infinite conjunctinos and disjunctions as well as for the quantifier $\exists^{\geq s/t}$. Infinite conjunctions $\bigwedge_{n\in\omega}\phi_n$ are considered true if all of the conjuncts ϕ_n are true, whereas infinite disjunctions $\bigvee_{n\in\omega}\phi_n$ are considered true if at least one of the disjuncts is true. The semantics in the case of quantifier $\exists^{\geq s/t}$ is defined in the following way:

$$\mathcal{M} \models \exists^{\geq s/t} x\phi(x) \Leftrightarrow (M, \{a \in M \mid (\mathcal{M}, a) \models \phi(x)\}) \in \exists^{\geq s/t} \tag{6.3}$$

6.2 Strong extension axioms

Models of the k-extension axiom are all elementarily equivalent with respect to first order logic formulas up to k variables. In the case of $\mathcal{L}^{k}_{\infty\omega}(\exists^{\geq s/t})$ the normal extension axioms do not suffice to characterize one equivalence class of $\equiv_{\mathcal{L}^{k}_{\infty\omega}(\exists^{\geq s/t})}$. Shelah [35] introduced so called strong extension axioms, which naturally correspond to the quantifiers we are considering. We use the following concept of a k-type of graphs to define the strong extension axioms.

Definition 6.2.1. A $k-$*type* $t(x_1, \ldots, x_k)$ over graphs is a maximal consistent set of formulas $E(x_i, x_j)$, $\neg E(x_i, x_j)$ and identity and negated identity formulas with variables $\{x_1, \ldots, x_k\}$. A k-type t is *proper* if it includes all the negated identity formulas (except of the form $\neg \approx xx$). We denote the conjunction over formulas in $t(x_1, \ldots, x_k)$ by $\phi_t(x_1, \ldots, x_k)$.

Definition 6.2.2. A $k+1-$type $s(x_1, \ldots, x_{k+1})$ extends $k-$type $t(x_1, \ldots, x_k)$, if $t \subseteq s$. A $k+1-$type $s(x_1, \ldots, x_{k+1})$ extends $t(x_1, \ldots, x_k)$ *properly*, if $(\neg \approx x_{k+1}x_i) \in s$ for all i, $0 < i \leq k$.

Definition 6.2.3. Suppose $\mathcal{G}$ is a graph and $(v_1, \ldots, v_k)$ a sequence of vertices of $\mathcal{G}$. We say that sequence $(v_1, \ldots, v_k)$ *realizes* the $k-$type $t(x_1, \ldots, x_k)$ in $\mathcal{G}$, if

$$(\mathcal{G}, v_1, \ldots, v_k) \models \phi_t(x_1, \ldots, x_k).$$

We denote the type realized by $(v_1, \ldots, v_k)$ in $\mathcal{G}$ by $t^{\mathcal{G}}_{\bar{v}}$.

Definition 6.2.4. Suppose $t(x_1, \ldots, x_k)$ is a proper $k-$type and $s(x_1, \ldots, x_{k+1})$ a $k+1$-type properly extending $t(x_1, \ldots, x_k)$. For all $\alpha \in \mathbb{R}$, $0 < \alpha < 1$, *the strong extension axiom $SEA^{\alpha}_k(t, s)$ associated to a pair of types* (t, s) is the following sentence:

$$\forall x_1 \ldots \forall x_k \, (\phi_t(x_1, \ldots, x_k) \; \rightarrow \exists^{\geq \alpha/2^k} x_{k+1} \; \phi_s(x_1, \ldots, x_k, x_{k+1})).$$

The *strong extension axiom* SEA_k^α *over graphs* is a conjunction of $SEA_k^\alpha(t,s)$ over pairs (t,s), where s properly extends t.

When k distinct vertices $(v_1,\ldots,v_k)$ of a graph $\mathcal{G}$ are fixed, there is a natural partition of the remaining $n-k$ vertices into 2^k disjoint sets by means of which proper extensions of $t_{\bar{v}}^{\mathcal{G}}$ the vertices realize. Now the k-extension axiom says that every set of the partition is nonempty whereas the strong extension axioms say that every set of the partition contains almost the expected number of vertices, i.e. at least $|V^G|\cdot\alpha/2^k$ vertices,$\alpha<1$.

Shelah showed that the strong extension axioms have asymptotic probability 1 [35]. Blass and Gurevich have extended Shelah's work in [3]. The following Lemma and Theorem can be found in [3].

Lemma 6.2.5. *Fix numbers* β,r *in the open interval* $(0,1)$*. There is a constant* $c\in(0,1)$ *such that the following is true for every positive integer* m*. Let* X *be the number of successes in* m *independent trials, each having probability* r *of success. Then probability* $P(X\leq\beta mr)\leq c^m$*.*

The limit probability of the strong extension axiom SEA_k^α is calculated in [3] for $\alpha=1/2$. They also state that limit can be calculated similarly for all $\alpha<1$.

Theorem 6.2.6. *For all* $k\in\mathbb{N}$ *and for all* $\alpha\in\mathbb{R}$, $0<\alpha<1$*, it holds that*

$$\mu(SEA_k^\alpha)=1. \tag{6.4}$$

Proof. The strong extension axiom SEA_k^α fails in a random graph $\mathcal{G}$, if at least one of the axioms $SEA_k^\alpha(t,t')$ fail in $\mathcal{G}$. Let n be the size of the graph and $(v_1,\ldots,v_k)$ sequence of vertices of $\mathcal{G}$, such that $t_{\bar{v}}^{\mathcal{G}}=t$. There exists $n-k$ vertices in $\mathcal{G}$ that could extend t to t'. Let X be a random variable that gives the number of vertices of $\mathcal{G}$ which extend $t_{\bar{v}}$ to t'. Suppose $\beta\in(0,1)$, $\beta>\alpha$, such that for all large $n\in\mathbb{N}$ holds $\alpha n\leq\beta(n-k)$. Now 6.2.5 yields

$$\begin{aligned}P(X\leq\alpha n/2^k)&\leq P(X<\beta(n-k)/2^k)\\&\leq c^{n-k}.\end{aligned}$$

There are at most n^k k-sequences $(v_1,\ldots,v_k)$ that could realize type t. Thus the probability for $SEA_k^\alpha(t,t')$ to fail in a random graph can be estimated in the following way:

$$\mu_n(\neg SEA_k^\alpha(t,t'))\leq n^k c^{n-k}$$

It holds that $0<c<1$, so this bound tends to 0 as $n\to\infty$. Since the number of pairs of (t,t') is finite for fixed k and not dependent on the cardinality of the graph n, also the probability that SEA_k^α fails tends to 0. $\square$

6.3 Monotone $(k, \mathcal{Q})$-pebble game

The Ehrenfeucht-Fraïssé games are a family of methods considered in model theory. They are used to determine whether or not given structures are elementarily equivalent. Especially in finite model theory, where many of the model theoretical tools to show inexpressibility results fail, the Ehrenfeuch-Fraïssé method stays valid. The monotone $(k, \mathcal{Q})$-pebble game introduced in [29] characterizes the elementary equivalence between two structures with respect to logics extended by unary simple monotone generalized quantifiers.

Definition 6.3.1. Suppose $\mathcal{G}$ and $\mathcal{G}'$ are graphs and k is a positive natural number. Let $m \leq k$ and vertices $v_1, \ldots, v_m \in G$ and $v'_1, \ldots, v'_m \in G'$. We write

$$(\mathcal{G}, v_1, \ldots, v_m) \equiv_{\mathcal{L}^k_{\infty\omega}(\exists^{\geq s/t})} (\mathcal{G}', v'_1, \ldots, v'_m)$$

if for every formula $\varphi(x_1, \ldots, x_m) \in \mathcal{L}^k_{\infty\omega}(\exists^{\geq s/t})$ the following holds

$$(\mathcal{G}, v_1, \ldots, v_m) \models \varphi(x_1, \ldots, x_m) \quad \Leftrightarrow \quad (\mathcal{G}', v'_1, \ldots, v'_m) \models \varphi(x_1, \ldots, x_m).$$

We say that $\mathcal{G}$ is $\mathcal{L}^k_{\infty\omega}(\exists^{\geq s/t})$-*equivalent* with $\mathcal{G}'$, denoted by

$$\mathcal{G} \equiv_{\mathcal{L}^k_{\infty\omega}(\exists^{\geq s/t})} \mathcal{G}',$$

if the graphs $\mathcal{G}$ and $\mathcal{G}'$ satisfy the same sentences of $\mathcal{L}^k_{\infty\omega}(\exists^{\geq s/t})$.

The monotone $(k, \mathcal{Q})$-pebble game is defined for a set $\mathcal{Q}$ of monotone simple unary quantifiers.

Definition 6.3.2. Suppose $\mathcal{Q} = \{Q_i : i \in I\}$ is set of monotone simple unary generalized quantifiers and k is a natural number. The *monotone* $(k, \mathcal{Q})$-*pebble game* on structures $\mathcal{G}$ and $\mathcal{G}'$ is played in turns between two players: Player I, which we call Abelard and player II, which we call Eloise. Abelard moves first. There are two possible types of moves for him.

1. **Pebble move**: Abelard chooses one of the structures $\mathcal{G}$ or $\mathcal{G}'$, and plays a pebble on an element of the structure he chose. After this Eloise plays a pebble on an element of the other structure.

2. **Quantifier move**: Abelard chooses one of the structures, say $\mathcal{G}$, and a quantifier $Q_i \in \mathcal{Q}$ and a set $A \subseteq V^G$, such that $(V^G, A) \in Q_i$.

   Eloise responds by choosing a subset B of vertices of the other structure $\mathcal{G}'$, such that $(V^{G'}, B) \in Q_i$. After this Abelard plays a pebble on an element of B and then Eloise responds by playing a pebble on an element of A.

There are k pairs of pebbles in the game. When all the pebbles are played, Abelard can choose to remove one pair of pebbles and the game resumes. **Winning condition**: Let v_i be the vertices of $\mathcal{G}$ and v'_i vertices of $\mathcal{G}'$ pebbled on the i:th round. Eloise wins the game, if she is able to play so that the mapping $v_i \mapsto v'_i$ is a partial isomorphism between $\mathcal{G}$ and $\mathcal{G}'$ after each played round. Otherwise Abelard wins.

Theorem 6.3.3. *Let $\mathcal{Q} = \{Q_i : i \in I\}$ set of monotone simple unary generalized quantifiers, $\mathcal{G}$ and $\mathcal{G}'$ graphs and $k \in \mathbb{N}$. Then the following are equivalent:*

1. $\mathcal{G} \equiv_{\mathcal{L}^k_{\infty\omega}(\mathcal{Q})} \mathcal{G}'$.
2. *Eloise has a winning strategy in the monotone $(k, \mathcal{Q})-$pebble game on graphs $\mathcal{G}$ and $\mathcal{G}'$.*

Proof. [29] □

6.4 The 0-1 law

We will show that the strong extension axioms SEA^{α}_{k-1} provide a winning strategy for *Eloise* in the monotone $(k, \exists^{\geq s/t})$-pebble game played on large graphs when $t \neq 2^m$ for all $m \in \mathbb{N}$, or when $t = 2^m$ for some $m \in \mathbb{N}$ and $k \leq m$.

We define two characteristics for a logic $\mathcal{L}^k_{\infty\omega}(\exists^{\geq s/t})$. A lower bound n_ω for the cardinality of the graphs and a lower bound α_ω for the parameter α of the strong extension axiom SEA^{α}_k.

Definition 6.4.1. Suppose $k, s, t \in \mathbb{N}$. Let p_k be the least natural number for s/t and k, such that

$$p_k/2^k > s/t.$$

Suppose $\mathcal{M}$ is a finite structure and $a_i \in M$, $i \leq k$, distinct elements. Furthermore, suppose that the elements of the domain are uniformly distributed over the proper extensions of the type $t^{\mathcal{M}}_{\bar{a}}$. Now, the intuitive meaning of the characteristic number p_k is that, if we want to pick a set $A \subseteq Dom(\mathcal{M})$ which is in the quantifier $\exists^{\geq s/t}$, then the set A must contain realizers to at least p_k different proper extensions of the type $t^{\mathcal{M}}_{\bar{a}}$.

We want to be sure that the number of realizers of p_k different k-types given us by SEA^{α}_k is big enough to be in the quantifier $\exists^{\geq s/t}$. We also want that the realizers of any $(p_k - 1)$ different types plus the already played vertices not to be big enough to be in $\exists^{\geq s/t}$. The following lemma shows that we can have that as long as we consider large enough graphs.

Lemma 6.4.2. *Suppose $k \in \mathbb{N}$, $\exists^{\geq s/t}$ a quantifier as in 6.1.5, $t \neq 2^m$ for all $m \in \mathbb{N}$ or $t = 2^m$ and $k < m$ for some $m \in \mathbb{N}$. Then there is $\alpha_k \in \mathbb{R}$, $0 \leq \alpha_k < 1$, and $n_k \in \mathbb{N}$, such that the following two conditions hold whenever $\alpha \geq \alpha_k$ and $n \geq n_k$:*

1. $\alpha \cdot p_k/2^k \geq s/t$.
2. $\alpha \cdot (p_k - 1)n/2^k + k + (1-\alpha)n < s/t \cdot n$.

Proof. Suppose $t \neq 2^m$ for all $m \in \mathbb{N}$, or $t = 2^m$ and $k < m$. Now it holds by Definition 6.4.1 and by the assumption on t that $p_k/2^k > s/t$, thus we can choose α_k, such that $\alpha \cdot p_k/2^k \geq s/t$, for all $\alpha \geq \alpha_k$. The second condition is equivalent to:

$$(p_k - 1)\alpha \cdot /2^k + k/n + (1-\alpha) < s/t.$$

The left side of the equation tends to $(p_k - 1)/2^k$ as n tends to infinity and α tends to one. By 6.4.1 and assumption on t, it holds that $(p_k - 1)/2^k < s/t$. Thus we can choose n_k and α_k, such that also the second condition holds for all α, $\alpha_k < \alpha < 1$ and $n \geq n_k$. □

Definition 6.4.3. Let n_i and α_i, $i \leq k$, be defined as in 6.4.2, such that conditions 1) and 2) in 6.4.2 hold. Then, let

$$n_\omega = \max\{n_i \mid i \leq k\},$$

$$\alpha_\omega = \max\{\alpha_i \mid i \leq k\}.$$

Eloise wins the monotone $(k+1, \exists^{\geq s/2^k})$-pebble game on graphs $\mathcal{G}$ and $\mathcal{G}'$, if graphs $\mathcal{G}$ and $\mathcal{G}'$ are larger than n_ω and they both satisfy the strong extension axiom SEA_k^α where $\alpha \geq \alpha_\omega$.

Theorem 6.4.4. *Suppose $t \neq 2^m$ for all $m \in \mathbb{N}$ or $t = 2^m$ for some $m \in \mathbb{N}$ and $k \leq m$. Let $\mathcal{G}$ and $\mathcal{G}'$ be graphs, $|V^G|, |V^{G'}| \geq n_\omega$, and suppose both satisfy SEA_k^α for some $\alpha > \alpha_\omega$. Then Eloise has a winning strategy in the monotone $(k+1, \exists^{\geq s/t})$-pebble game on graphs $\mathcal{G}$ and $\mathcal{G}'$.*

Proof. The proof is by induction on the length of the game.

First move: Since the graph relation is irreflexive, it does not matter which vertex Eloise pebbles.

Induction step: Suppose l pebbles have been played, $l < k+1$, on vertices $v_1, \ldots, v_l \in G$ and $v'_1, \ldots, v'_l \in G'$, and it holds that $v_i \mapsto v'_i$ is a partial isomorphism between $\mathcal{G}$ and $\mathcal{G}'$, i.e. $t_{\bar{v}}^{\mathcal{G}} = t_{\bar{v}'}^{\mathcal{G}'}$.

Suppose Abelard plays the ordinary pebble move, let us say a vertex $v \in \mathcal{G}$. Then Eloise plays a vertex $v' \in \mathcal{G}'$ such that $t_{\bar{v},v}^{\mathcal{G}} = t_{\bar{v},v'}^{\mathcal{G}'}$. SEA_k^α guarantees that Eloise always finds such v'.

Suppose Abelard plays a set $A \subseteq V^G$, such that $(V^G, A) \in \exists^{\geq s/t}$. By Lemma 6.4.2 A consists of vertices that realize at least p_l different proper extensions of $t_{\bar{v}}^{\mathcal{G}}$. The strategy for Eloise is to play a set $B \subseteq V^{G'}$, such that $v' \in B$ iff $t_{\bar{v}',v'}^{\mathcal{G}'} = t_{\bar{v},v}^{\mathcal{G}}$ for some $v \in A$. Since $\mathcal{G}'$ satisfies the strong extension axioms $SEA_k^{\alpha_\omega}$, there are realizers for every possible extension of $t_{\bar{v}',v'}^{\mathcal{G}'}$ in B. Thus B consists of all the vertices that realize these (at least p_l) different proper extensions of $t_{\bar{v}'}^{\mathcal{G}'}$. It follows from from the fact that $\mathcal{G}'$ satisfies the strong extension axioms $SEA_k^{\alpha_\omega}$ with Lemma 6.4.2 that $|B| \geq s/t|V^{G'}|$, thus B is a legal move for Eloise. When Abelard chooses some $v' \in B$, Eloise finds always a corresponding vertices $v \in A$, such that $t^{\mathcal{G}'}_{\bar{v}',v'} = t^{\mathcal{G}}_{\bar{v},v}$ because of the way B was chosen. $\square$

The 0-1 law for the logics $\mathcal{L}^k_{\infty\omega}(\exists^{\geq s/t})$, where $t \neq 2^m$ for all $m \in \mathbb{N}$ or $t = 2^m$ for some $m \in \mathbb{N}$ and $k \leq m$ follows from the last theorem:

Theorem 6.4.5. *The* 0-1 *law holds for the logics* $\mathcal{L}^k_{\infty\omega}(\exists^{\geq s/t})$*, when* $t \neq 2^m$ *for all* $m \in \mathbb{N}$ *or* $t = 2^m$ *for some* $m \in \mathbb{N}$ *and* $k \leq m$

Proof. Suppose $t \neq 2^m$ for all $m \in \mathbb{N}$ or $t = 2^m$ for some $m \in \mathbb{N}$ and $k \leq m$. Suppose $\phi \in \mathcal{L}^k_{\infty\omega}(\exists^{\geq s/t})$ is a sentence and $\mathcal{G}$ is a graph of cardinality larger than n_0, such that $\mathcal{G} \models SEA^{\alpha_0}_{k-1}$.

Suppose $\mathcal{G} \models \phi$. Then by Theorem 6.3.3 and Theorem 6.4.4 every large enough graph(larger than n_ω) $\mathcal{G}'$, which satisfies SEA^{α}_{k-1} also satisfies ϕ. Thus by Theorem 6.2.6 $\mu(\phi) = \mu(SEA^{\alpha}_{k-1}) = 1$.

Suppose $\mathcal{G} \not\models \phi$. Then $\mathcal{G} \models \neg\phi$. By Theorem 6.3.3 and Theorem 6.4.4 every $\mathcal{G}'$ which satisfies SEA^{α}_{k-1} also satisfies $\neg\phi$. Thus by Theorem 6.2.6 $\mu(\neg\phi) = \mu(SEA^{\alpha}_{k-1}) = 1 \leftrightarrow \mu(\phi) = 0$. $\square$

6.4.1 Sentence for breaking the 0-1 law

Next we will show that the 0-1 law does not hold for logics $\mathcal{L}^k_{\infty\omega}(\exists^{\geq s/t})$, where $t = 2^m$ for some $m \in \mathbb{N}$ and $k \geq m$. We will construct sentences which have no limit probability and therefore break the 0-1 law. We will first show two lemmas we will use to calculate the probability of the sentences to be satisfied in a random graph.

Definition 6.4.6. Suppose $\mathcal{G}$ is a graph and $v \in V^{\mathcal{G}}$. Let $deg(v)$ denote degree of v, i.e. the number of vertices $w \in V^{\mathcal{G}}$ for which $(v, w) \in E^{\mathcal{G}}$ hold.

We will first show the following lemma which we will need in the proof. Let X_i, $i \in \mathbb{N}$, be mutually independent random variables defined in the following way:

- $P(X_i = 1) = \frac{1}{2}$.
- $P(X_i = 0) = \frac{1}{2}$.

Let us denote $S(n)$ for the sum $\Sigma_{i=1}^n X_i$.

Lemma 6.4.7. *For each* $k \in \mathbb{N}$ *the following holds:*

$$P(S(2n-k) = n) = \frac{1 + \mathcal{O}(n^{-1})}{\sqrt{\pi n}} \tag{6.5}$$

Proof.

$$P((S(2n-k)=n)) = \binom{2n-k}{n} \cdot \frac{1}{2^{2n-k}}$$

Applying Stirling's formula $n! \approx \sqrt{2\pi n}(\frac{n}{e})^n \cdot (1+\mathcal{O}(n^{-1}))$ yields

$$\frac{1+\mathcal{O}(n^{-1})}{\sqrt{\pi n}}.$$

□

Let $I(i,j)$ be the characteristic function of the edge relation E^G:

- $I(i,j) = 1$, if $(v_i v_j) \in E^G$.
- $I(i,j) = 0$, if $(v_i v_j) \notin E^G$.

Except for the cases $I(i,j)$ and $I(j,i)$ the values are independent of each other. Let

$$A_i = \sum_{j=1}^{2n-k} I_{(i,j)} = n.$$

Thus A_i is the event that the vertex v is connected to exactly n vertices out of $2n-k$ vertices.

We want to show that one can find a vertex with degree exactly $V^G/2$ in even cardinality graphs even if every vertex has some k edges fixed.

Lemma 6.4.8. *For each $k \in \mathbb{N}$ it holds that*

$$P(\bigcup_{i=1}^{2n-k} A_i)) \to_{n\to\infty} 1. \tag{6.6}$$

Proof. Notice that the events A_i are not independent. The probability that both of the events A_i or A_j are realized, $P(A_i \cap A_j)$, can be calculated in the following way:

$$\begin{aligned} P(A_i \cap A_j) &= \sum_{m=0}^{1} P(I_{ij}=m) P\left(\sum_{k=1}^{2n-k} I_{ik} = n | (I_{ij}=m)\right) \\ &= \frac{1}{2}\left[\binom{2n-k-1}{n}\frac{1}{2^{2n-k-1}}\right]^2 + \frac{1}{2}\left[\binom{2n-k-1}{n-1}\frac{1}{2^{2n-k-1}}\right]^2 \\ &= \frac{1+\mathcal{O}(n^{-1})}{\pi n}. \end{aligned}$$

We will use so-called Chung-Erdös equality [4] to give a lower bound for the probability $P(\bigcup_{i=1}^{2n-k} A_i))$.

$$P\left(\bigcup_{i=1}^{N} A_i\right) \geq \frac{\left[\sum_{i=1}^{N} P(A_i)\right]^2}{\sum_{i=1}^{N}\sum_{j=1}^{N} P(A_i \cap A_j)},$$

which yields

$$
\begin{aligned}
P\left(\bigcup_{i=1}^{2n-k} A_i\right) &\geq \frac{\left[\frac{(2n-k)(1+\mathcal{O}(n^{-1})}{\sqrt{\pi n}}\right]^2}{\frac{(2n-k)(1+\mathcal{O}(n^{-1}))}{\sqrt{\pi n}} + \frac{(2n-k)((2n-k-1))(1+\mathcal{O}(n^{-1}))}{\pi n}} \\
&= 1 - \mathcal{O}(n^{-1/2}).
\end{aligned}
$$

□

Next we will show that the 0-1 law does not hold for the logic $\mathcal{L}^{m+1}_{\infty\omega}(\exists^{\geq s/2^m})$. We will construct a sentence which is true in almost all the graphs of cardinality divisible by 2^m and false in all other graphs. The idea is that when the cardinality of the graph is divisible by 2^m we can partition the universe into 2^m disjoint sets of size $|V^G|/2^m$ with asymptotic probability 1. If the cardinality of the graph is not divisible by 2^m the division is clearly impossible. Thus the probability of the sentence will oscillate between zero and some probability converging to one.

Theorem 6.4.9. *$\mathcal{L}^{m+1}_{\infty\omega}(\exists^{\geq s/2^m})$ fails the 0-1 law.*

Proof. Let us first as an example consider the simplest case, that is the quantifier $\exists^{\geq 1/2}$. Then the sentence $\Phi_{1/2}$ is the following:

$$\Phi_{1/2} =: \exists x(\exists^{\geq 1/2} y(Exy) \wedge \exists^{\geq 1/2} y(\neg Exy)) \tag{6.7}$$

Let us look at the probability $\mu_n(\Phi_{1/2})$. Clearly $\mu_n(\Phi_{1/2}) = 0$ for graphs of odd cardinality. For graphs of even cardinality $\mu_n(\Phi_{1/2}) = P\left(\bigcup_{i=1}^n A_i\right)$, which has by the asymptotic probability 1 by 6.4.8 with $k = 1$. Thus, when we look at the sequence defined by $\mu_n(\Phi_{1/2})$:

$$0, 0, \mu_2(\Phi_{1/2}), 0, \mu_4(\Phi_{1/2}), 0, \mu_6(\Phi_{1/2}), \ldots$$

we can see, that it oscillates between 0 and the probability $\mu_n(\Phi_{1/2})$ for even n:s, which as a subsequence tends to 1. Thus $\mu_n(\Phi_{1/2})$ does not converge to a limit.

Case $\Phi_{1/2^m}$, $m \in \mathbb{N}$: The idea is the same. We split the universe of the graph into 2^m disjoint sets of size exactly $|V|/2^m$. Clearly if the graph is not divisible by 2^m, this cannot be done. The straightforward way of saying that there exists m elements which partition the universe into exactly 2^m sets of size $|V^G|/2^m$ does not seem to work. The probability of finding such m elements decreases too fast when m grows. This can be remedied by actually looking for $2^m - 1$ elements, which partition the universe into 2^m equally sized sets. The sentence gets the following form:

$$\Phi_{1/2^m} = \exists x_1 (\bigwedge_{i_1=0}^{1} \exists x_2 (\bigwedge_{i_2=0}^{1} \exists x_3 (\bigwedge_{i_3=0}^{1} \ldots \exists x_m (\bigwedge_{i_m=0}^{1} \exists^{\geq 1/2^m} y \phi_{\bar{i}}(\bar{x}, y))) \ldots).$$

The sentence always branches twice after each existential quantifier. Each branch of the sentence is characterized by a sequence $\bar{\imath} = (i_1, i_2, i_3, \ldots, i_m)$ in which the value of i_j is determined in the conjunction after the $j : th$ existential quantification in $\Phi_{1/2^m}$. Thus i_j is 1, if that branch "turns to right" at the j:th conjunction and 0 if it "turns to left".

At the end of each branch we say that there are at least $|V^G|/2^m$ many elements satisfying $\phi_{\bar{\imath}} = \bigwedge_{j=1}^{m} \phi_{i_j}$, which is a conjunction of atomic and negated $\{E\}$-formulas defined in the following way:

- $\phi_{i_j} =: E(x_j y)$, if $i(j) = 1$.
- $\phi_{i_j} =: \neg E(x_j y)$, if $i(j) = 0$.

We estimate the probability $\mu_n(\Phi_{1/2^m})$ as a product of the probabilities of each split.

1. $\exists x_1$: We look for $v_1 \in V^G$, such that $deg(v) = |V^G|/2$. By 6.4.8, the probability of finding such v tends to 1 in even graphs when the size of the graph tends to infinity.

2. $\exists x_2$: We look $v_2 \in V^G$, such that the degree of v_2 in the set $\{v \in V^G \mid (v_1, v) \in E^G\}$ is exactly $|\{v \in V^G \mid (v_1, v_2) \in E^G\}|/2 = |V^G|/4$. Also, we look for a $v_3 \in V^G$, such that the degree of v_3 in the set $\{v \in V^G \mid (v_1, v) \notin E^G\}$ is exactly $|\{v \in V^G \mid (v_1, v) \in E^G\}|/2 = |V^G|/4$. By, 6.4.8 the probability of finding such v_2 tends to 1 when the size of the graph tends to infinity as well as finding such v_3.

3. $\exists x_m$: Suppose we have already split the graph into 2^{m-1} sets. Now, we look for a 2^{m-1} new elements, one for each set, to split the set into two halves of equal size. Again, by 6.4.8 the probability of finding a splitting element for each set tends to 1 as the size of the graph tends to infinity.

Now the probability of the sentence $\Phi_{1/2^m}$ can be estimated as the product of the probabilities of each split to be successful. Since $m \in \mathbb{N}$ is fixed we get that also the product of these probabilities tends to 1 in even domains as as the size tends to infinity. On the other hand, in domains odd cardinality the limit probability is 0. Thus $\mu_n(\Phi_{1/2^m})$ is not defined.

Case $\Phi_{s/2^m}$. We will change the sentence $\Phi_{1/2^m}$ so that the set characterized at the end of each branch of the formula are of size $s/2^m$. We do it in the following way. First note that each reduced[2] rational number $s/2^m$ has a unique representation in the following form:

$$s/2^m = c_1 \cdot 1/2^1 + c_2 \cdot 1/2^2 + \ldots + c_m \cdot 1/2^m,$$

[2] 2^m and s have no common factors

where $c(j)$ is either 0 or 1 for $j \leq m$. Note that $\Sigma_{c_i=1} 2^{m-i} = s$. We denote the sequence $\bar{c} = (c_1, \ldots, c_m)$ as the characteristic sequence of $s/2^m$. Recall that each branch of the formula was characterized by a sequence $\bar{i}$ of 0s and 1s.

Now for each $c_j \in \bar{c}$, such that $c(j) = 1$ we create a sequence $\bar{ij}$ in the following way:

1. $\bar{ij}(l) = \bar{i}(l)$, when $\quad l < j$.
2. $\bar{ij}(j) = 0 \Leftrightarrow \bar{i}(j) = 1$.

Lets consider the following case as an example: The rational 15/16 has a presentation:

$$15/16 = 1 \cdot 8/16 + 1 \cdot 4/16 + 1 \cdot 2/16 + 1 \cdot 1/16.$$

Thus the characteristic sequence of 15/16 is $(1, 1, 1, 1)$. Let us consider then a branch $(1, 1, 0, 0)$ in the sentence $\Phi_{1/16}$. For each $c_j \in \bar{c}$ we construct a sequence $\bar{ij}$ as defined earlier by the conditions 1) and 2): $\bar{i1} = (0)$, $\bar{i2} = (1, 0)$, $\bar{i3} = (1, 1, 1)$ and $\bar{i4} = (1, 1, 0, 1)$. Thus the characteristic formulas are the following:

$$\begin{aligned}
\phi_{\bar{i1}} &= : \neg E x_1 y, \\
\phi_{\bar{i2}} &= : E x_1 y \wedge \neg E x_2 y, \\
\phi_{\bar{i3}} &= : E x_1 y \wedge E x_2 y \wedge E x_3 y, \\
\phi_{\bar{i4}} &= : E x_1 y \wedge E x_2 y \wedge \neg E x_3 y \wedge E x_4 y.
\end{aligned}$$

Now the characteristic formula for the branch $(1, 1, 0, 0)$ will be a disjunction of the formulas $\phi_{\bar{ij}}$, $j \leq 4$.

$$\Psi_{(1,1,0,0)} =: \bigvee_{j=1}^{4} \phi_{\bar{ij}}.$$

In this way we change the characteristic formula of each branch. Now $\Phi_{s/2^m}$ gets the following form:

$$\Phi_{s/2^m} = \exists x_1 (\bigwedge_{i_1=0}^{1} \exists x_2 (\bigwedge_{i_2=0}^{1} \exists x_3 (\bigwedge_{i_3=0}^{1} \ldots \exists x_m (\bigwedge_{i_m=0}^{1} \exists^{\geq s/t} y (\Psi_{\bar{i}}(\bar{x}, y))) \ldots))).$$

Let us first show that $\Phi_{1/2^m} \Rightarrow \Phi_{s/2^m}$: Suppose $\mathcal{G} \models \Phi_{1/2^m}$. Then there are $a_1, a_2, a_3, \ldots, a_{2^m-1} \in V^G$ such that they split the graph into 2^m disjoint sets. Let

$$\begin{aligned}
\phi_{\bar{a}}^{\bar{i},G} &= \{a \in V^G \mid (G, \bar{a}, a) \models \phi_{\bar{i}}(\bar{x}, y))\}. \\
\Psi_{\bar{a}}^{\bar{i},G} &= \{a \in V^G \mid (G, \bar{a}, a) \models \Psi_{\bar{i}}(\bar{x}, y))\}.
\end{aligned}$$

Let $\bar{i} = (i_1, \ldots, i_m)$ be a characteristic sequence of a branch in the sentence $\Phi_{1/2^m}$ and let $\bar{c} = (c_1, \ldots, c_m)$ denote the characteristic sequence of $s/2^m$. We will show that each $\Psi_{\bar{a}}^{\bar{i},G}$ is a union of s different $\phi_{\bar{a}}^{\bar{i},G}$.

When we construct the characteristic formula $\Psi_{\bar{i}}(\bar{x}, y)))$ from $\bar{i}$ we go through all the $c_j \in \bar{c}$, such that $c_j = 1$. For each $c_j = 1 \in \bar{c}$ the new j-sequence $\bar{i_j} = (i_1, \ldots, |1 - i_j|)$ corresponds to all the m-sequences of the following form:

$$(i_1, \ldots, |1 - i_j|, *, \ldots, *),$$

where * means that the parameter can be either 0 or 1. One can observe that the first j parameters are fixed, but the last $(m - j)$ parameters can be either 0 or 1. Thus, for each $c_j = 1$ the new sequence $\bar{ij}$ corresponds to 2^{m-j} different m-sequences. Now it follows from the definition of the characteristic sequence of $s/2^m$ that

$$\Sigma_{c_j=1} 2^{m-j} = s.$$

Thus it holds that $|\Psi_{\bar{a}}^{\bar{i},G}| \geq s/2^m \cdot |V^G|$.

Suppose the cardinality of $\mathcal{G}$ is not divisible by 2^m, but still the sentence $\phi_{s/2^m}$ is true in $\mathcal{G}$. Then there is a sequence $(a_1, \ldots, a_{2^m})$ of vertices of $\mathcal{G}$, such that for each binary sequence $(i_1, \ldots, i_m)$ it holds:

$$|\Psi_{\bar{a}}^{\bar{i},G}| \geq |V^G| \cdot s/2^m.$$

Since $|V^G$ is not divisible by 2^m it holds actually that

$$|\Psi_{\bar{a}}^{\bar{i},G}| > |V^G| \cdot s/2^m \tag{6.8}$$

We already showed earlier that each $\Psi_{\bar{a}}^{\bar{i},G}$ is a union of s different $\phi_{\bar{a}}^{\bar{i},G}$. Since these sets are all disjoint, it means that each element is contained in exactly s different $\Psi_{\bar{a}}^{\bar{i},G}$. Thus

$$\Sigma_{\bar{i}} |\Psi_{\bar{a}}^{\bar{i},G}| = s \cdot |V^G|.$$

But by (6.8) it holds that

$$\Sigma_{\bar{i}} |\Psi_{\bar{a}}^{\bar{i},G}| > 2^m \cdot |V^G| \cdot s/2^m > s \cdot |V^G|,$$

which is a contradiction. □

Now we can give the following characterization for the 0-1 law of the logic $\mathcal{L}_{\infty\omega}^k(\exists^{\geq s/t})$.

Theorem 6.4.10. *The logic $\mathcal{L}_{\infty\omega}^k(\exists^{\geq s/t})$ has the* 0-1 *law if and only if one of the conditions is met :*

1. $t \neq 2^m$ *for all* $m \in \mathbb{N}$.
2. $k \leq \log_2 t$.

Proof. When 1 or 2 holds, see Theorem 6.4 When neither 1, nor 2 holds, see Theorem 6.4.9. □

Chapter 7

Future work

Gödel's completeness theorem establishes a perfect match between the truth and provability in first-order logic. This means that there is a mechanical way of generating precisely those first-order sentences that are true in all models, i.e the set of valid FO-sentences is recursively enumerable. Such perfect match between semantics and syntax is generally not possible for extensions of first-order logic like dependence logic. This is an immediate consequence of the high expressive power of these logics. For example, one can express in $\mathcal{D}$ that the domain of the structure is infinite:

$$\Phi : \exists x_4 \forall x_0 \exists x_1 \forall x_2 \exists x_3 (=(x_2, x_3) \wedge \not\approx x_1 x_4 \wedge (\approx x_0 x_2 \leftrightarrow \approx x_1 x_3)) \tag{7.1}$$

The sentence (7.1) asserts the existence of a bijective mapping between the domain and a proper subset of the domain. This is only possible if the domain is infinite (for more details see [39]). Now for all FO-sentences ψ, the fact that ψ is true in all finite structures is equivalent to the fact that $\mathcal{D}$-sentence $\Phi \vee \psi$ is valid. If $\mathcal{D}$ would be axiomatizable it would yield a recursive enumeration of valid FO-sentences over finite structures. This contradicts Trakhtenbrot's theorem, which states that the set of valid FO-sentences is not recursively enumerable over finite structures.

As the axiomatization of the whole $\mathcal{D}$ is not possible, we look for fragments of $\mathcal{D}$ which are potentially axiomatizable. Naturally, we turn our attention to the fragment of coherent formulas, which indeed could have an axiomatization as the whole fragment is contained in FO.

We will start off by characterizing the consequence relation of the dependence atoms. Some connected work has been done in the context of functional dependencies by Armstrong [1] in database theory, which we will use as our starting point. Armstrong gave a finite set of axioms and inference rules and showed that a given set of functional dependencies Σ entails a functional dependence if and only if it can be derived from Σ by using Armstrong's axioms. The Armstrong's axioms can be adopted for dependence atoms and with few additional rules they are complete for dependence atoms. The set of dependence atoms extended with identities and conjunction and negation can be also axiomatized without bigger problems. However, when we try to add disjunctions it becomes more complicated. We propose inference rules for disjunctions of dependence atoms and conjecture that the axiomatization is sound and complete.

Another interesting concept in the context of functional dependencies is so-called *Armstrong relation*. Armstrong relations are canonical models which arise naturally in proofs in database theory. Given a set of functional dependencies Σ, an Armstrong relation for Σ is a relation R, which satisfies exactly the functional dependencies that logically follow from Σ and fails all other functional dependencies. Armstrong relations exist for any given set of functional dependencies [1]. We are interested in the existence of Armstrong relations for fragments of dependence logic.

Another possible direction for future work is the topic of other dependence-type properties. In the natural language there are several similar concepts and expressions to "dependence", e.q. "totally determines", "function of", "mutual dependence", "liable to", etc., which one could try to formalize. We will give an example what we call "mutual dependence": Two variables x and y are mutually dependent if it holds that x determines functionally the value of y and y determines functionally the value of x. We will show that this simplest case has a complete axiomatization.

7.1 Functional dependencies

Functional dependencies play a important part in database designing as well as in database normalization and denormalization in data base theory. The most common model for a database is the *relational model* [5]. The model describes a database by a collection of predicates over a finite set of predicate variables. Functional dependencies are expressions of the form

$$X \to Y, \tag{7.2}$$

where X and Y are set of variables. If R is a relation over a set of variables U, such that $X, Y \subseteq U$, then R is said to obey the functional dependence (7.2), if for all tuples $s, s' \in R$, if the tuples agree on the variables in X, then they also agree on the variables in Y.

As one can observe, the relationship between the relation R and a functional dependence $(X \to Y)$ is almost identical to that of a team $\mathcal{X}$ and a dependence atom $=(x, y)$. Functional dependence like in the equation (7.2) can be expressed in dependence logic with the following formula:

$$\bigwedge_{y \in Y} =(X, y) \tag{7.3}$$

7.1.1 Armstrong's axioms

In this section we will introduce the Armstrong's axioms and go trough the completeness proof [1].

Definition 7.1.1. (*Armstrong's axioms*) Let U be a set of variables and X, Y, $Z \subseteq U$.

1. (Reflexivity): If $Y \subseteq X$, then $X \to Y$.
2. (Augmentation): If $X \to Y$, then $XZ \to YZ$.
3. (Transitivity): If $X \to Y$ and $Y \to Z$, then $X \to Z$.

Given a set of functional dependencies Σ, we write $\Sigma \models X \to Y$ when all the models of Σ are also models of $X \to Y$. We write$\Sigma \vdash X \to Y$, when $X \to Y$ can be derived from Σ using the Armstrong's axioms.

Proposition 7.1.2. *(Soundness) If $\Sigma \vdash X \to Y$, then $\Sigma \models X \to Y$.*

Proof. Easy □

Lemma 7.1.3. *If $X \to Y$ and $X \to Z$ hold, then also $X \to YZ$ holds.*

Proof. Easy □

The following closure set is used to show the completeness of the axioms.

Definition 7.1.4. Suppose Σ is a set of functional dependencies over a set of variables U and $X \subseteq U$. The closure of X with respect to Σ, X^*, is the set of all attributes $y \in U$, such that $X \to y$ can be derived from Σ using the Armstrong axioms.

Lemma 7.1.5. *Suppose Σ is a set of functional dependencies over a set of variables U and $X, Y \subseteq U$. Then the following are equivalent:*

1. $\Sigma \vdash X \to Y$.
2. $Y \subseteq X^*$.

Proof. Suppose $\Sigma \vdash X \to Y$ and $y \in Y$. Then by (A1) it holds that $\Sigma \vdash Y \to \{y\}$. Further, by (A3) holds $\Sigma \vdash X \to \{y\}$. Thus $Y \subseteq X^*$.

Suppose $Y \subseteq X^*$. Then $X \to \{y\}$ holds for every $y \in Y$. Then, by 7.1.3 it holds that $T \vdash X \to Y$. □

We will give the proof for completeness of the Armstrong'a axioms [1].

Proposition 7.1.6. *(Completeness) Suppose Σ is a set of functional dependencies over a set of variables U. If $\Sigma \models X \to Y$, then $\Sigma \vdash X \to Y$.*

$(X \subseteq) X^*$	$(y \in) U \setminus X^*$
$1\,1\,1 \ldots 1\,1\,1$	$0\,0\,0 \ldots 0\,0\,0$
$1\,1\,1 \ldots 1\,1\,1$	$1\,1\,1 \ldots 1\,1\,1$

Table 7.1: R

Proof. Suppose $\Sigma \models X \to Y$ but $\Sigma \not\vdash X \to Y$. By 7.1.5 it holds that $Y \subsetneq X^*$. Thus, there is $y \in Y$, such that $y \notin X^*$. Now let R be the database in table 7.1.

We will show that R satisfies each functional dependence in Σ, but fails $X \to Y$. First, it is easy to see that $R \not\models X \to Y$ holds, since both rows agree on X but disagree on Y.

Suppose $Z \to W \in \Sigma$. If $Z \subseteq X^*$, then by 7.1.5 it holds that $\Sigma \vdash X \to Z$. By (A3) $\Sigma \vdash X \to W$ holds. Then $W \subseteq X^*$ by definition of X^*. Thus, both of the rows in R agree on W. Thus $R \models Z \to W$ holds.

Suppose $Z \nsubseteq X^*$. Then both of the rows in R disagree on Z and again it follows that $R \models Z \to W$ holds. □

In context of functional dependencies there has been interest in canonical structures called Armstrong relations. Given a set of functional dependencies Σ, an Armstrong relation is a relation, which satisfies exactly the functional dependencies which logically follow from Σ and fail all the others functional dependencies. It was shown by Armstrong that these structures exist for any given set of functional dependenciesm[1].

Definition 7.1.7. Suppose Σ is a set of functional dependencies over attribute set U and Σ^* is the set of all functional dependencies that can be inferred from Σ by using the Armstrong axioms. An *Armstrong relation for* Σ is a relation R^Σ, such that the following hold:

- $R \models \phi$, for all $\phi \in \Sigma$.
- $R \not\models \phi$, for all $\phi \notin \Sigma^*$.

Theorem 7.1.8. *[1]An Armstrong relation R^Σ exists for each set of functional dependencies Σ.*

7.2 Dependence atoms

We would like to find deductive system for as large fragment of dependence logic as possible. Before even considering quantifiers we would like to characterize the logical consequence of atomic level expressions of $\mathcal{D}$. It turns out that the Armstrong's axioms work well for dependence atoms and when extending this with identity, conjunction and negation the completeness can be shown analogous to the Armstrong's completeness proof.

Definition 7.2.1. Let $\mathcal{D}_1$ be the following set of $\mathcal{D}$-formulas:

- $=(x_{i_0}, \ldots, x_{i_k}) \in \mathcal{D}_1$ for all $i_0, \ldots, i_k \in \mathbb{N}$.
- $\approx x_i x_j \in \mathcal{D}_1$ for all $i, j \in \mathbb{N}$.
- $\neg \approx x_i x_j \in \mathcal{D}_1$ for all $i, j \in \mathbb{N}$.
- If $\phi, \psi \in \mathcal{D}_1$, then $(\phi \wedge \psi) \in \mathcal{D}_1$.

We will denote X, Y, $Z \ldots$ for sets of variables and $x, y, z, \ldots$ for single variables. We use the set notation for dependence atoms; we write $=(X, y)$ instead of $=(x_1, \ldots, x_k, y)$ as the ordering of X does not have any effect on the satisfiability or provability of $=(X, y)$.

Definition 7.2.2. We propose the following rules for dependence atoms:

1. $=(X, x)$, for all $x \in X$.
2. If $X \subseteq Y$ and $=(X, z)$, then $=(Y, z)$.
3. If $=(X, y)$, then $=(\pi(X), y)$ for all permutations $\pi : X \to X$.
4. If $i < k$, $=(x_1, \ldots, x_n, y_i)$ and $=(y_1, \ldots, y_k)$, then $=(y_1, \ldots y_{i-1}, x_1, \ldots, x_n, y_{i+1}, \ldots y_k)$.
5. If $=(X, y, y, z)$, then $=(X, y, z)$.

It is quite straightforward to show that these rules are all sound. The completeness of the rules for dependence atoms can be shown analogously to the Armstrong's axioms. When we allow identity formulas and negation we extend the rules in the following way:

1. Identity relation $\approx$ is reflexive, symmetric and transitive.
2. Substitution: If $\phi(x)$ and $\approx xy$, then $\phi(x/y)$, where x/y means that some occurrences of x are replaced by y in ϕ.
3. If $\phi \wedge \theta$, then θ.
4. If $\phi \wedge \theta$, then ϕ
5. If ϕ and θ, then $\phi \wedge \theta$.
6. If $\approx xy$ and $\neg \approx xy$, then $\perp$.

The proof of the completeness of this axiomatization is again analogous to that of the Armstrong's axioms.

We can show that the armstrong relation can be constructed for any set dependence atoms and identity formulas.

7.3 Disjunction

The disjunction in dependence logic is more complex than the disjunction in classical first order logic. Notice that if ϕ and ψ are both FO-equivalent formulas, then the disjunction $\phi \vee \psi$ is just the classical disjunction. However, when we take disjunctions over non-FO-formulas, for example over dependence atoms, it is more complex. Recall, that a disjunction of two formulas is satisfied by a team if and only if the team can be divided into two sub-teams, such that one of the sub-teams satisfies the first disjunct and the other sub-team satisfies the second disjunct. Now some intuitively clear entailments which hold with the classical disjunction fail when it comes to the disjunction in dependence logic.

Example 7.3.1. $=(x, y) \vee =(x, y)$ does not imply $=(x, y)$: One can observe that the following team as in table 7.2 satisfies $=(x, y) \vee =(x, y)$, since $\{s_0\} \models =(x, y)$ and $\{s'\} \models =(x, y)$ hold, but the whole team does not satisfy $=(x, y)$.

assignment	x	y
s	0	0
s'	0	1

Table 7.2:

Definition 7.3.2. Let $\mathcal{D}_2$ be defined the following way:

- $=(x_{i_0}, \ldots, x_{i_k}) \in \mathcal{D}_2$ for all $i_0, \ldots, i_k \in \mathbb{N}$.
- If $\phi, \psi \in \mathcal{D}_2$, then $(\phi \vee \psi) \in \mathcal{D}_2$.

Definition 7.3.3. Suppose ϕ, θ, $\psi \in \mathcal{D}_2$. We propose the following inference rules for disjunctions of dependence atoms:

E1. If $\phi \vee \theta$, then $\theta \vee \phi$.

E2. If ϕ, then $\phi \vee \theta$.

E3. If $\phi \vdash \theta$ with the rules 7.2.2+$E1$+$E2$, and $\phi \vee \psi$, then $\theta \vee \psi$.

Showing the soundness of these rules is quite straightforward. The completeness on the other hand is not. For each given set of formulas T and a formula ϕ, which is not derivable from T, we construct a relation which satisfies all formulas in T, but fail ϕ. The problem comes with constructing these relations. In the cases of dependence atoms and identities it was enough to consider just two tuple relations. With disjunctions this is not anymore he case. We are at the moment able to show completeness only in certain restricted cases when we restrict the size of the formulas and allow only one distinct dependence atom to appear in a formula.

Conjecture 1. The rules 7.3.3 with the rules 7.2.2 are complete for disjunctions of dependence atoms.

Next theorem states that dependence atoms that follow logically from a set $T \subseteq \mathcal{D}_2$ follow already from the set of dependence atoms in T.

Theorem 7.3.4. *Suppose $T \subseteq \mathcal{D}_2$, T_a is the set of dependence atoms in T and ϕ a dependence atom. Then the following are equivalent:*

1. $T \models \phi$
2. $T_a \vdash \phi$

Proof. Idea of the proof: Suppose that there is a $T \subset \mathcal{D}_2$ and a dependence atom ϕ such that $T \models \phi$ holds, but $T \vdash \phi$ does not hold. Then we can construct a two row relation $R_{T,\phi}$ (like in Theorem 7.1.6), which satisfies all dependence atoms in T and fails ϕ. Since $R_{T,\phi}$ has only two rows it trivially satisfies all the disjunctions of dependence atoms, thus is a model for T. Then ϕ can not logically follow from T. $\square$

We consider that this result supports our conjecture 1. The intuition behind conjecture 1. is that every logical consequence of a disjunction of dependence atoms is actually obtained by combining the consequences of the disjuncts. This is essentially what the rule E3 says. As we stated in Theorem 7.3.4, this indeed holds for the dependence atoms. All the functional dependencies that logically follow from a given set of disjunctions of dependence atoms T, actually follow from the dependence atoms in T. Thus, no combination of disjunctions of dependence atoms can generate a new dependence. It would seem natural that similar law would hold not just in the atomic level, but also in the case of disjunctions.

The Armstrong relations do not exists in general when we allow disjunctions of dependence atoms.

Proposition 7.3.5. *Let T be the following set of formulas: $\{=(x) \vee =(x), =(x,y)\}$. Then, there is no Armstrong relation for T.*

Proof. The formula $=(x) \vee =(x)$ implies that x gets at most two different values. Furthermore, since $=(x,y)$ holds, also y gets at most two different values. Thus, there are three different "types" of teams, which satisfy T: tables 7.3, 7.4 and 7.5.

assignment	x	y
s_0	0	0
s_0	0	0

Table 7.3:

assignment	x	y
s_0	0	0
s_1	1	0

Table 7.4:

assignment	x	y
s_0	0	0
s_1	1	1

Table 7.5:

One can observe that each of the three types of relations satisfy T. Also, that all the three types satisfy either $=(y)$ or $=(y,x)$. Relation types 7.3 and 7.5 satisfy $=(y,x)$ and 7.4 satisfies $=(y)$. If we would have an armstrong relation for T it would be one of these types. Thus it would satisfy one of the formulas $=(y)$ or $=(y,x)$. But neither of these formulas can be derived from T. $\square$

7.4 Example: Mutual dependence

Another possible direction for future work is the topic of other dependence-type properties. In fact, database theory recognizes several different types of dependence, e.q. Multi valued dependence, which in contrast to the functional dependence, requires that certain tuples are present in a relation [11]. On the other hand one can consider "mutual dependence of variables" i.e. dependence in which several variables each determine each other. In the simplest case this means the mutual dependence of two variables x and y on each other: x determines functionally y and y determines functionally y. We will show that this simple case permits a complete axiomatization ala Armstrong:

Definition 7.4.1. $\{\sim\}$-formulas are formulas of form $\sim(x,y)$ over a set of variables Var. The semantics of the relation can be given in terms of $\mathcal{D}_1$-formula

$$X \models \sim(x,y) \Leftrightarrow X \models =(x,y) \wedge =(y,x) \tag{7.4}$$

Definition 7.4.2. (*Axioms*) The relation $\sim$ is an equivalence relation:

1. Symmetry; If $\sim(x,y)$, then $\sim(y,x)$.
2. Reflexivity; $\sim(x,x)$ for all x.
3. Transitivity; If $\sim(x,y)$ and $\sim(y,z)$, then $\sim(x,z)$.

We will show next that 7.4.2 are sound and complete for.

Theorem 7.4.3. *Suppose T is a set of $\{\sim\}$-formulas over a set of variables $Var(T)$. Then the following are equivalent:*

1. $T \models \sim(x, y)$.
2. $T \vdash \sim(x, y)$.

Proof. Soundness follows immediately from the semantics of the conjunction and dependence atoms.

Completeness: Suppose $T \models \sim(x, y)$, but $T \nvdash \sim(x, y)$.

We denote the equivalence classes of the binary relation $\sim_T$ defined on $Var(T)$ by condition $T \vdash \sim(x, y)$ with $[x]$. To show counter example, we just separate two classes $[x]$ and $[y]$. We join all the other classes with $[y]$. Now let $\mathcal{X}$ be the following team of table 7.6: Clearly, $\mathcal{X}$ fails $\sim(x, y)$ as $=\!(x, y)$ does not hold.

$[x]$	$[y]$
1 1 1 ... 1 1 1	0 0 0 ... 0 0 0
1 1 1 ... 1 1 1	1 1 1 ... 1 1 1

Table 7.6: Team $\mathcal{X}$.

Next we will show that $\mathcal{X}$ satisfies all formulas in T.

Suppose $\sim(z, v) \in T$, such that z and v are in different equivalence classes, say $z \in [x]$ and $v \in [y]$. Then, $\sim(x, z)$ and $\sim(v, y)$ can both be derived by symmetry. Then, by transitivity we can derive $\sim(x, y)$, which is a contradiction with the assumption that $T \nvdash \sim(x, y)$. Thus z and v are in a same class $[z]$. Then $\mathcal{X}$ satisfies $\sim(z, v)$. $\square$

The general case, where the mutually dependent variables are finite sets of first-order variables is also interesting. It should also allow axiomatization analogous to the Armstrong's proof, as it is definable with dependence atoms and conjunction.

Chapter 8

Conclusions

Dependence logic as a whole is a powerful and complex language. On finite models, which have been our main concern in this work, dependence logic covers all of non-deterministic polynomial time (NP). Indeed, one can consider it as an alternative language for NP. Thus any result on hierarchies or fragments of dependence logic have a potential of giving interesting information about NP.

We studied here relatively simple fragments of dependence logic. We showed that even in these simple fragments one finds NP-complete problems. Finding the borderline between PTIME and NP inside dependence logic seems like an interesting general problem. Ideally one would find large fragments with a dichotomy results, i.e. in this fragment every problem is either PTIME or that it contains NP-complete problems. Another concept that we used to draw watershed lines inside dependence logic is that of coherence. Again we found that the dividing line between coherence and incoherence can be met already in a relatively simple fragment of dependence logic. A third criterion we have used to find structure inside dependence logic is asymptotic probability and the 0-1-law.

In Chapter 3 we gave a characterization to the k-coherence of a formula. It turned out that universal quantification and conjunction preserve coherence while disjunction and existential quantification do not. Further, we showed that disjunction of two distinct dependence atoms is not coherent and that one can define this also with using existential quantifiers. We also showed that all coherent formulas are equivalent to first-order sentences when an additional relation symbol is given interpreting the team.

In Chapter 4 we studied the computational complexity of the model checking for quantifier-free dependence logic formulas. We gave a characterization for the complexity in terms of number of disjunctions and the coherence of the disjuncts. All quantifier-free formulas without disjunctions are coherent, and therefore equivalent to first order sentences. Thus they can be verified in LOGSPACE. When we allow one disjunction in the formulas, the model checking is NL. We also show that the model checking of the formula $=(x,y) \vee =(z,v)$ is complete for

NL. Finally, we showed that by allowing two or more disjunctions we can already express team properties, which are NP-complete.

We also gave some results on the combined complexity on the model checking for dependence atoms over all finite teams. We showed that the combined complexity of the model checking for dependence atoms over all finite teams is in LOGSPACE. If we allow disjunctions one disjucntion, we get that the combined complexity can be done in NL and in fact is complete for NL. Lastly, the combined complexity of the model checking will become NP-complete when we allow two or more disjunctions. The notable difference to the data complexity results was that here we considered disjunctions over a single formula.

In chapter 5, we studied the 0-1 law for dependence logic sentences in the normal form. We showed that the set on universal and existential sentences have the 0-1 law and that with sentences of prefix $\forall\forall\exists\exists$ already fail the 0-1 law. We also showed that all the quantifier-free formulas have the 0-1 law over uniform distribution of teams.

In Chapter 6, we studied the 0-1 law for $\mathcal{L}^k_{\infty\omega}(\exists^{s/t})$. We showed that a dichotomy holds for the 0-1 law of $\mathcal{L}^k_{\infty\omega}(\exists^{s/t})$ in terms of the number of variables in the language and the form of the quantifier.

In the last part of this thesis we described briefly a work in progress and a direction for future work. We discussed the problem of finding an axiomatization for fragments of dependence logic. We pointed out that dependence atoms can be axiomatized analogously to the functional dependencies and that when we consider disjunctions over dependence atoms we are not yet able to show the completeness of our axiomatization. We considered as an alternative direction for future work the study of other dependence-type properties. We considered one explicit example, which we called "mutual dependence". We showed that it allows sound and complete axiomatization.

Bibliography

[1] W.W. Armstrong, *Dependency Structures of Database Relationships*, in Proc. IFIP 74, North Holland, Amsterdam, 580-583, 1974.

[2] A. Blass and Y. Gurevich, *Henkin quantifiers and complete problems*, Annals of Pure and Applied Logic, 32:1–16, 1986

[3] A. Blass, Y. Gurevich, *New Zero-One Law and Strong Extension Axioms*, Bulletin of the European Association for Theoretical Computer Science ,72(October 2000), 103-122, 2000.

[4] K.L. Chung, P. Erdös, *On the application of the Borel-Cantelli lemma* Trans. Amer. Math. Soc. 72, 179-186, 1952.

[5] E.F. Codd, *Derivability, Redundancy, and Consistency of Relations Stored in Large Data Banks*, IBM Research Report, 1969.

[6] S. Cook, *A hierarchy for nondeterministic time complexity*, Proceedings of the fourth annual ACM symposium on Theory of computing, 187–192. 1972

[7] S. A. Cook, *The complexity of theorem-proving procedures*, Annual ACM Symposium on Theory of Computing, Shaker Heights, Ohio, United States, 151 - 158, 1971.

[8] H.B. Enderton, *Finite partially-ordered quantifiers*, Z. Math., 393-397, 1970.

[9] R. Fagin, *Generalized First-Order Spectra and Polynomial-Time Recognizable Sets*, Complexity of Computation, ed. R. Karp, SIAM-AMS Proceedings 7, 27-41, 1974.

[10] R. Fagin, *Probabilities on finite models*, J. Symbolic Logic 41 (1), 50-58, 1976.

[11] R. Fagin, *Multivalued dependencies and a new Normal form for Relational Databases*, ACM Transactions on Database Systems, Volume 2, Issue 3, 262 - 278, 1977.

[12] M.R. Garey, D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, New York: W.H. Freeman. ISBN 0-7167-1045-5, 1979.

[13] V. Geffert, *Space hierarchy theorem revised*, Theoretical Computer Science, volume 295, number 1-3, 171-187, 2003.

[14] G. Gottlob, P. G. Kolaitis, T. Schwentick *Existential second-order logic over graphs: Charting the tractability frontier*, Journal of the Association for Computing Machinery, 2000.

[15] Yu.V. Glebskii, D.I. Kogan, M.I.Liogon'kii, V.A. Talanov, *Range and degree of realizability of formulas in the restricted predicate calculus*, Kibernetika, 5, 17-28, 1969.

[16] E. Grädel, P. G. Kolaitis, L. Libkin, *Finite Model Theory And Its Applications* Springer, 2007.

[17] E. Grädel, *Capturing complexity classes by fragments of second order logic*, Structure in Complexity Theory Conference, Proceedings of the Sixth Annual Volume , Issue , 30 Jun-3 Jul 1991, 341 - 352, 1991.

[18] J. Hartmanis, R. E. Stearns, *On the computational complexity of algorithms*, Transactions of the American Mathematical Society 117: 285-306, 1965.

[19] L. Henkin, *Some Remarks on Infinitely Long Formulas*, Infinitistic Methods, Proceedings of the Symposium on Foundations of Mathematics, Warsaw, 167-183, 1961.

[20] J. Hintikka, *The principles of mathematics revisited*, Cambridge Uni- versity Press, Cambridge, 1996.

[21] J. Hintikka, G. Sandu, *Informational independence as a semantical phenomenon*, In Logic, methodology and philosophy of science, VIII (Moscow, 1987), volume 126 of Stud. Logic Found. Math., North-Holland, Amsterdam, 571-589 1989.

[22] W. Hodges, *Compositional semantics for a language of imperfect information.* Log. J. IGPL, 5(4):539-563 (electronic), 1997.

[23] N. Immerman, *Relational Queries Computable in Polynomial Time*, 14th ACM STOC Symp, 147-152, 1982.

[24] R. Kaila, *On Probabilistic elimination of generalized quantifiers*, Random Structures and Algorithms, 19, 1-36, 2001.

[25] R. M. Karp, *Reducibility Among Combinatorial Problems*, in R. E. Miller and J. W. Thatcher (editors). Complexity of Computer Computations. New York: Plenum. pp. 85-103, 1972.

[26] V.V Knyazev, *Zero-one law for an extension of first order predicate language*, Kibernetika, 2, 110-113, 1990.

[27] P. G. Kolaitis, M. Y. Vardi 0-1 *laws for Fragments of Existential Second-Order Logic: A Survey*, MFCS, 84-98, 2000.

[28] V. Nurmi, *Dependence Logic: Investigations into Higher Order Semantics defined on teams*, PhD thesis, University of Helsinki, 2009.

[29] P. G. Kolaitis, J. Väänänen, *Generalized quantifiers and pebble games on finite structures*, Annals of Pure and Applied Logic, 74, 23-75, 1995.

[30] P. G. Kolaitis, M.Y. Vardi, *0-1 law for infinitary logics*, in Proc. 5th IEEE Symp. on Logic in Computer Science, 156-167, 1990.

[31] J. Kontinen, J. Väänänen *On Definability in Dependence logic*, Journal of Logic, Language and Information, Volume 18, Nro. 3, 2009.

[32] P. Lindström, *First Order Predicate Logic with generalized quantifiers*, Theoria 32:186-195, 1966.

[33] C.H. Papadimitriou, *Computational Complexity*, Addison Wesley, 1993.

[34] W.J. Savitch, *Relationship between nondeterministic and deterministic tape classes*, Journal of Computer and System Sciences, 4:177-192, 1970.

[35] S. Shelah, *Choiceless polynomial time logic: inability to express*, Computer Science Logic 2000, Springer Lecture Notes in Computer Science 1862, 72-125, 2000.

[36] A. Tarski, *The concept of truth in the languages of the deductive sciences*" (Polish), *Prace Towarzystwa Naukowego Warszawskiego*, Wydzial III Nauk Matematyczno-Fizycznych 34, Warsaw, 1933; reprinted in Zygmunt 1995, pages 13-172; expanded English translation in Tarski, 152-278, 1983.

[37] M. Y. Vardi, *The complexity of relational query languages*, Annual ACM Symposium on Theory of Computing Proceedings of the fourteenth annual ACM symposium on Theory of computing., 137 - 146, 1982.

[38] M.Y. Vardi, *On the complexity of Bounded- Variable Queries*, Symposium on Principles of Database Systems, Proceedings of the fourteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems, 266 - 276, 1995.

[39] J. Väänänen, *Dependence Logic: A New Approach to Independence Friendly Logic*, Cambridge University Press, 2007.

[40] W. J. Walkoe Jr., *Finite partially-ordered quantification*, J. Symbolic Logic, 35, 535-555, 1970.

Samenvatting

In dit proefschrift bestuderen wij de eigenschappen van fragmenten van de afhankelijkheidslogica $\mathcal{D}$ [39] genomen over eindige structuren. Een belangrijk concept voor het maken van een onderscheid tussen formules is de zogenaamde k-coherentie van een formule. Het gelden van een k-coherente formule in alle teams kan worden teruggebracht tot het gelden in de k-element sub-teams. We beschrijven de coherentie van kwantorloze $\mathcal{D}$-formules, en we geven een voorbeeld van een formule die niet k-coherent is voor elke $k \in \mathbb{N}$. We laten zien dat alle coherente formules equivalent zijn aan eerste-orde zinnen wanneer een extra predicaat het team interpreteert.

We beschrijven de rekenkundige complexiteit van het testen van modellen voor $\mathcal{D}$-formules. Een klassiek voorbeeld in de beschrijvende complexiteitstheorie is Fagin's theorema [9], welke een verband aangeeft tussen Σ_1^1-formules en talen in NP. $\mathcal{D}$-formules hebben een definitie in Σ_1^1 en vice versa [39]. Gegeven Fagin's resultaat, betekent dit dat de eigenschappen die definieerbaar zijn in $\mathcal{D}$ over eindige structuren precies die eigenschappen zijn welke herkend worden in NP. We gebruiken het concept van coherentie om de rekenkundige complexiteit van het testen van de modellen voor $\mathcal{D}$-formules te kenmerken. We bepalen drie drempels in de rekenkundige complexiteit van het testen van modellen: 1) het testen kan plaatsvinden in logaritmische ruimte (L), 2) het kan plaatsvinden in niet-deterministische logaritmische ruimte (NL), 3) het testen is compleet voor niet-deterministische polynomiale tijd (NP). Voorts geven we complete gevallen voor NL en NP.

Een ander criterium dat we gebruiken om structuur te vinden in afhankelijkheidslogica is asymptotische waarschijnlijkheid en de 0-1-wet. We laten zien dat de 0-1-wet geldt voor universele en existentiële $\mathcal{D}$-zinnen en ook voor kwantorloze formules, gegeven de atomische waarschijnlijkheid van $1/2$.

In het tweede deel van het proefschrift kenmerken we de 0-1-wet voor proportionele kwantoren genomen over een uniforme distributie van eindige grafieken. We zullen een precieze drempel geven voor wanneer de 0-1 wet geldt voor $\mathcal{L}^k_{\infty\omega}(\exists^{s/t})$

en voor wanneer deze niet geldt.

Abstract

We study the properties of fragments of dependence logic $\mathcal{D}$ [39] over finite structures. One essential notion used to distinguish between $\mathcal{D}$-formulas is so-called k-coherence of a formula. Satisfaction of a k-coherent formula in all teams can be reduced to the satisfaction in the k-element sub-teams. We will characterize the coherence of quantifier-free $\mathcal{D}$-formulas and give an example of a formula which is not k-coherent for any $k \in \mathbb{N}$. We show that all coherent formulas are equivalent to first-order sentences when there is an extra predicate interpreting the team.

We also seek to characterize the computational complexity of model checking of $\mathcal{D}$-formulas. A classic example in the field of descriptive complexity theory is the Fagin's theorem [9], which establishes a perfect match between Σ_1^1-formulas and languages in NP. $\mathcal{D}$-formulas are known to have a definition in Σ_1^1 and vice versa [39]. When we combine this with Fagin's result we get that the properties definable in $\mathcal{D}$ over finite structures are exactly the ones recognized in NP.

We use the notion of coherence to give a characterization for the computational complexity of the model checking for $\mathcal{D}$-formulas. We establish three thresholds in the computational complexity of the model checking, namely when the model checking can be done in logarithmic space (L), in non-deterministic logarithmic space (NL) and when the checking becomes complete for non-deterministic polynomial time (NP). We give complete instances for NL and NP.

Another criterion we use to find structure inside dependence logic is asymptotic probability and the 0-1-law. We show that the 0-1-law holds for universal and existential $\mathcal{D}$-sentences as well as for all the quantifier-free formulas in the case of atomic probability 1/2.

In the second part of the thesis we give a characterization for the 0-1-law for proportional quantifiers over uniform distribution of finite graphs. We will give a precise threshold when the 0-1- law holds for $\mathcal{L}^k_{\infty\omega}(\exists^{s/t})$ and when it does not.

Titles in the ILLC Dissertation Series:

ILLC DS-2001-01: **Maria Aloni**
Quantification under Conceptual Covers

ILLC DS-2001-02: **Alexander van den Bosch**
Rationality in Discovery - a study of Logic, Cognition, Computation and Neuropharmacology

ILLC DS-2001-03: **Erik de Haas**
Logics For OO Information Systems: a Semantic Study of Object Orientation from a Categorial Substructural Perspective

ILLC DS-2001-04: **Rosalie Iemhoff**
Provability Logic and Admissible Rules

ILLC DS-2001-05: **Eva Hoogland**
Definability and Interpolation: Model-theoretic investigations

ILLC DS-2001-06: **Ronald de Wolf**
Quantum Computing and Communication Complexity

ILLC DS-2001-07: **Katsumi Sasaki**
Logics and Provability

ILLC DS-2001-08: **Allard Tamminga**
Belief Dynamics. (Epistemo)logical Investigations

ILLC DS-2001-09: **Gwen Kerdiles**
Saying It with Pictures: a Logical Landscape of Conceptual Graphs

ILLC DS-2001-10: **Marc Pauly**
Logic for Social Software

ILLC DS-2002-01: **Nikos Massios**
Decision-Theoretic Robotic Surveillance

ILLC DS-2002-02: **Marco Aiello**
Spatial Reasoning: Theory and Practice

ILLC DS-2002-03: **Yuri Engelhardt**
The Language of Graphics

ILLC DS-2002-04: **Willem Klaas van Dam**
On Quantum Computation Theory

ILLC DS-2002-05: **Rosella Gennari**
Mapping Inferences: Constraint Propagation and Diamond Satisfaction

ILLC DS-2002-06: **Ivar Vermeulen**
A Logical Approach to Competition in Industries

ILLC DS-2003-01: **Barteld Kooi**
Knowledge, chance, and change

ILLC DS-2003-02: **Elisabeth Catherine Brouwer**
Imagining Metaphors: Cognitive Representation in Interpretation and Understanding

ILLC DS-2003-03: **Juan Heguiabehere**
Building Logic Toolboxes

ILLC DS-2003-04: **Christof Monz**
From Document Retrieval to Question Answering

ILLC DS-2004-01: **Hein Philipp Röhrig**
Quantum Query Complexity and Distributed Computing

ILLC DS-2004-02: **Sebastian Brand**
Rule-based Constraint Propagation: Theory and Applications

ILLC DS-2004-03: **Boudewijn de Bruin**
Explaining Games. On the Logic of Game Theoretic Explanations

ILLC DS-2005-01: **Balder David ten Cate**
Model theory for extended modal languages

ILLC DS-2005-02: **Willem-Jan van Hoeve**
Operations Research Techniques in Constraint Programming

ILLC DS-2005-03: **Rosja Mastop**
What can you do? Imperative mood in Semantic Theory

ILLC DS-2005-04: **Anna Pilatova**
A User's Guide to Proper names: Their Pragmatics and Semanics

ILLC DS-2005-05: **Sieuwert van Otterloo**
A Strategic Analysis of Multi-agent Protocols

ILLC DS-2006-01: **Troy Lee**
Kolmogorov complexity and formula size lower bounds

ILLC DS-2006-02: **Nick Bezhanishvili**
Lattices of intermediate and cylindric modal logics

ILLC DS-2006-03: **Clemens Kupke**
Finitary coalgebraic logics

ILLC DS-2006-04: **Robert Špalek**
Quantum Algorithms, Lower Bounds, and Time-Space Tradeoffs

ILLC DS-2006-05: **Aline Honingh**
The Origin and Well-Formedness of Tonal Pitch Structures

ILLC DS-2006-06: **Merlijn Sevenster**
Branches of imperfect information: logic, games, and computation

ILLC DS-2006-07: **Marie Nilsenova**
Rises and Falls. Studies in the Semantics and Pragmatics of Intonation

ILLC DS-2006-08: **Darko Sarenac**
Products of Topological Modal Logics

ILLC DS-2007-01: **Rudi Cilibrasi**
Statistical Inference Through Data Compression

ILLC DS-2007-02: **Neta Spiro**
What contributes to the perception of musical phrases in western classical music?

ILLC DS-2007-03: **Darrin Hindsill**
It's a Process and an Event: Perspectives in Event Semantics

ILLC DS-2007-04: **Katrin Schulz**
Minimal Models in Semantics and Pragmatics: Free Choice, Exhaustivity, and Conditionals

ILLC DS-2007-05: **Yoav Seginer**
Learning Syntactic Structure

ILLC DS-2008-01: **Stephanie Wehner**
Cryptography in a Quantum World

ILLC DS-2008-02: **Fenrong Liu**
Changing for the Better: Preference Dynamics and Agent Diversity

ILLC DS-2008-03: **Olivier Roy**
Thinking before Acting: Intentions, Logic, Rational Choice

ILLC DS-2008-04: **Patrick Girard**
Modal Logic for Belief and Preference Change

ILLC DS-2008-05: **Erik Rietveld**
Unreflective Action: A Philosophical Contribution to Integrative Neuroscience

ILLC DS-2008-06: **Falk Unger**
Noise in Quantum and Classical Computation and Non-locality

ILLC DS-2008-07: **Steven de Rooij**
Minimum Description Length Model Selection: Problems and Extensions

ILLC DS-2008-08: **Fabrice Nauze**
Modality in Typological Perspective

ILLC DS-2008-09: **Floris Roelofsen**
Anaphora Resolved

ILLC DS-2008-10: **Marian Counihan**
Looking for logic in all the wrong places: an investigation of language, literacy and logic in reasoning

ILLC DS-2009-01: **Jakub Szymanik**
Quantifiers in TIME and SPACE. Computational Complexity of Generalized Quantifiers in Natural Language

ILLC DS-2009-02: **Hartmut Fitz**
Neural Syntax

ILLC DS-2009-03: **Brian Thomas Semmes**
A Game for the Borel Functions

ILLC DS-2009-04: **Sara L. Uckelman**
Modalities in Medieval Logic

ILLC DS-2009-05: **Andreas Witzel**
Knowledge and Games: Theory and Implementation

ILLC DS-2009-06: **Chantal Bax**
Subjectivity after Wittgenstein. Wittgenstein's embodied and embedded subject and the debate about the death of man.

ILLC DS-2009-07: **Kata Balogh**
Theme with Variations. A Context-based Analysis of Focus

ILLC DS-2009-08: **Tomohiro Hoshi**
Epistemic Dynamics and Protocol Information

ILLC DS-2009-09: **Olivia Ladinig**
Temporal expectations and their violations

ILLC DS-2009-10: **Tikitu de Jager**
"Now that you mention it, I wonder…": Awareness, Attention, Assumption

ILLC DS-2009-11: **Michael Franke**
Signal to Act: Game Theory in Pragmatics

ILLC DS-2009-12: **Joel Uckelman**
More Than the Sum of Its Parts: Compact Preference Representation Over Combinatorial Domains

ILLC DS-2009-13: **Stefan Bold**
Cardinals as Ultrapowers. A Canonical Measure Analysis under the Axiom of Determinacy.

ILLC DS-2010-01: **Reut Tsarfaty**
Relational-Realizational Parsing

ILLC DS-2010-02: **Jonathan Zvesper**
Playing with Information

ILLC DS-2010-03: **Cédric Dégremont**
The Temporal Mind. Observations on the logic of belief change in interactive systems

ILLC DS-2010-04: **Daisuke Ikegami**
Games in Set Theory and Logic