RWTH-Aachen University
Mathematical Foundations of Computer Science

Bachelor Thesis

# Decidability and Bisimulation for Logics of Functional Dependence

Phil Pützstück

supervised by

Prof. Dr. Erich Grädel

September 2020

# Abstract

Due to the importance of notions of dependence throughout all of science, various logics and logical frameworks for reasoning about dependence and independence have been proposed. Recently, Baltag and van Benthem introduced the logic LFD in [6] as a minimalistic "core" logic of functional dependence. It focuses on a local sense of dependence to retain classical semantics, endowing it with a modal character and most remarkably, decidability.

In this thesis, we aim to improve the understanding of LFD in connection with other logics, answering some of the open questions from [6]. Most notably, we solidify the modal viewpoint of LFD by defining a notion of bisimulation for which we show an analogue of van Benthem's Theorem, precisely capturing the range of LFD within first-order logic under various translations. We also compare its expressive power with guarded fragments of first-order logic, and relate LFD to a logic with team semantics. Lastly, we clarify the limits of LFD's decidability by proving the undecidability of a natural extension, and analyse the complexity of its satisfiability and model checking problem.

# Contents

# Chapter 1

# Introduction

## 1.1 Dependence in a Logical Context

As the phenomenon of dependence appears in many different areas of science and daily life, numerous different notions of dependence have been studied in mathematics and philosophy. Here we focus on its strongest form, namely functional dependence, where given variables deterministically determine other variables.

One of the earliest systematic studies of functional dependence was in the context of database theory by Armstrong [3], where the following basic structural properties emerged. Let $X, Y, Z$ be sets of variables.

- (Reflexivity) If $Y \subseteq X$, then $Y$ functionally depends on $X$.
- (Augmentation) If $Y$ functionally depends on $X$, then $Y \cup Z$ functionally depends on $X \cup Z$.
- (Transitivity) If $Z$ functionally depends on $Y$ and $Y$ functionally depends on $X$, then $Z$ functionally depends on $X$.

These are also called Armstrong's Axioms, and present a sound and complete way to infer all possible functional dependencies for a given set of variables. Hence this already represents a language in which we can reason about statements concerning functional dependence, albeit a very weak one. To obtain more expressive power, the idea of extending logics with the ability to talk about some form of dependence has come up.

A prominent example of extending classical first-order logic $\mathsf{FO}$ to talk about (functional) dependencies and independencies between its variables is independence-friendly logic (IF-logic), introduced by Hintikka and Sandu in [24]. The idea used there, commonly referred to as dependence between quantifiers, allows one to explicitly state on which other variables some quantified variable depends. As an example of how this can affect the meaning of sentences, consider the statement that some function $f\colon \mathbb{R} \to \mathbb{R}$ is continuous; for all $x \in \mathbb{R}$ and all $\varepsilon > 0$ there exists some $\delta > 0$ such that for all $y \in \mathbb{R}$ with $|x - y| < \delta$ we have $|f(x) - f(y)| < \varepsilon$. The quantification pattern is of the form

$$\forall x \forall \varepsilon \exists \delta \forall y \, \varphi(x, \varepsilon, \delta, y).$$

In the case of classical first-order semantics, the $\delta$ depends on $\varepsilon$ *and* $x$. But in IF-logic we could now express that $\delta$ only depends on $\varepsilon$, and not on $x$. This corresponds to uniform continuity, which is well known to be a stronger notion than usual continuity. In this case, uniform continuity can still be expressed by a first-order sentence with a different quantification pattern, but this is in general not possible for every formula or sentence of IF-logic. Indeed, even though the quantifiers in IF-logic are essentially first-order (ranging over elements in structures), it was shown that its expressive power coincides with that of existential second-order logic $\Sigma^1_1$.

Initially, the only known semantics for IF-logic were based on Skolem functions or games of imperfect information [31]. As a means to provide a compositional, model-theoretic semantics for IF-logic, the now called team semantics came to be, originating in Hodge's paper on compositional semantics for logics of imperfect information [26]. In team semantics, formulae are evaluated over *sets* of assignments, called teams. Together with the idea of Väänänen [34] to view dependencies as atomic properties of teams, rather than annotations of quantifiers, team semantics presents the fundamental framework of many modern logics of dependence, independence, and imperfect information, see e.g. [1].

A drawback of the approaches mentioned above is that most of them lead to inherently non-classical semantics. For example, logics based on team semantics are usually not closed under classical negation, and contain connectives or atoms that violate classical laws such as the law of the excluded middle.

The rather different approach of creating dependencies between variables that is used in [6] for LFD aims to retain most classical features of FO. It is based on generalized assignment semantics, which originates in a general relativization technique in algebraic logic, see e.g. [32]. Intuitively, we consider classical Tarski semantics for FO, but drop the assumption that all possible assignments of variables to values are available, thus inducing dependencies between variables via gaps in the assignment space. As a consequence, our classical structures $\mathfrak{A}$ now come together with a fixed set of variables $V$ and a fixed set of admissible assignments $T \subseteq A^V$ (also called team), where $A$ is the universe of $\mathfrak{A}$. Quantifiers are then restricted to yield assignments in this set $T$ instead of the usual full space of assignments $A^V$. Apart from this, formulae are still evaluated on a single "current" assignment, which allows us to have classical semantics for boolean connectives. Consequently, the logic focuses on a local sense of dependence by introducing atoms $D_X y$. When evaluated at some current assignment $s \in T$, these state that within the whole team, fixing the values of $X$ to their current ones (at $s$) implies fixing the value of $y$ to its current one (at $s$). We quote the authors' brief conclusion from [6, Section 1.3]:

> "The resulting logic of functional dependence LFD is quite expressive. While capturing the main properties of functional dependence, it retains all classical boolean operators with their standard semantics and laws; thus showing that *dependence is not an intrinsically non-classical phenomenon*. Moreover, LFD is remarkably simple and well-behaved, having transparent axiomatizations, with nice meta-properties [...]. As it will become clear, LFD is also a *modal* logic, with interesting connections with epistemic logics and inquisitive logics. Finally, LFD offers a platform for studying concrete notions of dependence in many fields."

In [6], a lot of proof-theoretic results were shown, including a complete proof-calculus for LFD. The modal perspective was introduced early, and many results were proven with both classical and modal techniques. For example, a proof of decidability via syntactic type models was given in [6, Section 4], a purely proof-theoretic one was developed in [6, Section 5], and finally in [6, Section 6] the decidability of LFD was also shown via a modal filtration argument. Apart from this, LFD is a fragment of first-order logic, and inherits properties such as compactness and recursive enumerability of its validities. Various extensions and applications of LFD in concrete settings were explored in [6, Sections 7, 8]. In this thesis we aim to answer some of the many open questions raised in said paper, improving the understanding of LFD – especially in the sense of its expressive power and relation to other logics – from a model-theoretic point of view.

## 1.2 Contributions & Structure

The original contributions and structure of this thesis can be summarised as follows. In Chapter 2 we formally define the syntax and semantics of LFD. We also consider a natural extension of LFD and prove its undecidability, answering an open question from [6, Section 7.2] and clarifying the gap between decidability and undecidability for logics of functional dependence.

The question for a "notion of bisimulation for LFD which captures its precise range within the first-order language over standard models" was raised in [6, Section 3.1]. We answer this in Chapters 3 and 4 as follows. In Chapter 3 we define a notion of bisimulation for LFD and relate it to logical equivalence via an analogue of the classical Ehrenfeucht-Fraïssé Theorem. We also give many examples demonstrating the limits of LFD's expressive power. Continuing in Chapter 4, we extensively cover possible first-order translations of LFD, and proceed to prove an analogue of van Benthem's Theorem in Section 4.1.5, characterising LFD as the LFD-bisimulation-invariant fragment of FO under reasonable first-order translations.

At the end of Chapter 4 we compare LFD to guarded fragments of FO and show how to relate it to well-known logics with team semantics.

Section 5.1 gives some basic complexity bounds of the satisfiability problem of LFD, and Section 5.2 characterises the complexity of the model checking problem for LFD.

Lastly, we discuss some ramifications of our results on the (still open) question of whether LFD has the finite model property in Chapter 6. Although technically not part of this thesis, we also want to mention the python library written by the author[1], which was used to find and minimize most of the examples in this thesis, and may be helpful to others.

## 1.3 Notation & Conventions

- Classical structures will be denoted by uppercase gothic letters such as $\mathfrak{A}, \mathfrak{B}, \mathfrak{C}$ and their corresponding universes by the corresponding latin letters $A, B, C$.
- We will write $\mathrm{ar}(R)$ for the arity of relation symbols $R$.
- Tuples of elements or variables will be written as bold lowercase letters such as $\mathbf{a}, \mathbf{b}, \mathbf{x}, \mathbf{y}$, and we denote the set of their components by enclosing them with brackets, i.e. $[\mathbf{a}] \coloneqq \{a \mid a \text{ occurs in some position of } \mathbf{a}\}$. If $s\colon V \to A$ and $\mathbf{x} = (x_1, \dots, x_n) \in V^n$, we will write $s(\mathbf{x}) \coloneqq (s(x_1), \dots, s(x_n)) \in A^n$.
- Throughout this thesis we will sometimes assume variable or symbols to be "fresh". What we mean by this is that said variable or symbol has not been used before in the respective context, so that no conflicting usage can occur.
- Disjoint unions are emphasized by using $\uplus$ instead of $\cup$.
- The class of all ordinals is denoted by **Ord**.

[1]https://git.rwth-aachen.de/philpuetzstueck/lfd-sat.

# Chapter 2

# LFD and Extensions

## 2.1 Syntax and Semantics of LFD

As we have seen in the introduction, the structures relevant for us come together with a fixed set of variables $V$ and a fixed set of admissible assignments (henceforth called the team), similar to the fixed vocabulary $\tau$ of classical structures.

**Definition 2.1** (Type)**.** Let $\tau$ be a relational vocabulary and $V$ a set of variables. We call $(\tau, V)$ a type. If $\tau$ and $V$ are finite, then we say $(\tau, V)$ is a finite type.

**Definition 2.2** (Dependence models, [6, Definition 2.1])**.** A dependence model $\mathbf{M}$ of type $(\tau, V)$ consists of a $\tau$-structure $\mathfrak{M}$ with universe $M$, together with a nonempty team $T \subseteq M^V$. We usually write $\mathbf{M} = (\mathfrak{M}, T)$. We also use $T_{\mathbf{M}}$ to refer to the team of $\mathbf{M}$. Pointed dependence models $\mathbf{M}, s$ distinguish a "current" assignment $s \in T_{\mathbf{M}}$. The class of all pointed $(\tau, V)$ dependence models is given by

$$\mathcal{DEP}[\tau, V] \coloneqq \{\mathbf{M}, s \mid \mathbf{M} \text{ is a } (\tau, V) \text{ dependence model and } s \in T_{\mathbf{M}}\}.$$

We will use bold capital letters such as $\mathbf{M}, \mathbf{N}$ for dependence models, $\mathfrak{M}, \mathfrak{N}$ for their underlying structures, and $M, N$ for their universes.

For relation symbols $R \in \tau$ we will also denote their interpretation by $R^{\mathbf{M}}$ instead of $R^{\mathfrak{M}}$. For a team $T$ with domain $V$, a variable $x \in V$ and a tuple $\mathbf{x}$ of variables in $V$ we use the notation

$$T(x) \coloneqq \{s(x) \mid s \in T\} \qquad \text{and} \qquad T(\mathbf{x}) \coloneqq \{s(\mathbf{x}) \mid s \in T\}.$$

In examples where $V$ is finite, it is convenient to fix some enumeration $\mathbf{v}$ of $V$ so we can denote assignments $s$ by their tuple of values $s(\mathbf{v})$. Conversely, given some fitting tuple of values $\mathbf{a}$, the notation $\mathbf{v} \mapsto \mathbf{a}$ represents an assignment $s$ with $s(\mathbf{v}) = \mathbf{a}$. This allows us to view $T$ as a relation or database, i.e. a table where each variable has its own column in the order of $\mathbf{v}$, and the rows are given by $s(\mathbf{v})$ for each $s \in T$.

**Example 2.3.** Consider $V = \{x, y, z\}$ with the order $\mathbf{v} = xyz$. If we have a team that contains the assignments

$$xyz \mapsto 000, \qquad xyz \mapsto 110, \qquad xyz \mapsto 121, \qquad xyz \mapsto 221,$$

then we can represent said team in a table such as

| $x$ | $y$ | $z$ |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 1 | 0 |
| 1 | 2 | 1 |
| 2 | 2 | 1 |

.

In this table, we can see that $y$ determines $z$ (or: $z$ functionally depends on $y$), because *fixing the value of $y$ fixes the value of $z$*. Indeed, fixing the value of $y$ to 0 or 1 corresponds to only considering a single row, whereas fixing $y$ to 2 corresponds to looking at the bottom two rows, in which $z$ has the constant value 1. At the same time we can see that in these two bottom rows, $x$ can still vary, so $x$ does not functionally depend on $y$.

As mentioned in the introduction and apparent from the definition of dependence models, we want to evaluate formulae with respect to a "current" assignment to retain classical boolean semantics. Hence, it also makes sense to consider the following *local* notion of dependence.

**Example 2.4** (Local functional dependence)**.** Consider again the table of Example 2.3. We say that $x$ locally depends on $y$ at the assignment $(1, 1, 0)$, because fixing $y$ to be its value *in this specific assignment*, namely 1, also fixes $x$ to its value in said assignment. As another example, we have that $y$ locally depends on $z$ at the assignment $(1, 2, 1)$ (or likewise at $(2, 2, 1)$), because $z = 1$ entails $y = 2$ in the regarded table.

Notice that in this context, usual functional dependence is just the universal version of this local dependence: $y$ depends on $X$ in the whole team iff $y$ locally depends on $X$ at every assignment in the team. For this reason, we also refer to usual functional dependence as *global* (functional) dependence.

To talk about this form of dependence, atoms $D_X y$ are introduced for sets of variables $X$ and variables $y$. They are read as "$X$ locally determines $y$" or "$y$ locally depends on $X$". It was also argued that from this local viewpoint, it is more natural to consider so-called dependence quantifiers $\mathsf{D}_X$ that are in a sense "dual" to the ones we know from first-order logic; intuitively, rather than letting the variables in $X$ vary freely, they fix the current values of $X$, and allow all other variables to vary. In other words, within the whole team, fixing the values of $X$ to the current ones fixes $\varphi$ to be true. We give the formal definitions of syntax and semantics below.

**Definition 2.5** (Syntax of LFD, [6, Definition 3.1])**.** For a type $(\tau, V)$ the syntax of formulae in $\mathsf{LFD}(\tau, V)$ is given by

$$\varphi \quad ::= \quad R\mathbf{x} \quad | \quad D_X y \quad | \quad \neg\varphi \quad | \quad \varphi \wedge \varphi \quad | \quad \mathsf{D}_X \, \varphi,$$

where $R \in \tau$ is a relation symbol, $\mathbf{x} \in V^{\mathrm{ar}(R)}$ is a tuple of variables of appropriate length, $X \subseteq V$ is *finite*, and $y \in V$.

**Notation 2.6** (Abbreviations of formulae, [6, p. 9])**.**

1. Boolean connectives $\varphi \vee \psi, \varphi \to \psi, \varphi \leftrightarrow \psi$ are defined as usual.
2. We use $D_{\mathbf{x}} y$ instead of $D_{[\mathbf{x}]} y$ and likewise for $\mathsf{D}_{\mathbf{x}} \, \varphi$.
3. We set $D_X Y := \bigwedge_{y \in Y} D_X y$ for finite $Y \subseteq V$, and also $D_X \mathbf{y} := D_X[\mathbf{y}]$.
4. The dual of $\mathsf{D}_X \, \varphi$ is defined as $\mathsf{E}_X \, \varphi := \neg \, \mathsf{D}_X \, \neg\varphi$.
5. For the special case $X = \varnothing$ we use $\mathsf{A} \, \varphi := \mathsf{D}_\varnothing \, \varphi$ and $\mathsf{E} \, \varphi := \mathsf{E}_\varnothing \, \varphi$.

Formulae of the form $R\mathbf{x}$ and $D_X y$ are the atoms of LFD. The former are called relational atoms, the latter dependence atoms. We refer to the $\mathsf{D}_X$ and $\mathsf{E}_X$ as dependence quantifiers or modalities of LFD, and also call $\mathsf{A}$ and $\mathsf{E}$ global modalities. As we will often deal with dependence on *sets* of assignments, the following will allow us to give cleaner definitions.

**Definition 2.7** (Agreement relation)**.** On some team $T$ with domain $V$ we define equivalence relations $=_X \subseteq T \times T$ for $X \subseteq V$ as follows: given $s, t \in T$ let

$$s =_X t \qquad \text{iff} \qquad s(x) = t(x) \text{ for all } x \in X.$$

In particular $=_\varnothing \; = T \times T$. We also write $s =_{\mathbf{x}} t$ instead of $s =_{[\mathbf{x}]} t$ for tuples of variables $\mathbf{x}$. The following notation will also be useful later on: let $s \,\overline{\cap}\, t$ refer to the inclusion-maximal set on which $s$ and $t$ agree, so

$$s \,\overline{\cap}\, t := \{x \in V \mid s =_x t\}.$$

**Definition 2.8** (Semantics of LFD,[6, Definition 3.2])**.** Consider a type $(\tau, V)$ and $\mathbf{M}, s \in \mathcal{DEP}[\tau, V]$. The satisfaction relation uses the usual semantics of boolean connectives together with

$$\begin{array}{lll} \mathbf{M}, s \models R\mathbf{x} & \text{iff} & s(\mathbf{x}) \in R^{\mathbf{M}}. \\ \mathbf{M}, s \models D_X y & \text{iff} & t =_X s \text{ implies } t =_y s \text{ for all } t \in T_{\mathbf{M}}. \\ \mathbf{M}, s \models \mathsf{D}_X \varphi & \text{iff} & \mathbf{M}, t \models \varphi \text{ for all } t \in T_{\mathbf{M}} \text{ with } t =_X s. \end{array}$$

Here $R, \mathbf{x}, X, y$ are as in Definition 2.5. Now $\mathsf{E}_X, \mathsf{E}, \mathsf{A}$ have the expected semantics:

$$\begin{array}{lll} \mathbf{M}, s \models \mathsf{E}_X \varphi & \text{iff} & \mathbf{M}, t \models \varphi \text{ for some } t \in T_{\mathbf{M}} \text{ with } t =_X s. \\ \mathbf{M}, s \models \mathsf{E} \varphi & \text{iff} & \mathbf{M}, t \models \varphi \text{ for some } t \in T_{\mathbf{M}}. \\ \mathbf{M}, s \models \mathsf{A} \varphi & \text{iff} & \mathbf{M}, t \models \varphi \text{ for all } t \in T_{\mathbf{M}}. \end{array}$$

The semantics given above formalize the *local* notion of dependence we introduced in Example 2.4. It is also this locality, together with the semantics of the dependence quantifiers $\mathsf{D}_X$ and $\mathsf{E}_X$, which further emphasizes the modal character we mentioned in the introduction. Indeed, notice the similarities to the modalities $\Box$ and $\Diamond$ of propositional modal logic ML; the binary relations $=_X$ on our teams can be viewed as the accessibility relations of the modality $\mathsf{D}_X$ and its dual $\mathsf{E}_X$ on the team. In this sense, a dependence model $\mathbf{M}$ induces a type of Kripke model that has $T_{\mathbf{M}}$ as its universe and the accessibility relations $(=_X)_{X \subseteq V}$. We formalize this idea in Section 4.1.2.

Also note that usual functional dependence is expressible in LFD via $\mathsf{A}\, D_X y$, which guarantees that $X$ locally determines $y$ at every assignment in the team, i.e. that $X$ determines $y$ in the whole team.

**Fact 2.9** ([6, Fact 2.4])**.** Inspired by Armstrong's Axioms, which were mentioned in the introduction, we obtain the following LFD-validities for our local dependence atoms. For every set of variables $V$ the following formulae hold at every pointed $(\varnothing, V)$ dependence model:

1. (Reflexivity) $D_X x$ for all (finite) $X \subseteq V$ and $x \in X$.
2. (Monotonicity) $D_X y \to D_Z y$ for all (finite) $Z \subseteq V$, $X \subseteq Z$, and $y \in V$.
3. (Transitivity) $(D_X Y \wedge D_Y Z) \to D_X Z$ for all (finite) $X, Y, Z \subseteq V$.

Moreover, [6, Proposition 2.5] gives an interesting representation theorem for relations with such properties. Specifically, for any relation $D \subseteq \mathcal{P} \times V$ which satisfies Reflexivity and Transitivity as above, there exists a dependence model where $D$ is the local dependence relation at all assignments.

**Definition 2.10** (Free variables, [6, Definition 3.3])**.** The free variables $\mathrm{Free}(\varphi)$ of a formula $\varphi \in \mathsf{LFD}$ are defined as usual for boolean connectives and relations, so

$$\mathrm{Free}(R\mathbf{x}) = [\mathbf{x}], \qquad \mathrm{Free}(\neg\varphi) = \mathrm{Free}(\varphi), \qquad \mathrm{Free}(\varphi \wedge \psi) = \mathrm{Free}(\varphi) \cup \mathrm{Free}(\psi),$$

together with

$$\mathrm{Free}(D_X y) = X \qquad \text{and} \qquad \mathrm{Free}(\mathsf{D}_X\, \varphi) = X.$$

We call $\varphi$ a sentence if $\mathrm{Free}(\varphi) = \varnothing$. In this case $\varphi$ must be a boolean combination of formulae of the form $\mathsf{A}\,\psi, \mathsf{E}\,\psi$ or $D_\varnothing y$. Clearly $\mathsf{A}\,\psi$ and $\mathsf{E}\,\psi$ do not require a current assignment to be evaluated. Moreover, $\mathbf{M}, s \models D_\varnothing y$ if and only if $t =_y s$ for all $t \in T_\mathbf{M}$ with $t =_\varnothing s$. Since $t =_\varnothing s$ is vacuously true for all pairs of assignments $(t, s)$, we see that the atoms $D_\varnothing y$ just state that $y$ is constant throughout the team. Hence, if $\varphi$ is a sentence, then we can simply write $\mathbf{M} \models \varphi$ whenever $\mathbf{M}, s \models \varphi$ for some (or equivalently, for all) $s \in T_\mathbf{M}$.

**Fact 2.11** (Locality, [6, Fact 3.4])**.** If $\mathrm{Free}(\varphi) \subseteq X$ and $s =_X t$ for $s, t \in T_\mathbf{M}$, then $\mathbf{M}, s \models \varphi$ iff $\mathbf{M}, t \models \varphi$. In the same fashion, if $\mathbf{M}, s \models \varphi$, then $\mathbf{M}, s$ can forget about variables and relations that do not occur in $\varphi$ and still satisfy $\varphi$. Hence, for all questions regarding formulae in $\mathsf{LFD}(\tau, V)$ it suffices to consider $(\tau, V)$ dependence models, even though $\varphi$ can also be evaluated on models of larger types.

Lastly we want to restate the following facts about LFD that were already mentioned in the introduction:

1. LFD is essentially a fragment of FO. The so-called standard translation of LFD into FO is introduced in [6, Section 3.2], and will be covered in depth in Section 4.1, where we also give two other first-order translations of LFD.

2. LFD is decidable. See e.g. [6, Theorem 4.11].

## 2.2 LFD with Functions and Explicit Equality

**LFD with functions.** In [6, Section 7.1] the extension of LFD with functions is introduced. As for FO, one regards 0-ary functions as constants, and defines terms $h$ inductively as variables $x$ or functions $f$ applied to a tuple of terms $\mathbf{h}$. One then allow sets of terms $H$ to replace sets of variables in the dependence formulae; the atom $D_H h$ states that the value of the term $h$ depends locally on the set of values of the terms in $H$, and likewise for $\mathsf{D}_H\, \varphi$. Agreement on sets of terms $s =_H t$ is then the basic notion, from which semantics of $D_H h$ and $\mathsf{D}_H\, \varphi$ is derived as for ordinary LFD (cf. Definition 2.8). The specifics are not important here.

In [6, Fact 7.3] it is shown that this extension of LFD retains the decidability of LFD. For this, the authors give a satisfiability-preserving translation of formulae in this functional extension to formulae in plain LFD. The translation encodes terms as a fresh variables. For example, given a unary function $f$ and a variable $x$ the term $fx$ is simulated by introducing a new variable $v_{fx}$ and requiring a global functional dependence of $v_{fx}$ on $x$ via $\mathbb{A} D_x v_{fx}$. The main reason we mentioned this extension is to give some perspective on decidable extensions of LFD, and to talk about the following interesting fact that is required for the proof of the reduction from LFD with functions to plain LFD.

**Fact 2.12** ([6, Fact 7.2])**.** For every dependence model $\mathbf{M}, s$ there exists an LFD-equivalent dependence model $\mathbf{N}, t$ such that

$$T_{\mathbf{N}}(x) \cap T_{\mathbf{N}}(y) = \varnothing, \qquad x, y \in V,\ x \neq y,$$

where $V$ denotes the domain of $T_{\mathbf{N}}$. In words, this states that if $x, y$ are distinct variables, then the set of values taken by $x$ is disjoint from the set of values taken by $y$. We call such models $\mathbf{N}$ *distinguished*.

*Proof.* Postponed to Example 3.14. □

Fact 2.12 implies that LFD cannot define (or even enforce) explicit equality. Indeed, any dependence model $\mathbf{M}, s$ with $s(x) = s(y)$ has an LFD-equivalent dependence model $\mathbf{N}, t$ where $t(x) \neq t(y)$. It is therefore natural to consider an extension of LFD which allows to express this.

**LFD with explicit equality.** The extension $\mathsf{LFD}^=$, introduced in [6, Section 7.2], allows on top functions and constants also the use of equality atoms $h = h'$ between terms $h$ and $h'$. It is also remarked that $\mathsf{LFD}^=$ is still a fragment of FO, since the standard translation embedding LFD into FO (surveyed in [6, Section 3.2] and Section 4.1.1) can easily be extended to $\mathsf{LFD}^=$ (cf. Remark 4.7).

Baltag and van Benthem gave a complete Hilbert-style proof system for $\mathsf{LFD}^=$ in [6, Section 7.2] . Since the extension of LFD with functions retained the decidability of plain LFD, it seemed plausible to assume that the same holds for $\mathsf{LFD}^=$. It was however left as an open question, as the previous proof techniques did not seem to work. In the next section we give a negative answer to this by proving that $\mathsf{LFD}^=$ is undecidable. In particular, this already holds for the relational fragment of $\mathsf{LFD}^=$, which only extends LFD by equality atoms $x = y$ for variables $x, y$. Since we can furthermore simulate function symbols by fresh variables as described in the previous section (originally [6, Fact 7.3]), we will only ever consider the relational fragment of $\mathsf{LFD}^=$ in this thesis, in accordance with the fact that our types $(\tau, V)$ only contain relational vocabularies.

**Definition 2.13** ($\mathsf{LFD}^=$)**.** In this thesis, $\mathsf{LFD}^=$ will denote the extension of $\mathsf{LFD}$ by equality atoms $x = y$ between variables $x, y$ with the semantics:

$$\mathbf{M}, s \models x = y \qquad \text{iff} \qquad s(x) = s(y).$$

This is the relational variant of the extension described in [6, Section 7.2].

## 2.3 Undecidability of LFD with Explicit Equality

We claim that Fact 2.12 can be seen as one of the main reasons that $\mathsf{LFD}$ or $\mathsf{LFD}$ with functions is decidable, while $\mathsf{LFD}^=$ is not. The fact makes it considerably more difficult to express any sort of paths of relations (think orders) or confluence within $\mathsf{LFD}$. To understand what we mean by this, consider $\mathsf{FO}$, where a simple sentence such as $\forall x \exists y Rxy$ yields an infinite $R$-path (or loop) in the universe of its models. The crucial point is that $x$ can take *any* value, in particular the values taken by $y$. On the other hand, if we consider the very similar $\mathsf{LFD}$-sentence $\mathsf{A}\,\mathsf{E}_x\, Rxy$, then this implies only that every value of $x$ has an $R$-partner in the values of $y$. We can assume without loss of generality that the values of $x$ are disjoint from the values of $y$. Hence, we will find an $R$-partner in the values of $y$ for every value of $x$, but nothing more. This existence of $R$-partners for the values of $y$ is not implied by our sentence. Overall, instead of obtaining a Skolem function which we can iterate arbitrarily often to obtain an infinite path of $R$-successors as in $\mathsf{FO}$, we just get some collection of $x$- and $y$-values that are $R$-pairs, i.e. $R$-paths that stop after a single step.

We can see how this limits $\mathsf{LFD}$'s ability to enforce confluence, grids, or cartesian products within models, which often lead to undecidability. But this does not apply to $\mathsf{LFD}^=$. Namely, the crucial part for the undecidability of $\mathsf{LFD}^=$ is that Fact 2.12 fails to hold, and we can use equality to "copy values between variables". For example, we can enforce $T(x) \subseteq T(y)$, as described in the following lemma.

**Lemma 2.14.** For variables $x, y$ we define the $\mathsf{LFD}^=(\varnothing, \{x, y\})$-sentence

$$\psi_{x \subseteq y} \;:=\; \mathsf{A}\,\mathsf{E}_x (y = x).$$

For every dependence model $\mathbf{M}$ on which we can evaluate $\psi_{x \subseteq y}$ we obtain

$$\mathbf{M} \models \psi_{x \subseteq y} \qquad \Longrightarrow \qquad T_{\mathbf{M}}(x) \subseteq T_{\mathbf{M}}(y).$$

*Proof.* Let $\mathbf{M} \models \psi_{x \subseteq y}$ and $a \in T_{\mathbf{M}}(x)$, so $s(x) = a$ for some $s \in T_{\mathbf{M}}$. Per semantics of $\mathsf{A}$ we have $\mathbf{M}, s \models \mathsf{E}_x(y = x)$, so there exists $t \in T_{\mathbf{M}}$ with $t =_x s$ and $\mathbf{M}, t \models x = y$. Hence

$$a = s(x) = t(x) = t(y) \in T_{\mathbf{M}}(y).$$

□

We show in Example 3.15 that the reverse implication "$T(x) \subseteq T(y)$ implies $\mathbf{M} \models \psi_{x \subseteq y}$" fails to hold. Moreover, we prove that this is not because we have the wrong definition of $\psi_{x \subseteq y}$, but rather because the inclusion $T(x) \subseteq T(y)$ cannot be defined in $\mathsf{LFD}^=$ at all.

**Conservative Reduction Classes.** We will give a brief introduction to conservative reduction classes. For a more complete background we refer the reader to [12].

Let $X, Y$ be fragments of $\mathsf{FO}$. A computable function $f \colon X \to Y$ is a *conservative reduction* if it simultaneously translates the satisfiability and finite satisfiability problem:

1. $\psi \in \mathrm{Sat}(X)$ iff $f(\psi) \in \mathrm{Sat}(Y)$,
2. $\psi \in \text{Fin-Sat}(X)$ iff $f(\psi) \in \text{Fin-Sat}(Y)$.

Here $\mathrm{Sat}(X)$ are the satisfiable formulae of $X$, and $\text{Fin-Sat}(X)$ are the finitely satisfiable formulae of $X$, i.e. those that have a finite model. Similarly, let $\text{Non-Sat}(X)$ denote the unsatisfiable formulae, $\mathrm{Val}(X)$ the valid formulae, and $\text{Inf-Axioms}(X)$ the infinity axioms of $X$, i.e. those formulae that are satisfiable but have no finite models.

Now we say that $Y$ is a *conservative reduction class* if there exists a conservative reduction $f \colon \mathsf{FO} \to Y$. Essentially, these are fragments of $\mathsf{FO}$ for which the satisfiability and finite satisfiability problem are exactly as hard as for $\mathsf{FO}$. A well-known theorem of Trakhtenbrot states that $\text{Fin-Sat}(\mathsf{FO}), \text{Non-Sat}(\mathsf{FO})$ and $\text{Inf-Axioms}(\mathsf{FO})$ are pairwise recursively inseparable[1] and therefore undecidable, see [12, Theorem 2.1.30]. It is easy to see that this carries over to conservative reduction classes. In particular, if a fragment $X$ of $\mathsf{FO}$ is a conservative reduction class, then $\mathrm{Sat}(X)$, $\text{Non-Sat}(X)$, $\mathrm{Val}(X)$, $\text{Fin-Sat}(X)$, and $\text{Inf-Axioms}(X)$ are all undecidable.

One of the classical conservative reduction classes is the Kahr-Class, usually denoted as $[\forall\exists\forall, (\omega, 1)]$. It consists of those $\mathsf{FO}$-sentences $\forall x \exists y \forall z \varphi(x, y, z)$ where $\varphi$ is a quantifier-free $\mathsf{FO}$-formula which may only use an unbounded number of monadic

[1] Two disjoint sets $X, Y$ are called recursively inseparable if there is no recursive (decidable) set $R$ such that $X \subseteq R$ and $R \cap Y = \varnothing$.

predicates and a single binary relation. For the proof that this is indeed a conservative reduction class, we refer the reader to [12, Chapter 3.1].

Below, we construct a conservative reduction from the Kahr-Class into an analogous fragment of $\mathsf{LFD}^=$, namely $\mathsf{LFD}^=$-sentences with an unbounded number of monadic predicates and a single binary relation, which we denote by $\mathsf{LFD}^=[(\omega, 1)]$.

**Theorem 2.15.** $\mathsf{LFD}^=[(\omega, 1)]$ is a conservative reduction class. In particular, $\mathsf{LFD}^=$ is undecidable.

*Proof.* As we mentioned above, $\mathsf{LFD}^=$ is a fragment of $\mathsf{FO}$, which we discuss later in Remark 4.7. Since the Kahr-Class is already a conservative reduction class, by transitivity of reductions it suffices to show that there exists a conservative reduction

$$[\forall\exists\forall, (\omega, 1)] \to \mathsf{LFD}^=[(\omega, 1)], \quad \psi \mapsto \psi^*. \tag{2.1}$$

To this end we prove the following two claims:

1. A (finite) classical model of $\psi$ induces a (finite) dependence model of $\psi^*$.
2. A (finite) dependence model of $\psi^*$ induces a (finite) classical model of $\psi$.

Given $\psi = \forall x \exists y \forall z \varphi(x, y, z)$ in the Kahr-Class, note that $\varphi \in \mathsf{LFD}^=[(\omega, 1)]$. Therefore we can define the translation of $\psi$ as

$$\psi^* := \mathsf{A}\left(\varphi(x, y, z) \wedge D_x y \wedge \bigwedge_{i=0}^{5} \vartheta_i\right),$$

where the $\vartheta_i$ are formulae inspired by the $\psi_{x \subseteq y}$ from Lemma 2.14, which allow us to copy values between variables. They are necessary to construct a cartesian product included in $T_{\mathbf{M}}(x, z)$ for the regarded dependence models $\mathbf{M}$. We require an extra variable $v$ as an additional "temporary storage" to copy values between variables. So while we keep $\varphi$ to contain only the variables $x, y, z$, overall we will need four variables, whose order we fix to $x, y, z, v$. The $\vartheta_i$ are defined as:

$$\begin{aligned}
\vartheta_0 &= \mathsf{E}_{xyv}(z = x) && \text{(copy } x \text{ to } z\text{)},\\
\vartheta_1 &= \mathsf{E}_{xyv}(z = y) && \text{(copy } y \text{ to } z\text{)},\\
\vartheta_2 &= \mathsf{E}_{xyz}(v = x) && \text{(copy } x \text{ to } v\text{)},\\
\vartheta_3 &= \mathsf{E}_{xyz}(v = y) && \text{(copy } y \text{ to } v\text{)},\\
\vartheta_4 &= \mathsf{E}_{zv}(x = z) && \text{(copy } z \text{ to } x\text{)},\\
\vartheta_5 &= \mathsf{E}_{zv}(x = v) && \text{(copy } v \text{ to } x\text{)}.
\end{aligned}$$

To prove Claim 1, assume that we have a model $\mathfrak{A}$ of $\psi$ with universe $A$. Thus the Skolem normal form of $\psi$ is satisfied in some expansion of $\mathfrak{A}$. More specifically, there exists a function $f\colon A \to A$ such that

$$\mathfrak{A} \models \varphi(a, fa, b), \qquad a, b \in A.$$

We construct the dependence model $\mathbf{M} = (\mathfrak{A}, T)$ with team $T$ defined by

$$T \coloneqq \{(a, fa, b, c) \mid a, b, c \in A\}.$$

Remember that we denote assignments by their tuple of values. So here $(a, fa, b, c)$ represents the assignment $(x, y, z, v) \mapsto (a, fa, b, c)$. It is clear that $y$ globally depends on $x$, i.e. $\mathbf{M} \models \mathsf{A}\, D_x y$, and that by choice of $T$ we also have $\mathbf{M} \models \mathsf{A}\,\varphi(x, y, z)$. The $\vartheta_i$ are satisfied at all assignments in $T$, since $T(x, z, v) = A^3$ is a cartesian product of the whole universe. Because $\mathsf{A}$ distributes over conjunction, i.e. $\mathsf{A}\,\vartheta_1 \wedge \mathsf{A}\,\vartheta_2 \equiv \mathsf{A}(\vartheta_1 \wedge \vartheta_2)$ for all $\vartheta_i \in \mathsf{LFD}$, we obtain $\mathbf{M} \models \psi^*$. Notice that if $\mathfrak{A}$ is a finite model, then $\mathbf{M}$ is finite as well, since they share the same universe $A$. This concludes the proof of Claim 1.

For the converse, Claim 2, suppose that we have a dependence model $\mathbf{M} = (\mathfrak{M}, T)$ such that $\mathbf{M} \models \psi^*$. Because of the global dependence $\mathbf{M} \models \mathsf{A}\, D_x y$ there exists a function $f\colon T(x) \to T(y)$ such that

$$t(y) = f(t(x)), \qquad t \in T. \tag{2.2}$$

Note that we have $T(y) \subseteq T(x)$, since $\vartheta_1$ and $\vartheta_4$ allow us to copy values from $y$ to $z$ and from there to $x$, just as in Lemma 2.14. Hence we have $f\colon T(x) \to T(x)$, i.e. we can iterate $f$ on values of $x$. Fix some arbitrary $s \in T$ and set $\underline{0} \coloneqq s(x)$, as well as $\underline{i} \coloneqq f^i \underline{0}$ for $i \in \mathbb{N}$.[2] Since $\mathfrak{M}$ is relational, we know that all subsets of the universe of $\mathfrak{M}$ induce a substructure of $\mathfrak{M}$. We construct our classical model for $\psi$ as

$$\mathfrak{A} \coloneqq \mathfrak{M} \restriction A \quad \text{where} \quad A \coloneqq \{\underline{i} \mid i \in \mathbb{N}\}.$$

The function $f \restriction A\colon A \to A$ plays the role of the Skolem function for $y$ in the quantification $\forall x \exists y \forall z$ of $\psi$. Now we need to make sure that $\varphi(a, fa, b)$ actually holds in $\mathfrak{M}$ (and thus in $\mathfrak{A}$) for all $a, b \in A$. Per assumption we know

$$\mathbf{M} \models \mathsf{A}\,\varphi(x, y, z) \qquad \text{and thus} \qquad \mathfrak{M} \models \varphi(t(x), f(t(x)), t(z)), \qquad t \in T.$$

Hence it suffices to show that

$$A \times A \subseteq T(x, z). \tag{2.3}$$

In the following we write $*$ as placeholder for not further specified elements of $\mathfrak{M}$.

[2] Here $f^i$ represents the $i$-fold composition of $f$.

By the expression $t \xrightarrow{\vartheta_i} t'$ for some $t \in T$ we mean to express that the existence of $t' \in T$ follows from $\mathbf{M}, t \models \vartheta_i$ (by applying the "copy-rule" which $\vartheta_i$ represents). Notice that if $t \in T$ with $t(x) = \underline{i}$, then $t(y) = f(\underline{i}) = \underline{i+1}$ by Eq. (2.2).

1. $(\underline{0}, \underline{0}) \in T(x, z)$ :
   We know $s$ looks like $(\underline{0}, \underline{1}, *, *)$. Since $s \in T$ and

   $$s = (\underline{0}, \underline{1}, *, *) \xrightarrow{\vartheta_0} (\underline{0}, \underline{1}, \underline{0}, *) =: t \qquad \text{(copy } x \text{ to } z\text{)}$$

   we see that $t \in T$ with $t(x, z) = (\underline{0}, \underline{0})$.

2. If $(\underline{0}, \underline{j}) \in T(x, z)$, then also $(\underline{0}, \underline{j+1}) \in T(x, z)$ :
   Per assumption we have $t = (\underline{0}, \underline{1}, \underline{j}, *) \in T$. Together with the derivation

   $$\begin{aligned} t &= (\underline{0}, \underline{1}, \underline{j}, *) \\ &\xrightarrow{\vartheta_2} (\underline{0}, \underline{1}, \underline{j}, \underline{0}) && \text{(copy } x \text{ to } v\text{)} \\ &\xrightarrow{\vartheta_4} (\underline{j}, \underline{j+1}, \underline{j}, \underline{0}) && \text{(copy } z \text{ to } x\text{)} \\ &\xrightarrow{\vartheta_1} (\underline{j}, \underline{j+1}, \underline{j+1}, \underline{0}) && \text{(copy } y \text{ to } z\text{)} \\ &\xrightarrow{\vartheta_5} (\underline{0}, \underline{1}, \underline{j+1}, \underline{0}) =: t' && \text{(copy } v \text{ to } x\text{)} \end{aligned}$$

   we obtain $t' \in T$ with $t'(x, z) = (\underline{0}, \underline{j+1})$.

3. If $(\underline{i}, \underline{j}) \in T(x, z)$ then also $(\underline{i+1}, \underline{j}) \in T(x, z)$ :
   Per assumption we have $t = (\underline{i}, \underline{i+1}, \underline{j}, *) \in T$. Together with the derivation

   $$\begin{aligned} t &= (\underline{i}, \underline{i+1}, \underline{j}, *) \\ &\xrightarrow{\vartheta_3} (\underline{i}, \underline{i+1}, \underline{j}, \underline{i+1}) && \text{(copy } y \text{ to } v\text{)} \\ &\xrightarrow{\vartheta_5} (\underline{i+1}, \underline{i+2}, \underline{j}, \underline{i+1}) =: t' && \text{(copy } v \text{ to } x\text{)} \end{aligned}$$

   we obtain $t' \in T$ with $t'(x, z) = (\underline{i+1}, \underline{j})$.

Now the inclusion (2.3) follows from the definition of $A$ and a trivial induction. By the above argument this proves $\mathfrak{A} \models \varphi(a, fa, b)$ for all $a, b \in A$ and hence

$$\mathfrak{A} \models \forall x \exists y \forall z \varphi(x, y, z).$$

Thus $\mathfrak{A}$ is a classical model for $\psi$. Again it is clear that if $\mathbf{M}$ is finite, then $\mathfrak{A}$ is finite as well, since its universe $A$ is a subset of the universe of $\mathbf{M}$. This concludes the proof of Claim 2. □

# Chapter 3

# Bisimulation

We define bisimulation and standard logical tools for $\mathsf{LFD}$. An analogue of the classical Ehrenfeucht-Fraïssé Theorem relating bisimilarity with logical equivalence is shown in Section 3.1. We use this theorem in Section 3.2 to prove the undefinability of various natural statements about the team of a dependence model. Due to it being only a small overhead, we will treat $\mathsf{LFD}$ and $\mathsf{LFD}^=$ simultaneously and also consider infinitary variants of these logics. Henceforth let $\mathcal{L}$ denote $\mathsf{LFD}$ or $\mathsf{LFD}^=$.

**Definition 3.1** (Bisimulation)**.** Let $\mathbf{M}$ and $\mathbf{N}$ be two dependence models of the same type $(\tau, V)$. A binary relation $Z \subseteq T_{\mathbf{M}} \times T_{\mathbf{N}}$ is an $\mathcal{L}$-bisimulation between $\mathbf{M}$ and $\mathbf{N}$ if for all $(s,t) \in Z$:

1. $\mathbf{M}, s$ and $\mathbf{N}, t$ agree on the atoms of $\mathcal{L}$:
   (a) For all $R \in \tau$ and $\mathbf{x} \in V^{\mathrm{ar}(R)}$ we have $\mathbf{M}, s \models R\mathbf{x}$ iff $\mathbf{N}, t \models R\mathbf{x}$.
   (b) For all (finite) $X \subseteq V$ and $y \in V$ we have $\mathbf{M}, s \models D_X y$ iff $\mathbf{N}, t \models D_X y$.
   (c) (Only if $\mathcal{L} = \mathsf{LFD}^=$) For all $x, y \in V$ we have $s(x) = s(y)$ iff $t(x) = t(y)$.
2. (back) For all $t' \in T_{\mathbf{N}}$ and all *finite* $X \subseteq t' \overline{\sqcap}\, t = \{x \in V \mid t' =_x t\}$ there is some $s' \in T_{\mathbf{M}}$ with $(s', t') \in Z$ and $s' =_X s$.
3. (forth) For all $s' \in T_{\mathbf{M}}$ and all *finite* $X \subseteq s' \overline{\sqcap}\, s = \{x \in V \mid s' =_x s\}$ there is some $t' \in T_{\mathbf{N}}$ with $(s', t') \in Z$ and $t' =_X t$.

The reason we need to restrict ourselves to finite sets $X \subseteq t' \overline{\sqcap}\, t$ is that we want bisimilarity to correspond to logical equivalence (as we show later in Theorem 3.12) and that $\mathsf{LFD}$ only allows finite sets within our modalities $\mathsf{D}_X$ and $\mathsf{E}_X$. For finite types, we can give the following equivalent definition.

**Fact 3.2** (Bisimulation for finite types)**.** Let $\mathbf{M}$ and $\mathbf{N}$ be two dependence models of the same *finite* type $(\tau, V)$. Then $Z \subseteq T_{\mathbf{M}} \times T_{\mathbf{N}}$ is an $\mathcal{L}$-bisimulation between $\mathbf{M}$ and $\mathbf{N}$ if and only if

1. $\mathbf{M}, s$ and $\mathbf{N}, t$ agree on the atoms of $\mathcal{L}$.
2. (back) For all $t' \in T_{\mathbf{N}}$ there exists $s' \in T_{\mathbf{M}}$ with $(s', t') \in Z$ and $s' =_X s$, where $X = t' \overline{\cap}\, t = \{x \in V \mid t' =_x t\}$.
3. (forth) For all $s' \in T_{\mathbf{M}}$ there exists $t' \in T_{\mathbf{N}}$ with $(s', t') \in Z$ and $t' =_X t$, where $X = s' \overline{\cap}\, s = \{x \in V \mid s' =_x s\}$.

The union of $\mathcal{L}$-bisimulations is again an $\mathcal{L}$-bisimulation, hence there is always some inclusion-maximal $\mathcal{L}$-bisimulation $\sim_{\mathcal{L}}$ between two $(\tau, V)$ dependence models $\mathbf{M}$ and $\mathbf{N}$ (although it may be empty). In this sense we write $\mathbf{M}, s \sim_{\mathcal{L}} \mathbf{N}, t$ and say $\mathbf{M}, s$ is $\mathcal{L}$-bisimilar to $\mathbf{N}, t$ whenever there exists an $\mathcal{L}$-bisimulation $Z$ between $\mathbf{M}$ and $\mathbf{N}$ with $(s, t) \in Z$. If $\mathcal{L}$ is clear from context or irrelevant to it, we use $\sim$ instead of $\sim_{\mathcal{L}}$.

**Remark 3.3.** Notice that $\mathsf{LFD}^{=}$-bisimilar assignments induce partial isomorphisms of the underlying classical structures $\mathfrak{M}$ of $\mathbf{M}$ and $\mathfrak{N}$ of $\mathbf{N}$; because of the requirement that $s, t$ with $(s, t) \in Z$ agree on equalities, the map

$$\pi \colon M \restriction s(V) \overset{\cong}{\to} N \restriction t(V), \quad s(x) \mapsto t(x),\ x \in V$$

is a well defined bijection.[1] Since $s$ and $t$ also agree on relational facts, we get

$$\mathfrak{M} \models R\mathbf{a} \quad \text{iff} \quad \mathfrak{N} \models R\pi\mathbf{a}, \qquad R \in \tau, \quad \mathbf{a} \in s(V)^{\mathrm{ar}(R)}.$$

Hence $\pi$ is a partial isomorphism between $\mathfrak{M}$ and $\mathfrak{N}$.

**Definition 3.4** (Ordinal approximations to bisimulation)**.** Let $\mathbf{M}, s$ and $\mathbf{N}, t$ be pointed dependence models of the same type $(\tau, V)$. We write $\mathbf{M}, s \sim^0_{\mathcal{L}} \mathbf{N}, t$ if $\mathbf{M}, s$ and $\mathbf{N}, t$ agree on $\mathcal{L}$-atoms, and say that $\mathbf{M}, s$ and $\mathbf{N}, t$ are 0-$\mathcal{L}$-bisimilar. If $\mathbf{M}$ and $\mathbf{N}$ are clear from context, we shorten this to $s \sim^0_{\mathcal{L}} t$. We define $s \sim^\alpha_{\mathcal{L}} t$ for ordinals $\alpha$ by (transfinite) induction; in the case of successor ordinals, i.e. when defining $s \sim^{\alpha+1}_{\mathcal{L}} t$, we require the conditions

- $(\alpha+1)$-back: For all $t' \in T_{\mathbf{N}}$ and all *finite* $X \subseteq t' \overline{\cap}\, t$ there exists some $s' \in T_{\mathbf{M}}$ with $s' \sim^\alpha_{\mathcal{L}} t'$ and $s' =_X s$.

[1] Here $s(V)$ denotes the image $\{s(v) \mid v \in V\}$ of $s$, likewise for $t$.

- $(\alpha+1)$-forth: For all $s' \in T_{\mathbf{M}}$ and all *finite* $X \subseteq s' \sqcap s$ there exists some $t' \in T_{\mathbf{N}}$ with $s' \sim_{\mathcal{L}}^{\alpha} t'$ and $t' =_X t$.

For limit ordinals $\lambda$, we say that $s \sim_{\mathcal{L}}^{\lambda} t$ iff $s \sim_{\mathcal{L}}^{\alpha} t$ for all $\alpha < \lambda$, so essentially

$$\sim_{\mathcal{L}}^{\lambda} = \bigcap_{\alpha<\lambda} \sim_{\mathcal{L}}^{\alpha} . \tag{3.1}$$

Note that $\alpha$-$\mathcal{L}$-bisimilarity implies $\beta$-$\mathcal{L}$-bisimilarity for all $\beta < \alpha$. Furthermore, full $\mathcal{L}$-bisimilarity is now simply given by $\sim_{\mathcal{L}} = \bigcap_{\alpha\in\mathbf{Ord}} \sim_{\mathcal{L}}^{\alpha}$. As before, if $\mathcal{L}$ is clear from context or irrelevant to it, we use $\sim^{\alpha}$ instead of $\sim_{\mathcal{L}}^{\alpha}$.

**Remark 3.5.** In the case of finite types we can analogously adapt Fact 3.2 to

1. $(\alpha+1)$-back for finite types: For all $t' \in T_{\mathbf{N}}$ there exists $s' \in T_{\mathbf{M}}$ with $s' \sim_{\mathcal{L}}^{\alpha} t'$ and $s' =_X s$, where $X = t' \sqcap t$.

We want to emphasize that the back and forth conditions of our bisimulations do not require the regarded assignments $s'$ or $t'$ to actually agree on any variables with $s$ or $t$ respectively, i.e. $s' \sqcap s$ and $t' \sqcap t$ may be empty. The reason for this is that $\mathsf{LFD}$ has the global modalities $\mathsf{A} = \mathsf{D}_{\varnothing}$ and $\mathsf{E} = \mathsf{E}_{\varnothing}$[2], and that we want bisimilarity to correspond to logical equivalence. As an example, let $V$ be finite and

$$\psi := \mathsf{E}(Rx) \wedge \bigwedge_{v\in V} \neg\, \mathsf{E}_v\, Rx.$$

Then $\mathbf{M}, s \models \psi$ means that there is some $s' \in T_{\mathbf{M}}$ with $\mathbf{M}, s' \models Rx$, but that there is no $s''$ in any $=_x$-class of $s$ with this property, and thus $s' \sqcap s = \varnothing$. Since $\mathsf{LFD}$ is able to witness this $s'$, it is natural to require a bisimilar $t' \in T_{\mathbf{N}}$ for *every* $s' \in T_{\mathbf{M}}$, and not just for those $s'$ that agree with $s$ on some variable. Likewise for the back condition.

This forces every bisimulation to be *global*, meaning that every assignment in $T_{\mathbf{M}}$ is bisimilar to at least one assignment in $T_{\mathbf{N}}$, and vice versa. Likewise, it follows directly from Definition 3.4 that for ordinals $\alpha$, $\mathbf{M}, s \sim^{\alpha+1} \mathbf{N}, t$ entails a global $\alpha$-bisimulation, in the sense that each assignment is $\alpha$-bisimilar to one in the other team. For $\lambda$ a limit ordinal, $\mathbf{M}, s \sim^{\lambda} \mathbf{N}, t$ entails that for all $\alpha < \lambda$, every assignment is $\alpha$-bisimilar to one in the other team.

This is a common consequence of having global modalities; in the context of, say, $\mathsf{ML}$ with an explicitly added global modality, often denoted $\mathsf{ML}(\forall)$, the canonical bisimulation is just the global version of ordinary $\mathsf{ML}$-bisimulation.[3]

[2] We say that some modality is global if its corresponding accessibility relation is the all-relation, i.e. contains all possible pairs of objects, as is the case for $=_{\varnothing}$ on teams.

[3] A good reference for this and similar results on bisimulations for modal logic over special classes of frames is [14].

Since we already defined bisimulation for infinite ordinals, the corresponding step for our logics is to consider their infinitary variants.

**Definition 3.6** (Infinitary $\mathsf{LFD}$ / $\mathsf{LFD}^=$)**.** For an arbitrary type $(\tau, V)$, define the infinitary logic $\mathcal{L}_\infty(\tau, V)$ inductively by

1. $\mathcal{L}(\tau, V) \subseteq \mathcal{L}_\infty(\tau, V)$.
2. If $\varphi \in \mathcal{L}_\infty(\tau, V)$, then $\neg\varphi, \mathsf{D}_X\, \varphi \in \mathcal{L}_\infty(\tau, V)$ for all finite $X \subseteq V$.
3. If $\Phi \subseteq \mathcal{L}_\infty(\tau, V)$ is a *set*, then $\bigwedge \Phi \in \mathcal{L}_\infty(\tau, V)$.

The semantics of $\mathcal{L}$ extend to $\mathcal{L}_\infty$ in an obvious way.

**Definition 3.7** (Quantifier Rank)**.** The quantifier rank $\mathsf{qr}(\varphi)$ of a formula $\varphi \in \mathsf{LFD}^=_\infty$ is an ordinal defined recursively by

1. $\mathsf{qr}(\varphi) = 0$ for atoms $\varphi$.
2. $\mathsf{qr}(\neg\varphi) = \mathsf{qr}(\varphi)$.
3. $\mathsf{qr}\left(\bigwedge_{i\in I} \varphi_i\right) = \sup_{i\in I} \mathsf{qr}(\varphi_i)$.
4. $\mathsf{qr}(\mathsf{D}_X\, \varphi) = \mathsf{qr}(\varphi) + 1$.

Note that for finite conjunctions the supremum is just a maximum and hence *all non-infinitary formulae* $\varphi \in \mathsf{LFD}^=$ *have a finite quantifier rank* $\mathsf{qr}(\varphi) \in \mathbb{N}$.

The above definition of the quantifier rank does not capture the notion of “how far a formula looks into the model”, starting at some assignment $s$, as one would have for $\mathsf{ML}$ with Kripke semantics. For one part this is again because we have the global modalities $\mathsf{A}$ and $\mathsf{E}$ available. Moreover, the truth of dependence atoms is clearly influenced by other assignments than the one it is evaluated on, and hence $\mathsf{qr}(D_X y) = 0$ also does not capture the mentioned idea. The reason we defined the quantifier rank as above is that we treat all atoms equally in the definition of bisimulation, so even 0-bisimilar assignments have to agree on dependence atoms. This yields a correspondence of quantifier rank $\alpha$ and $\alpha$-bisimilarity for all $\alpha \in \mathbf{Ord}$, as shown below in our Ehrenfeucht-Fraïssé analogue Theorem 3.12.

**Definition 3.8** (Logical equivalence)**.** Consider pointed $(\tau, V)$ dependence models $\mathbf{M}, s$ and $\mathbf{N}, t$.

1. For a $k \in \mathbb{N}$ we say $\mathbf{M}, s$ and $\mathbf{N}, t$ are $\mathcal{L}$-equivalent up to (quantifier) rank $k$ if for all $\varphi \in \mathcal{L}(\tau, V)$ with $\mathsf{qr}(\varphi) \leq k$:

$$\mathbf{M}, s \models \varphi \qquad \text{iff} \qquad \mathbf{N}, t \models \varphi.$$

   In this case we write $\mathbf{M}, s \equiv_{\mathcal{L}}^{k} \mathbf{N}, t$.

2. $\mathcal{L}_\infty$-equivalence up to rank $\alpha \in \mathbf{Ord}$ is defined analogously.

3. The models are $\mathcal{L}$-equivalent, written $\mathbf{M}, s \equiv_{\mathcal{L}} \mathbf{N}, t$ if we can drop the restriction on the quantifier rank. Likewise for $\mathcal{L}_\infty$.

Since all $\varphi \in \mathcal{L}$ have a finite quantifier rank, $\mathcal{L}$-equivalence corresponds to $\mathcal{L}$-equivalence up to rank $k$ for all $k \in \mathbb{N}$. Furthermore, we usually write $\equiv_{\mathcal{L}}^{\infty}$ instead of $\equiv_{\mathcal{L}_\infty}$ in the case of unrestricted equivalence in the infinitary logic $\mathcal{L}_\infty$.

Notice that 0-$\mathcal{L}$-bisimilarity and $\mathcal{L}$-equivalence up to rank 0 coincide, so we can state the first requirement of $\mathcal{L}$-bisimulation, namely agreement on atoms, as $\mathbf{M}, s \sim_{\mathcal{L}}^{0} \mathbf{N}, t$ or equivalently $\mathbf{M}, s \equiv_{\mathcal{L}}^{0} \mathbf{N}, t$.

**Definition 3.9.** For $\varphi, \psi \in \mathcal{L}(\tau, V)$ we write $\varphi \equiv \psi$ if

$$\mathbf{M}, s \models \varphi \qquad \text{iff} \qquad \mathbf{M}, s \models \psi, \qquad\qquad \mathbf{M}, s \in \mathcal{DEP}[\tau, V].$$

Equivalence of formulae in $\mathcal{L}_\infty(\tau, V)$ is defined analogously.

## 3.1 An Ehrenfeucht-Fraïssé Analogue

Given a logic, a common goal is to find a correspondence between logical indistinguishability and behavioural equivalence in some structural form, often as a relation akin to bisimulation, a collection of partial isomorphisms, or a winning strategy of certain two-player games. For FO we have back-and-forth systems and Ehrenfeucht-Fraïssé games (cf. [25, Chapter 3.3]), whereas for ML one has ordinary bisimulation and the corresponding bisimulation games (cf. [11, Chapter 2.2]). The following results and in particular Theorem 3.12 show that the bisimulations defined above fulfill such a role for LFD and LFD$^=$. Again we let $\mathcal{L}$ denote either LFD or LFD$^=$.

**Lemma 3.10.** Let $(\tau, V)$ be a finite type. For all $\mathbf{M}, s \in \mathcal{DEP}[\tau, V]$ and $k \in \mathbb{N}$ there exists a formula $\chi_{\mathbf{M},s}^{k} \in \mathcal{L}(\tau, V)$ of quantifier rank $k$ that defines the $\sim_{\mathcal{L}}^{k}$-class of $\mathbf{M}, s$. More formally, we have for all $\mathbf{N}, t \in \mathcal{DEP}[\tau, V]$ that

$$\mathbf{N}, t \models \chi_{\mathbf{M},s}^{k} \qquad \text{iff} \qquad \mathbf{N}, t \sim_{\mathcal{L}}^{k} \mathbf{M}, s.$$

Moreover, up to $\mathcal{L}$-equivalence, the number of such $\chi^k_{\mathbf{M},s}$ depends only on $\tau, V$ and $k$, and is always finite.

*Proof.* We construct $\chi^k_{\mathbf{M},s}$ and prove the claims by induction on $k \geq 0$. We require $(\tau, V)$ to be finite so that up to $\mathcal{L}$-equivalence there are only finitely many formulae $\varphi \in \mathcal{L}(\tau, V)$ with $\mathsf{qr}(\varphi) = 0$, which allows us to define

$$\chi^0_{\mathbf{M},s} \coloneqq \bigwedge\{\varphi \in \mathcal{L}(\tau, V) \mid \mathsf{qr}(\varphi) = 0,\ \mathbf{M}, s \models \varphi\}.$$

Clearly $\chi^0_{\mathbf{M},s}$ has quantifier rank 0 and defines the $\sim^0_{\mathcal{L}}$-class of $\mathbf{M}, s$. Furthermore, the number of possible $\chi^0_{\mathbf{M},s}$ only depends on $\tau$, $V$ and $k = 0$. Now assume we proved the claim for $k \in \mathbb{N}$. So the $\chi^k_{\mathbf{M},s}$ define the $\sim^k_{\mathcal{L}}$-class of $\mathbf{M}, s$. Hence we can express the $(k+1)$-back and $(k+1)$-forth conditions from Definition 3.4 in $\mathsf{LFD}$ via

$$\varphi^{k+1}_{\text{back}} \coloneqq \bigwedge_{\substack{X \subseteq V \\ X \text{ finite}}} \mathsf{D}_X \bigvee_{\substack{s' \in T_{\mathbf{M}} \\ s' =_X s}} \chi^k_{\mathbf{M},s'}$$

$$\text{and} \quad \varphi^{k+1}_{\text{forth}} \coloneqq \bigwedge_{s' \in T_{\mathbf{M}}} \bigwedge_{\substack{X \subseteq s' \mathbin{\overline{\cap}} s \\ X \text{ finite}}} \mathsf{E}_X \, \chi^k_{\mathbf{M},s'}.$$

We set $\chi^{k+1}_{\mathbf{M},s} \coloneqq \varphi^{k+1}_{\text{forth}} \wedge \varphi^{k+1}_{\text{back}}$. By the induction hypothesis we know that $\mathsf{qr}(\chi^k_{\mathbf{M},s}) = k$ and the number of different $\chi^k_{\mathbf{M},s}$ depends only on $\tau, V$ and $k$. Thus we obtain by the above definitions that $\mathsf{qr}(\chi^{k+1}_{\mathbf{M},s}) = k + 1$ and that the number of $\chi^{k+1}_{\mathbf{M},s}$ also depends only on $\tau, V$ and $k + 1$. Since up to $\mathcal{L}$-equivalence there are only finitely many $\chi^k_{\mathbf{M},s}$, the conjunctions are essentially finite, so that $\chi^{k+1}_{\mathbf{M},s}$ is a well-defined formula in $\mathcal{L}(\tau, V)$. Therefore

$$\mathbf{N}, t \models \chi^{k+1}_{\mathbf{M},s} \qquad \text{iff} \qquad \mathbf{N}, t \sim^{k+1}_{\mathcal{L}} \mathbf{M}, s,$$

which concludes the induction step. □

**Lemma 3.11.** Let $(\tau, V)$ be an arbitrary type. For all $\mathbf{M}, s \in \mathcal{DEP}[\tau, V]$ and $\alpha \in \mathbf{Ord}$ there exists a formula $\chi^\alpha_{\mathbf{M},s} \in \mathcal{L}_\infty(\tau, V)$ of quantifier rank $\alpha$ that defines the $\sim^\alpha_{\mathcal{L}}$-class of $\mathbf{M}, s$. More formally, we have for all $\mathbf{N}, t \in \mathcal{DEP}[\tau, V]$ that

$$\mathbf{N}, t \models \chi^\alpha_{\mathbf{M},s} \qquad \text{iff} \qquad \mathbf{N}, t \sim^\alpha_{\mathcal{L}} \mathbf{M}, s.$$

*Proof.* The proof is analogous to the one for the finite case. One proceeds by transfinite induction, first defining $\chi^0_{\mathbf{M},s}$ as before and then $\chi^{\alpha+1}_{\mathbf{M},s}$ exactly as before: we simply replace $k$ with $\alpha$ in the definition of $\chi^{k+1}_{\mathbf{M},s}$ above. For limit ordinals $\lambda$, set

$$\chi^\lambda_{\mathbf{M},s} \coloneqq \bigwedge_{\alpha < \lambda} \chi^\alpha_{\mathbf{M},s},$$

which corresponds to definition of $\sim_{\mathcal{L}}^{\lambda}$, see Eq. (3.1) in Definition 3.4. □

**Theorem 3.12** (Ehrenfeucht-Fraïssé and Karp theorems for $\mathsf{LFD}$ and $\mathsf{LFD}^{=}$**).**
Let $\mathcal{L}$ denote $\mathsf{LFD}$ or $\mathsf{LFD}^{=}$. For $\mathbf{M}, s$ and $\mathbf{N}, t$ of the same *finite* type it holds that

$$\mathbf{M}, s \sim_{\mathcal{L}}^{k} \mathbf{N}, t \qquad \text{iff} \qquad \mathbf{M}, s \equiv_{\mathcal{L}}^{k} \mathbf{N}, t, \qquad\qquad k \in \mathbb{N}.$$

As a consequence we obtain that under those same conditions

$$\mathbf{M}, s \sim_{\mathcal{L}}^{\omega} \mathbf{N}, t \qquad \text{iff} \qquad \mathbf{M}, s \equiv_{\mathcal{L}} \mathbf{N}, t.$$

For arbitrary (i.e. not necessarily finite) types it holds that

$$\mathbf{M}, s \sim_{\mathcal{L}}^{\alpha} \mathbf{N}, t \qquad \text{iff} \qquad \mathbf{M}, s \equiv_{\mathcal{L}_\infty}^{\alpha} \mathbf{N}, t, \qquad\qquad \alpha \in \mathbf{Ord}$$

and therefore

$$\mathbf{M}, s \sim_{\mathcal{L}} \mathbf{N}, t \qquad \text{iff} \qquad \mathbf{M}, s \equiv_{\mathcal{L}}^{\infty} \mathbf{N}, t.$$

*Proof.* Consider some finite type $(\tau, V)$ and let $\mathbf{M}, s$ and $\mathbf{N}, t$ be $(\tau, V)$ dependence models. It was already mentioned that $\sim_{\mathcal{L}}^{0}$ corresponds to $\equiv_{\mathcal{L}}^{0}$. We proceed by induction, and assume that the claimed correspondence between $\sim_{\mathcal{L}}^{k}$ and $\equiv_{\mathcal{L}}^{k}$ has already been shown for some $k \in \mathbb{N}$.

1. "$\Longrightarrow$": We assume $\mathbf{M}, s \sim_{\mathcal{L}}^{k+1} \mathbf{N}, t$ and show $\mathbf{M}, s \equiv_{\mathcal{L}}^{k+1} \mathbf{N}, t$.

   A formula of quantifier rank $k+1$ is just a boolean combination of at least one formula of the form $\mathsf{D}_X\, \varphi$ with $\mathsf{qr}(\varphi) = k$, and other formulae with quantifier rank at most $k$. Since $(k+1)$-$\mathcal{L}$-bisimilarity entails $k$-$\mathcal{L}$-bisimilarity, the induction hypothesis yields $\mathbf{M}, s \equiv_{\mathcal{L}}^{k} \mathbf{N}, t$. So it suffices to show that $\mathbf{M}, s \models \mathsf{D}_X\, \varphi$ iff $\mathbf{N}, t \models \mathsf{D}_X\, \varphi$, for all (finite) $X \subseteq V$ and $\varphi \in \mathcal{L}(\tau, V)$ of quantifier rank $k$.

   To this end, suppose $\mathbf{M}, s \models \mathsf{D}_X\, \varphi$ for such an $X$ and $\varphi$. We want to show $\mathbf{N}, t \models \mathsf{D}_X\, \varphi$. If $t' \in T_{\mathbf{N}}$ is an arbitrary assignment in the $=_X$-class of $t$, then by the $(k+1)$-back condition there exists some $s' \in T_{\mathbf{M}}$ with

   $$\mathbf{M}, s' \sim_{\mathcal{L}}^{k} \mathbf{N}, t' \qquad \text{and} \qquad s' =_X s.$$

   Since we assumed $\mathbf{M}, s \models \mathsf{D}_X\, \varphi$, this implies $\mathbf{M}, s' \models \varphi$. By the induction hypothesis we infer from $\mathbf{M}, s' \sim_{\mathcal{L}}^{k} \mathbf{N}, t'$ that

   $$\mathbf{M}, s' \equiv_{\mathcal{L}}^{k} \mathbf{N}, t' \qquad \text{and therefore} \qquad \mathbf{N}, t' \models \varphi.$$

   As $t'$ was an arbitrary assignment in the $=_X$-class of $t$, we conclude that $\mathbf{N}, t \models \mathsf{D}_X\, \varphi$. The converse implication "$\mathbf{N}, t \models \mathsf{D}_X\, \varphi$ implies $\mathbf{M}, s \models \mathsf{D}_X\, \varphi$" follows analogously, using the $(k+1)$-forth condition instead. By our above argument we obtain that $\mathbf{M}, s \equiv_{\mathcal{L}}^{k+1} \mathbf{N}, t$, and conclude that $(k+1)$-$\mathcal{L}$-bisimilarity implies $\mathcal{L}$-equivalence up to rank $k+1$.

2. "$\Longleftarrow$": We assume that $\mathbf{M}, s \equiv_{\mathcal{L}}^{k+1} \mathbf{N}, t$ and show $\mathbf{M}, s \sim_{\mathcal{L}}^{k+1} \mathbf{N}, t$.

   Since $\mathbf{M}, s$ satisfies its own characteristic formula of rank $k+1$, we obtain

   $$\mathbf{N}, t \models \chi_{\mathbf{M},s}^{k+1} \qquad \text{which is equivalent to} \qquad \mathbf{M}, s \sim_{\mathcal{L}}^{k+1} \mathbf{N}, t$$

   by Lemma 3.10. Hence $\mathcal{L}$-equivalence up to rank $k+1$ implies $(k+1)$-$\mathcal{L}$-bisimilarity.

This concludes the induction step and with that the proof of the first part of the theorem; that in restriction to finite types $(\tau, V)$, the concepts of $k$-$\mathcal{L}$-bisimilarity and $\mathcal{L}$-equivalence up to rank $k$ coincide, for every $k \in \mathbb{N}$.

The second claim is an immediate consequence. We already mentioned in Definition 3.8 that in the case of finite types

$$\mathbf{M}, s \equiv_{\mathcal{L}} \mathbf{N}, t \qquad \text{iff} \qquad \mathbf{M}, s \equiv_{\mathcal{L}}^{k} \mathbf{N}, t, \quad k \in \mathbb{N}.$$

We also know that $\mathbf{M}, s$ and $\mathbf{N}, t$ are $\omega$-$\mathcal{L}$-bisimilar iff they are $k$-$\mathcal{L}$-bisimilar for all $k < \omega$, i.e. all $k \in \mathbb{N}$. Thus it follows from the first claim that

$$\mathbf{M}, s \sim_{\mathcal{L}}^{\omega} \mathbf{N}, t \qquad \text{iff} \qquad \mathbf{M}, s \equiv_{\mathcal{L}} \mathbf{N}, t.$$

The proof for the case of arbitrary types and $\mathcal{L}_\infty$ is an easy adaption of the above proof. For the sake of completeness, we give the details in Theorem C.3 in the appendix. □

**Corollary 3.13.**

1. For finite types $(\tau, V)$, if $\varphi \in \mathcal{L}(\tau, V)$ and $k = \mathsf{qr}(\varphi)$, then

   $$\varphi \;\equiv \bigvee_{\mathbf{M},s \models \varphi} \chi_{\mathbf{M},s}^{k},$$

   where the disjunction ranges over all $\mathbf{M}, s \in \mathcal{DEP}[\tau, V]$. We know this is a well-defined $\mathcal{L}$-formula since Lemma 3.10 tells us that up to $\mathcal{L}$-equivalence there are only finitely many such $\chi_{\mathbf{M},s}^{k}$.

2. For arbitrary types $(\tau, V)$, if $\varphi \in \mathcal{L}_\infty(\tau, V)$ and $\alpha = \mathsf{qr}(\varphi)$, then

   $$\varphi \;\equiv \bigvee_{\mathbf{M},s \models \varphi} \chi_{\mathbf{M},s}^{\alpha}.$$

*Proof.* We only prove the first claim, as the infinitary variant is analogous. Assume $\mathbf{M}, s \in \mathcal{DEP}[\tau, V]$ with $\mathbf{M}, s \models \varphi$. Then $\chi^k_{\mathbf{M},s}$ appears (up to $\mathcal{L}$-equivalence) in the disjunction on the right hand side, and hence $\mathbf{M}, s$ also models the disjunction.

Conversely, if $\mathbf{M}, s$ models the disjunction, then we must have $\mathbf{M}, s \models \chi^k_{\mathbf{N},t}$ for some $(\tau, V)$ dependence model $\mathbf{N}, t$ with $\mathbf{N}, t \models \varphi$. Per Lemma 3.10 we obtain that $\mathbf{M}, s \sim^k_{\mathcal{L}} \mathbf{N}, t$, which by Theorem 3.12 corresponds to $\mathbf{M}, s \equiv^k_{\mathcal{L}} \mathbf{N}, t$. As $\mathsf{qr}(\varphi) = k$, it follows that $\mathbf{M}, s \models \varphi$. □

## 3.2 Proving Undefinability via Bisimulation

The conditions for bisimulation are usually easy to check for models of small size. Given a reasonable representation of the considered models, we can also let a computer verify whether some relation is a bisimulation, or let it search for a bisimulation between given models.[4] Bisimulation can be used to show undefinability of some property of (pointed) dependence models, by finding two such models that are bisimilar but differ on said property. By Theorem 3.12, we then know that these models are logically indistinguishable in LFD / $\mathsf{LFD}^=$, so the considered property cannot be defined in the respective logic. In the following we give a few examples.

**Example 3.14.** Fact 2.12 (originally [6, Fact 7.2]) states that every dependence model has an LFD-equivalent distinguished model, in which distinct variables take disjoint sets of values. Although easy to see, we can now give a short proof. In [6, Fact 7.2], given a $(\tau, V)$ dependence model $\mathbf{M}$ with universe $M$, the authors define a $(\tau, V)$ dependence model $\mathbf{M}^d$ with universe $M^d := V \times M$, relations

$$R^{\mathbf{M}^d} := \{((x_1, m_1), \ldots, (x_{\mathrm{ar}(R)}, m_{\mathrm{ar}(R)})) \mid (m_1, \ldots, m_{\mathrm{ar}(R)}) \in R^{\mathbf{M}}\}$$

and $T_{\mathbf{M}^d} := \{s^d \mid s \in T_{\mathbf{M}}\}$ where $s^d(x) := (x, s(x))$. It is easy to see that

$$Z := \{(s, s^d) \mid s \in T_{\mathbf{M}}\}$$

is an LFD-bisimulation between $\mathbf{M}$ and $\mathbf{M}^d$ as defined in Definition 3.1. Indeed, per definition of the $R^{\mathbf{M}^d}$ we know that $s$ and $s^d$ always agree on relational atoms. Moreover, for all $s', s \in T$ and $X \subseteq V$ it holds that

$$s' =_X s \qquad \text{iff} \qquad s'^d =_X s^d.$$

Thus $s$ and $s^d$ agree on dependence atoms as well, and the only choice one has

[4]See e.g. the python library written by the author, which was used to find and minimize most of the examples in this thesis: https://git.rwth-aachen.de/philpuetzstueck/lfd-sat.

during the back and forth clauses (each assignment is bisimilar to exactly one other) always works out. Therefore we know by Theorem 3.12 that $\mathbf{M}, s \equiv^{\infty}_{\mathsf{LFD}} \mathbf{M}^d, s^d$ for all $s \in T_{\mathbf{M}}$, which proves the claim.

**Example 3.15.** Consider the $\mathsf{LFD}^=$-sentence $\psi_{x\subseteq y}$ defined in Lemma 2.14. If $\mathbf{M} \models \psi_{x\subseteq y}$, then for each $s \in T := T_{\mathbf{M}}$ there is some assignment $t \in T$ where $t(y) = t(x) = s(x)$. We proved that this represents a sufficient condition for $T(x) \subseteq T(y)$. But it is not necessary: consider the team

$$T = \{(0,1), (1,0)\}.$$

Here it is clear that $T(x) = \{0,1\} = T(y)$, but our formula $\psi_{x\subseteq y}$ is not satisfied in a model with this set of assignments. To prove that we cannot define an $\mathsf{LFD}^=$-formula that precisely defines the inclusion $T(x) \subseteq T(y)$, we can use Theorem 3.12. It suffices to show this for $\tau = \varnothing$ and $V = \{x, y\}$, because the example below can be adapted accordingly, e.g. by letting all new relation symbols $R \in \tau$ be interpreted as $R^{\mathbf{M}} = R^{\mathbf{N}} = \varnothing$, which guarantees that the bisimilar assignments below still agree on all relational atoms. Moreover, letting variables apart from $x, y$ be constant within both teams ensures agreement on dependence atoms, and then it is also easy to guarantee agreement on equality atoms.

Consider dependence models $\mathbf{M}, \mathbf{N}$ of type $(\varnothing, \{x, y\})$ with teams given by

$$T_{\mathbf{M}} := \{(a,b), (b,a)\} \qquad \text{and} \qquad T_{\mathbf{N}} := \{(1,2), (2,0)\}.$$

Note that $T_{\mathbf{M}}(x) \subseteq T_{\mathbf{M}}(y)$, but $T_{\mathbf{N}}(x) \not\subseteq T_{\mathbf{N}}(y)$. Now let $Z$ be the binary relation on $T_{\mathbf{M}} \times T_{\mathbf{N}}$ defined by

$$(a,b)\; Z\; (1,2) \qquad \text{and} \qquad (b,a)\; Z\; (2,0).$$

It is easy to verify that $Z$ is an $\mathsf{LFD}^=$-bisimulation and hence $\mathbf{M}, s \equiv^{\infty}_{\mathsf{LFD}^=} \mathbf{N}, t$ by Theorem 3.12, for all $(s,t) \in Z$. Indeed, note that $\neg D_{\varnothing} x, \neg D_{\varnothing} y, D_x y, D_y x$ and $x \neq y$ hold at all assignments in both teams, so the pairs of assignments are already 0-bisimilar. Furthermore, in both teams we see that the present two assignments do not agree on any variables, so evidently the only choice we have at the back and forth clauses always works out. We conclude that the inclusion $T(x) \subseteq T(y)$ is not $\mathsf{LFD}^=_{\infty}$-definable.

This begs the question of what happens when we extend $\mathsf{LFD}$ with the ability to define inclusion, e.g. with atoms $x \subseteq y$ that have the semantics

$$\mathbf{M}, s \models x \subseteq y \qquad \text{iff} \qquad T_{\mathbf{M}}(x) \subseteq T_{\mathbf{M}}(y).$$

What can be said about this extension? Can the proof of undecidability for $\mathsf{LFD}^=$ be adapted?

**Example 3.16** ($\mathsf{LFD}^=$-Undefinability of cartesian products)**.** In the proof of undecidability of $\mathsf{LFD}^=$ we had to *embedd* a cartesian product into $T(x,z)$ which included all pairs that appear in the universal $\mathsf{FO}$-quantification $\forall x \forall z$ of a smaller universe (cf. Eq. (2.3) in the proof of Theorem 2.15). The fact that this suffices to embed the Kahr-Class into $\mathsf{LFD}^=$ is crucial, since it can be shown that $\mathsf{LFD}^=$ cannot define cartesian products; neither within relations, in the sense that

$$\{(a,b) \mid \exists \mathbf{c} \colon (a,b,\mathbf{c}) \in R^{\mathbf{M}}\} \text{ is a cartesian product,}$$

nor within the team itself, in the sense that $T(x,y)$ is cartesian. The counterexamples can be found in the appendix; see Examples A.1 and A.2.

Unlike many logics with a notion of bisimulation, $\mathsf{LFD}$ is not invariant under disjoint unions of bisimilar models, because of the atom $D_{\varnothing}x$ which states that $x$ is a constant variable. Indeed, if $x$ is constant in two dependence models with disjoint universes, then clearly $x$ will no longer be constant in their disjoint union. We will come back to this in Section 4.2 where we compare $\mathsf{LFD}$ to the guarded fragment $\mathsf{GF}$ and show that this fact is enough to prove that under reasonable assumptions towards the first-order translation, $\mathsf{LFD}$ is not embeddable into such guarded fragments, see Proposition 4.47. The following weaker proposition and corollary are still helpful in demonstrating $\mathsf{LFD}$'s inability to reason about quantities.

**Proposition 3.17.** Consider some distinguished dependence model $\mathbf{M}, s$ and a copy $\mathbf{M}'$ of $\mathbf{M}$ such that $\mathbf{M}$ and $\mathbf{M}'$ are disjoint except for values which are taken by a variable that is constant throughout the whole team. Then $(\mathbf{M} \cup \mathbf{M}'), s$[5] is still distinguished, and $\mathbf{M}, s \sim_{\mathsf{LFD}} (\mathbf{M} \cup \mathbf{M}'), s$.

*Proof.* The bisimulation relates each assignment in the original model with itself and its copy, the latter two being in $T_{\mathbf{M} \cup \mathbf{M}'}$. They key point is that if $s \in T_{\mathbf{M}}$ and $t' \in T_{\mathbf{M}'}$, then $s =_X t'$ can only happen if all $x \in X$ are constant within both teams. The details can be found in Proposition A.3 in the appendix. ☐

It follows from this that except for trivial cases involving constant variables, $\mathsf{LFD}$ cannot define upper bounds on the size of the universe or the number of values some variable takes. The following corollary makes this precise.

[5]The union $\mathbf{M} \cup \mathbf{M}'$ has the universe $M \cup M'$, relations $R^{\mathbf{M}} \cup R^{\mathbf{M}'}$, and the team $T_{\mathbf{M}} \cup T_{\mathbf{M}'}$.

**Corollary 3.18.** There does not exist an $n \geq 2$ and a satisfiable $\varphi \in \mathsf{LFD}$ such that for every pointed dependence model $\mathbf{M}, s$ with $\mathbf{M}, s \models \varphi$ we have $|T_{\mathbf{M}}(x)| \leq n$, except if $\varphi$ already enforces $|T_{\mathbf{M}}(x)| = 1$, i.e. that $x$ is constant.

*Proof.* Assume that such an $n$ and $\varphi$ exist where $\varphi$ does not enforce $|T_{\mathbf{M}}(x)| = 1$. Then there is some model $\mathbf{M}, s \models \varphi$ with team $T$ such that $|T(x)| \geq 2$. In particular $\mathbf{M}, s \models \neg D_{\varnothing} x$. By Example 3.14 we can assume without loss of generality that $\mathbf{M}$ is distinguished (we can find a distinguished $\mathbf{N}, t$ with $\mathbf{N}, t \equiv_{\mathsf{LFD}} \mathbf{M}, s$, and hence $\mathbf{N}, t \models \varphi \wedge \neg D_{\varnothing} x$ as well). Applying Proposition 3.17 then yields a new (still distinguished) model $\mathbf{M}^1, s$ with team $T^1$ such that $\mathbf{M}^1, s \sim_{\mathsf{LFD}} \mathbf{M}, s$ and $|T^1(x)| = 2|T(x)|$. From here we can define $\mathbf{M}^2, \mathbf{M}^3, \ldots$ inductively, and after finitely many steps reach some $\mathbf{M}^k$ with $|T^k(x)| = 2^k |T(x)| > n$. But we also have $\mathbf{M}, s \sim_{\mathsf{LFD}} \mathbf{M}^1, s \sim_{\mathsf{LFD}} \cdots \sim_{\mathsf{LFD}} \mathbf{M}^k, s$, implying $\mathbf{M}, s \equiv_{\mathsf{LFD}} \mathbf{M}^k, s$ by Theorem 3.12 and hence $\mathbf{M}^k, s \models \varphi$. Contradiction. □

Notice that this is stronger than the usual undefinability-statement. By this corollary, LFD cannot even *enforce* such an upper bound to hold, i.e. that all models of $\varphi$ satisfy the upper bound, without requiring the converse implication. Lower bounds can be defined though; for example, the sentence $\mathsf{E}(Rx) \wedge \mathsf{E}(\neg Rx)$ is satisfiable and enforces $|T(x)| \geq 2$.

Because we heavily relied on distinguishedness, these results can not be adapted to $\mathsf{LFD}^=$. In fact, as an easy counterexample we have the following $\mathsf{LFD}^=$-sentence that enforces $|T(x)| \leq n$:

$$\varphi_n = \bigwedge_{i=1}^{n} D_{\varnothing} x_i \wedge \mathsf{A}\left(\bigvee_{i=1}^{n} x = x_i\right).$$

As innocuous as Corollary 3.18 may seem, if it were not true, so for example we could define some $\varphi_2 \in \mathsf{LFD}$ which reasonably enforces $|T_{\mathbf{M}}(x)| \leq 2$, then we would already be able to define an infinity axiom for LFD and settle the question of whether LFD has the finite model property (which, to the author's best knowledge, is still open at the point of writing this thesis). We formalize this later in Remark 6.4 and Proposition 6.5.

**Fact 3.19.** We can show that even $\mathsf{LFD}^=$ cannot define non-trivial upper bounds on the number of assignments in some $=_X$-class that satisfy some relational atom. By non-trivial we mean that $X$ should be a proper subset of $V$ (the $=_V$ class clearly contains only 1 assignment) and that $n \geq 1$. For the details, see Example A.4 in the appendix.

# Chapter 4

# Comparison with other Logics

In this section we compare $\mathsf{LFD}$ with first-order logic, guarded fragments of first-order logic, and logics with team semantics. From here on we will not concern ourselves with infinitary logics, so we fix $(\tau, V)$ to be a finite type, and $\mathbf{v}$ to be a tuple enumerating the variables in $V$. Before we start with first-order translations of $\mathsf{LFD}$ in the next subsection, we prove the following analogue of the well-known Scott normal form for $\mathsf{FO}^2$.

**Proposition 4.1** (Scott normal form for $\mathsf{LFD}$)**.** Let $\mathcal{L}$ denote either $\mathsf{LFD}$ or $\mathsf{LFD}^=$. For $\psi \in \mathcal{L}(\tau, V)$ there is a vocabulary $\tau^+ \supseteq \tau$ and $V^+ := V \uplus \{c\}$ for a fresh $c$ so that one can construct in polynomial time a formula $\varphi \in \mathcal{L}(\tau^+, V^+)$ of the form

$$\varphi \;=\; \alpha \;\wedge\; \mathsf{A}\beta \;\wedge\; \bigwedge_j \mathsf{A}\,\mathsf{E}_{X_j}\,\gamma_j \;\wedge\; \bigwedge_k \mathsf{A}(R_k\mathbf{v} \leftrightarrow D_{X_k}y_k) \;\wedge\; D_\varnothing c$$

such that the following holds:

1. $\alpha, \beta$ and all $\gamma_j$ are of quantifier rank 0 and contain no dependence atoms, i.e. they are boolean combinations of relations (and equalities if $\mathcal{L} = \mathsf{LFD}^=$).

2. $\varphi \models \psi$, meaning $\mathbf{M}, s \models \varphi$ implies $\mathbf{M}, s \models \psi$ for all $\mathbf{M}, s \in \mathcal{DEP}[\tau^+, V^+]$.

3. For all $\mathbf{M}, s \in \mathcal{DEP}[\tau, V]$ with $\mathbf{M}, s \models \psi$, we can find an expansion $\mathbf{M}^+, s^+$ to the type $(\tau^+, V^+)$ such that $\mathbf{M}^+, s^+ \models \varphi$.

The last two conditions guarantee that $\psi$ and $\varphi$ are satisfiable over the same universes. Also note that $\mathsf{qr}(\varphi) \leq 2$.

*Proof.* In the following, we will write $\varphi \rightsquigarrow \psi$ if $\varphi \models \psi \wedge D_\varnothing c$ and every model of $\psi$ can be expanded to a model of $\varphi$. Remember that $\mathsf{E}_X\,\varphi = \neg\,\mathsf{D}_X\,\neg\varphi$. We start by $\psi' := \psi[\mathsf{D}_\varnothing \mapsto \mathsf{D}_c] \wedge D_\varnothing c$, so $\psi'$ is the formula one obtains by replacing each occurrence of $\mathsf{D}_\varnothing$ in $\psi$ with $\mathsf{D}_c$ and adding the condition that $c$ is constant. Clearly $\psi' \rightsquigarrow \psi$, and $\psi'$ contains no dependence quantifiers $\mathsf{D}_X$ (or $\mathsf{E}_X$) with $X = \varnothing$. This transformation $\psi \mapsto \psi'$ can clearly be done in polynomial time.

Choose for each dependence atom $D_X y$ occurring in $\psi$ a fresh $|V|$-ary relation symbol $R_{X,y}$, and let $\psi'[D_X y \mapsto R_{X,y}\mathbf{v}]$ be the formula obtained from $\psi'$ by replacing all these $D_X y$ with their corresponding atoms $R_{X,y}\mathbf{v}$. Set

$$\vartheta := \psi'[D_X y \mapsto R_{X,y}\mathbf{v}] \wedge \bigwedge_{D_X y \text{ in } \psi} \mathsf{A}(R_{X,y}\mathbf{v} \leftrightarrow D_X y).$$

The attached global conditions for the $R_{X,y}\mathbf{v}$ guarantee that $\vartheta \rightsquigarrow \psi'$ and hence by transitivity $\vartheta \rightsquigarrow \psi$. Moreover, the number of dependence atoms in $\psi$ is linearly bounded in $|\psi|$, and so the above transformation $\psi' \mapsto \vartheta$ can be carried out in polynomial time.

Now we iterate the following transformation until $\vartheta$ has the desired form. Choose some subformula $\mathsf{D}_X\,\eta$ of $\vartheta$ where $X \neq \varnothing$, $\mathsf{qr}(\eta) = 0$, and $\eta$ contains no dependence atoms. This should not be one of the global conditions we attached at an earlier step, and rather stem from a subformula of the original $\psi$. Let $\mathbf{x}$ be a tuple enumerating $X$ and $R$ a fresh $|X|$-ary relation symbol. Define $\vartheta' := \vartheta[\mathsf{D}_X\,\eta \mapsto R\mathbf{x}]$ by replacing each occurrence of $\mathsf{D}_X\,\eta$ in $\vartheta$ by $R\mathbf{x}$. Note that since $\mathrm{Free}(R\mathbf{x}) = X$, the atom $R\mathbf{x}$ holds at an assignment $s$ if and only if it holds at each assignment in the $=_X$-class of $s$. From this we obtain the equivalences

$$\mathsf{E}_X\, R\mathbf{x} \equiv R\mathbf{x} \qquad \text{and} \qquad \neg R\mathbf{x} \vee \mathsf{D}_X\,\eta \equiv \mathsf{D}_X(\neg R\mathbf{x} \vee \eta).$$

Moreover, $\mathsf{A}\,\mathsf{D}_X\,\varphi' \equiv \mathsf{A}\varphi'$ for all $\varphi'$, $\mathsf{A}$ distributes over $\wedge$, and $\mathsf{E}_X$ distributes over $\vee$. Hence

$$\begin{aligned}
& \vartheta' \wedge \mathsf{A}(R\mathbf{x} \leftrightarrow \mathsf{D}_X\,\eta) \\
\equiv\ & \vartheta' \wedge \mathsf{A}(\neg R\mathbf{x} \vee \mathsf{D}_X\,\eta) \wedge \mathsf{A}(\neg\,\mathsf{D}_X\,\eta \vee R\mathbf{x}) \\
\equiv\ & \vartheta' \wedge \mathsf{A}\,\mathsf{D}_X(\neg R\mathbf{x} \vee \eta) \wedge \mathsf{A}(\mathsf{E}_X(\neg\eta) \vee \mathsf{E}_X\, R\mathbf{x}) \\
\equiv\ & \vartheta' \wedge \mathsf{A}\,\mathsf{D}_X(R\mathbf{x} \to \eta) \wedge \mathsf{A}\,\mathsf{E}_X(\eta \to R\mathbf{x}) \\
\equiv\ & \vartheta' \wedge \mathsf{A}(R\mathbf{x} \to \eta) \wedge \mathsf{A}\,\mathsf{E}_X(\eta \to R\mathbf{x}).
\end{aligned}$$

Call the last sentence $\vartheta''$. The above yields

$$\vartheta'' \equiv \vartheta' \wedge \mathsf{A}(R\mathbf{x} \leftrightarrow \mathsf{D}_X\,\eta) \rightsquigarrow \vartheta \rightsquigarrow \psi \qquad \text{and thus} \qquad \vartheta'' \rightsquigarrow \psi.$$

Moreover, $\vartheta''$ is one step closer to our desired form, since $\mathsf{A}$ distributes over conjunction. Lastly it is again easy to see that a single such transformation $\vartheta \mapsto \vartheta''$ can be carried out in polynomial time, and that we will only have to do polynomially many of these to reach our desired normal form $\varphi$. □

**Remark 4.2.** If we want the normal form to be a sentence, we can replace $\alpha$ by $\mathsf{E}\,\alpha$ in $\varphi$. Since $\mathsf{E}\,\alpha \equiv \mathsf{A}\mathsf{E}\,\alpha$, this would yield an $\mathcal{L}(\tau^+, V)$-sentence $\varphi'$ of the form

$$\varphi' \;=\; \mathsf{A}\beta \;\wedge\; \bigwedge_j \mathsf{A}\mathsf{E}_{X_j}\gamma_j \;\wedge\; \bigwedge_k \mathsf{A}(R_k\mathbf{v} \leftrightarrow D_{X_k}y_k) \;\wedge\; D_{\varnothing}c$$

such that $\beta$ and all $\gamma_j$ have the same constraints as before and:

1. $\varphi' \models \mathsf{E}\,\psi$.
2. For all $\mathbf{M}, s \in \mathcal{DEP}[\tau, V]$ with $\mathbf{M}, s \models \psi$ we can find an expansion $\mathbf{M}^+$ to the type $(\tau^+, V^+)$ such that $\mathbf{M}^+ \models \varphi'$.

Hence $\psi$ and $\varphi'$ are still satisfiable over the same universes.

The reason we excluded dependence atoms from $\alpha, \beta$ and the $\gamma_j$ is that they are usually the most complex part of LFD to translate into other logics, as can be seen in the examples of first-order translations discussed below. It should also become clear over the course of this thesis that the dependence atoms are essentially the only difference between LFD and propositional modal logic over a special class of frames. We therefore believe that the normal form as stated above is easier to work with as opposed to allowing dependence atoms to occur anywhere.

## 4.1 First-order translations

We give three examples of possible first-order translations of LFD. We start in Section 4.1.1 by recalling the standard translation defined in [6, Section 3.2]. Then in Section 4.1.2 we give a modal translation of LFD into $\mathsf{FO}^2$ with equivalence relations, over a special class of structures based on the standard relational models introduced in [6, Section 3.3]. A third example, given in Section 4.1.3, can be seen as a combination of the previous two and translates LFD into two-sorted monadic $\mathsf{FO}^2$ with unary functions, again over a certain class of structures.

### 4.1.1 The Standard Translation

We recall the standard translation as described in [6, Section 3.2], stating the translation of structures more explicitly and also defining the corresponding transformation of the classical structures back to dependence models. Remember that $(\tau, V)$ is a finite type and $\mathbf{v}$ an enumeration of $V$. In this context, $\mathcal{DEP}[\tau, V]$ corresponds to the following class of pointed classical structures.

**Definition 4.3** (The class $\mathcal{DFO}$)**.** Let $[\![T]\!]$ be a fresh $|V|$-ary relation symbol and consider $\sigma \coloneqq \tau \uplus \{[\![T]\!]\}$. For the rest of this subsection, this $\sigma$ will be fixed together with $(\tau, V)$. Define

$$\mathcal{DFO}[\sigma] \coloneqq \{\mathfrak{A}, \mathbf{a} \mid \mathfrak{A} \text{ is a } \sigma\text{-structure and } \mathbf{a} \in [\![T]\!]^{\mathfrak{A}}\}.$$

So we explicitly encode the team of a dependence model in the relation $[\![T]\!]$, much in the same way we think of teams when viewing them as a tables; each variable has its own column and the rows correspond to assignments (cf. Example 2.3).

**Definition 4.4.** We have translations between these classes, which for lack of a better name will be denoted as follows:

$$\begin{aligned} \mathsf{T}_{\mathrm{dep} \to \mathrm{dfo}} &\colon \mathcal{DEP}[\tau, V] \ \to \ \mathcal{DFO}[\sigma], \\ \mathsf{T}_{\mathrm{dfo} \to \mathrm{dep}} &\colon \mathcal{DFO}[\sigma] \ \to \ \mathcal{DEP}[\tau, V]. \end{aligned}$$

For a $(\tau, V)$ dependence model $\mathbf{M}, s$ with underlying $\tau$-structure $\mathfrak{M}$, let

$$[\![T]\!]^{\mathfrak{M}} \coloneqq T_{\mathbf{M}}(\mathbf{v}) \qquad \text{and} \qquad \mathsf{T}_{\mathrm{dep} \to \mathrm{dfo}}(\mathbf{M}, s) \coloneqq (\mathfrak{M}, [\![T]\!]^{\mathfrak{M}}), s(\mathbf{v})$$

so the corresponding first-order model of $\mathbf{M}$ is an expansion of $\mathfrak{M}$ by a suitably interpreted $[\![T]\!]$. Conversely, for $\mathfrak{A}, \mathbf{a} \in \mathcal{DFO}[\sigma]$ we set

$$T \coloneqq \{\mathbf{v} \mapsto \mathbf{b} \mid \mathbf{b} \in [\![T]\!]^{\mathfrak{A}}\} \qquad \text{and} \qquad \mathsf{T}_{\mathrm{dfo} \to \mathrm{dep}}(\mathfrak{A}, \mathbf{a}) \coloneqq (\mathfrak{A} \restriction \tau,\ T),\ (\mathbf{v} \mapsto \mathbf{a})$$

so we see that the corresponding dependence model of $\mathfrak{A}$ is the reduct of $\mathfrak{A}$ to $\tau$, together with the team $T$ obtained from $[\![T]\!]^{\mathfrak{A}}$. This shows that both translations are well-defined and in fact inverses of each other:

$$\mathsf{T}_{\mathrm{dep} \to \mathrm{dfo}} \circ \mathsf{T}_{\mathrm{dfo} \to \mathrm{dep}} = \mathrm{id} \qquad \text{and} \qquad \mathsf{T}_{\mathrm{dfo} \to \mathrm{dep}} \circ \mathsf{T}_{\mathrm{dep} \to \mathrm{dfo}} = \mathrm{id}\,.$$

Instead of pointed $\sigma$-structures, we could also consider $\sigma$-structures with $|V|$ extra constant symbols, interpreted to represent the current assignment.

**Definition 4.5** (The standard translation, [6, Definition 3.9])**.** The standard translation of formulae $\mathsf{tr}_{\mathrm{st}} \colon \mathsf{LFD}(\tau, V) \to \mathsf{FO}(\sigma)$ is given by:

1. $\mathsf{tr}_{\mathrm{st}}(R\mathbf{x}) = R\mathbf{x}$ for $R \in \tau$.

2. $\mathsf{tr}_{\mathrm{st}}$ commutes with boolean connectives.[1]

[1] By this we mean that $\mathsf{tr}_{\mathrm{st}}(\neg\varphi) = \neg\, \mathsf{tr}_{\mathrm{st}}(\varphi)$ and $\mathsf{tr}_{\mathrm{st}}(\varphi \wedge \psi) = \mathsf{tr}_{\mathrm{st}}(\varphi) \wedge \mathsf{tr}_{\mathrm{st}}(\psi)$.

3. $\mathsf{tr}_{\mathrm{st}}(\mathsf{D}_X\,\varphi) = \forall \mathbf{z}([\![T]\!]\mathbf{v} \to \mathsf{tr}_{\mathrm{st}}(\varphi))$ where $\mathbf{z}$ is an enumeration of all variables in $V \setminus X$, so in particular $[\mathbf{z}] \subseteq [\mathbf{v}]$.

4. $\mathsf{tr}_{\mathrm{st}}(D_X y) = \forall \mathbf{z}\forall \mathbf{z}'(([\![T]\!]\mathbf{v} \wedge [\![T]\!]\mathbf{v}[\mathbf{z} \mapsto \mathbf{z}']) \;\to\; y = y')$ where $\mathbf{z}$ is as above, $\mathbf{z}'$ and $y'$ are fresh copies of $\mathbf{z}$ respectively $y$, and $[\![T]\!]\mathbf{v}[\mathbf{z} \mapsto \mathbf{z}']$ is the atom obtained by replacing the variables $\mathbf{z}$ with $\mathbf{z}'$ in $[\![T]\!]\mathbf{v}$.

Note that although we require a fresh copy $V'$ of the set of variables $V$ for the translation of the dependence atoms $D_X y$, the fresh copies are never free in any translated formula. Indeed, LFD-formulae $\varphi$ always have the same free variables as their translations $\mathsf{tr}_{\mathrm{st}}(\varphi)$, cf. Definition 2.10. In particular we can write $\mathsf{tr}_{\mathrm{st}}(\varphi)(\mathbf{v})$ since the free variables of $\mathsf{tr}_{\mathrm{st}}(\varphi)$ are always among $[\mathbf{v}] = V$. This yields the intended evaluation of the $\mathsf{tr}_{\mathrm{st}}(\varphi)$ on the structures $\mathfrak{A}, \mathbf{a} \in \mathcal{DFO}$ pointed by $|V|$-tuples.

**Fact 4.6** ([6, Fact 3.10])**.** If $\varphi \in \mathsf{LFD}(\tau, V)$ and $\mathbf{M}, s \in \mathcal{DEP}[\tau, V]$ then

$$\mathbf{M}, s \models \varphi \qquad \text{iff} \qquad \mathsf{T}_{\mathrm{dep} \to \mathrm{dfo}}(\mathbf{M}, s) \models \mathsf{tr}_{\mathrm{st}}(\varphi).$$

Similarly, if $\mathfrak{A}, \mathbf{a}$ is a pointed $\sigma$-structure in $\mathcal{DFO}[\sigma]$, then

$$\mathfrak{A}, \mathbf{a} \models \mathsf{tr}_{\mathrm{st}}(\varphi) \qquad \text{iff} \qquad \mathsf{T}_{\mathrm{dfo} \to \mathrm{dep}}(\mathfrak{A}, \mathbf{a}) \models \varphi.$$

*Proof.* The first claim follows from a straightforward induction on $\varphi$. For the second claim, use the first and the fact that the translations are inverses of each other. □

From the fact that $\mathcal{DFO}$ is first-order axiomatizable (by $[\![T]\!]\mathbf{v}$) and the standard translation fixes the universe and preserves LFD in the above sense, it follows that LFD inherits compactness and Löwenheim-Skolem properties from FO. Moreover, since the translation is effective, the validities of LFD are recursively enumerable. This was already remarked in [6, Corollary 3.11].

**Remark 4.7.** We mentioned in Section 2.2 (see also [6, Section 7.2]) that the standard translation may easily be lifted to $\mathsf{LFD}^{=}$. Indeed, we can simply set $\mathsf{tr}_{\mathrm{st}}(x = y) = (x = y)$ and everything works out. Clearly the translation can also be adapted to exist between $\mathsf{LFD}_\infty/\mathsf{LFD}_\infty^{=}$ and infinitary FO.

### 4.1.2 A Modal Translation

In [6, Section 3.3] the modal perspective of LFD is highlighted via so-called standard relational models, and two LFD-preserving translations are given; from dependence models to standard relational models and vice versa. We briefly recall these definitions and introduce new notation to formalize the involved classes of structures. As a first-order correspondent to the standard relational models, we define a specific class of structures that interpret a fixed set of binary relations as equivalence relations. Over this class of structures, $\mathsf{LFD}$-formulae are then translated into a fragment of monadic $\mathsf{FO}^2$ with equivalence relations but without equality.

**Definition 4.8** (Standard relational model, [6, Definition 3.12])**.** A $(\tau, V)$ standard relational model $\mathcal{M} = (W, (\sim_x)_{x \in V}, \| \bullet \|)$ has a universe $W$, equivalence relations $\sim_x$ on $W$ for $x \in V$, and a valuation $\| \bullet \| = \| \bullet \|_{\mathcal{M}}$ sending atoms $R\mathbf{x}$ to a set of states $\|R\mathbf{x}\| \subseteq W$, for $R \in \tau$ and fitting tuples $\mathbf{x} \in V^{\mathrm{ar}(R)}$. Auxiliary relations $\sim_X := \bigcap_{x \in X} \sim_x$ are introduced for $X \subseteq V$, where $\sim_\varnothing = W \times W$ is the all-relation. We furthermore require the following condition:

$$\text{if } s \in \|R\mathbf{x}\| \text{ and } s \sim_{[\mathbf{x}]} t, \text{ then also } t \in \|R\mathbf{x}\|. \tag{4.1}$$

The class of $(\tau, V)$ standard relational models is denoted by $\mathcal{SRM}[\tau, V]$.

**Definition 4.9** ($\mathsf{LFD}$-semantics on $\mathcal{SRM}$, [6, Definition 3.13])**.** The semantics of boolean connectives are as usual, together with

$$\begin{array}{lll} \mathcal{M}, s \models R\mathbf{x} & \text{iff} & s \in \|R\mathbf{x}\|_{\mathcal{M}}. \\ \mathcal{M}, s \models D_X y & \text{iff} & t \sim_X s \text{ implies } t \sim_y s \text{ for all } t \in W. \\ \mathcal{M}, s \models \mathsf{D}_X \varphi & \text{iff} & \mathcal{M}, t \models \varphi \text{ for all } t \in W \text{ with } t \sim_X s. \end{array}$$

So the idea is to consider assignments as our atomic objects and preserve the information of "agreement on $X$" by the equivalences $\sim_X$, which makes it clear why we need to require the condition (4.1). Remember our discussion below Definition 2.8, where we gave a first hint at the modal perspective of $\mathsf{LFD}$; we noted that dependence models induce a quasi-Kripke model which has the team as its universe, and modalities $\mathsf{D}_X$ and $\mathsf{E}_X$ behaving respectively like $\Box$ and $\Diamond$ in $\mathsf{ML}$ with the accessibility relations $=_X$. The standard relational models formalize this idea, showing that we can essentially get an "object-free" semantics for $\mathsf{LFD}$. The correspondence to dependence models is given in the following.

**Definition 4.10** ([6, Fact 3.14, Definition 3.15])**.** We have the translations

$$\begin{aligned} \mathsf{T}_{\mathrm{dep} \to \mathrm{srm}} &: \mathcal{DEP}[\tau, V] \to \mathcal{SRM}[\tau, V], \\ \mathsf{T}_{\mathrm{srm} \to \mathrm{dep}} &: \mathcal{SRM}[\tau, V] \to \mathcal{DEP}[\tau, V]. \end{aligned}$$

For a $(\tau, V)$ dependence model $\mathbf{M}, s$ we define the valuation $\| \bullet \|_{\mathbf{M}}$ by

$$\|R\mathbf{x}\|_{\mathbf{M}} := \{ s \in T_{\mathbf{M}} \mid \mathbf{M}, s \models R\mathbf{x} \}$$

so the corresponding standard relational model of $\mathbf{M}, s$ is

$$\mathsf{T}_{\text{dep}\to\text{srm}}(\mathbf{M}, s) \coloneqq (T_{\mathbf{M}},\ (=_x)_{x\in V},\ \|\bullet\|_{\mathbf{M}}),\ s.$$

Conversely, given $\mathcal{N}, t = (W, (\sim_x)_{x\in V}, \|\bullet\|_{\mathcal{N}}) \in \mathcal{SRM}[\tau, V]$ we define the distinguished $(\tau, V)$ dependence model $\mathbf{N}, t^{\sim} = \mathsf{T}_{\text{srm}\to\text{dep}}(\mathcal{N}, t)$ as having

1. universe $N \coloneqq \{(x, [w]_x) \mid w \in W, x \in V\}$ where $[w]_x$ is the $\sim_x$-class of $w$,
2. team $T_{\mathbf{N}} \coloneqq \{w^{\sim} \mid w \in W\}$ with $w^{\sim}(x) \coloneqq (x, [w]_x)$ for $x \in V$,
3. and $n$-ary relations $R \in \tau$ interpreted as

$$R^{\mathbf{N}} \coloneqq \{((x_1, [w]_{x_1}), \dots, (x_n, [w]_{x_n})) \mid (x_1, \dots, x_n) = \mathbf{x} \in V^n \text{ and } w \in \|R\mathbf{x}\|_{\mathcal{N}}\}.$$

Note that $\mathsf{T}_{\text{dep}\to\text{srm}}$ forgets about elements in the universe that are not taken as values by any assignment, and $\mathsf{T}_{\text{srm}\to\text{dep}}$ maps elements that are in the same $\sim_V$-class to the same assignment.

**Fact 4.11** ([6, Facts 3.14, 3.16])**.** If $\varphi \in \mathsf{LFD}(\tau, V)$ and $\mathbf{M}, s \in \mathcal{DEP}[\tau, V]$ then

$$\mathbf{M}, s \models \varphi \qquad \text{iff} \qquad \mathsf{T}_{\text{dep}\to\text{srm}}(\mathbf{M}, s) \models \varphi.$$

Similarly, if $\mathcal{M}, s \in \mathcal{SRM}[\tau, V]$, then

$$\mathcal{M}, s \models \varphi \qquad \text{iff} \qquad \mathsf{T}_{\text{srm}\to\text{dep}}(\mathcal{M}, s) \models \varphi.$$

*Proof.* Both claims follow from a straightforward induction on $\varphi$. ☐

In [29] the decidability and complexity of $\mathsf{FO}^2$ extended with equivalence relations was considered. We want to translate $\mathsf{LFD}$ over standard relational models into a fragment of such an extension of $\mathsf{FO}^2$. Adapting their notation we consider finite relational vocabularies $\tau_0 \uplus \tau_{\text{eq}}$ where $\tau_{\text{eq}}$ is a set of distinguished binary relations; usually $\tau_{\text{eq}} = \{\sim_x \mid x \in V\}$. We denote the class of all $(\tau_0 \uplus \tau_{\text{eq}})$-structures that interpret the relations in $\tau_{\text{eq}}$ as equivalences by $\mathcal{EQ}[\tau_0; \tau_{\text{eq}}]$ and refer to them as equivalence structures.

**Definition 4.12** (The class $\mathcal{EQD}$)**.** Define the vocabulary $\sigma \coloneqq \tau_0 \uplus \tau_{\text{eq}}$ with

$$\tau_0 \coloneqq \{R_{\mathbf{x}} \mid R \in \tau,\ \mathbf{x} \in V^{\text{ar}(R)}\} \qquad \text{and} \qquad \tau_{\text{eq}} \coloneqq \{\sim_x \mid x \in V\}.$$

All $R_{\mathbf{x}}$ are monadic predicates and every $\sim_x$ is a binary relation. For the rest of this subsection, this $\sigma$ will be fixed together with $(\tau, V)$. Similarly to the auxiliary relations $\sim_X$ for standard relational models we define auxiliary notation

$$s \sim_X t \coloneqq \bigwedge_{x \in X} s \sim_x t, \qquad X \subseteq V.$$

To obtain meaningful translations between standard relational models and equivalence structures we need to preserve the special requirement (4.1) of the former sort. Hence define the sentence

$$\psi_\sigma \coloneqq \bigwedge_{R_\mathbf{x} \in \tau_0} \forall s \forall t ((R_\mathbf{x} s \wedge s \sim_{[\mathbf{x}]} t) \;\to\; R_\mathbf{x} t).$$

Finally the class of pointed equivalence structures corresponding to $\mathcal{SRM}[\tau, V]$ is

$$\mathcal{EQD}[\sigma] \coloneqq \{\mathfrak{A}, s \mid \mathfrak{A} \in \mathcal{EQ}[\tau_0; \tau_{\mathrm{eq}}],\; s \in \mathfrak{A},\; \mathfrak{A} \models \psi_\sigma\}.$$

**Definition 4.13** (From $\mathcal{SRM}$ to $\mathcal{EQD}$)**.** We have the translations

$$\mathsf{T}_{\mathrm{srm} \to \mathrm{eqd}} \colon \mathcal{SRM}[\tau, V] \;\to\; \mathcal{EQD}[\sigma],$$
$$\mathsf{T}_{\mathrm{eqd} \to \mathrm{srm}} \colon \mathcal{EQD}[\sigma] \;\to\; \mathcal{SRM}[\tau, V].$$

Both will leave the universe, the point $s$, and the interpretation of the $(\sim_x)_{x \in V}$ unchanged. Given $\mathcal{M}, s \in \mathcal{SRM}[\tau, V]$, its translation $\mathfrak{A}, s = \mathsf{T}_{\mathrm{srm} \to \mathrm{eqd}}(\mathcal{M}, s)$ is then completely specified by

$$R^{\mathfrak{A}}_\mathbf{x} \coloneqq \|R\mathbf{x}\|_\mathcal{M}.$$

Conversely, if $\mathfrak{B}, t \in \mathcal{EQD}[\sigma]$, then $\mathcal{N}, t = \mathsf{T}_{\mathrm{eqd} \to \mathrm{srm}}(\mathfrak{B}, t)$ has the valuation

$$\|R\mathbf{x}\|_\mathcal{N} \coloneqq R^{\mathfrak{B}}_\mathbf{x}.$$

Since satisfaction of $\psi_\sigma$ exactly corresponds to the requirement (4.1) of standard relational models, it is easy to see that these translations are well-defined. Moreover, they are inverses of each other:

$$\mathsf{T}_{\mathrm{srm} \to \mathrm{eqd}} \circ \mathsf{T}_{\mathrm{eqd} \to \mathrm{srm}} = \mathrm{id} \qquad \text{and} \qquad \mathsf{T}_{\mathrm{eqd} \to \mathrm{srm}} \circ \mathsf{T}_{\mathrm{srm} \to \mathrm{eqd}} = \mathrm{id}\,.$$

The modal translation of $\mathsf{LFD}$-formulae is based on the standard translation of $\mathsf{ML}$ into $\mathsf{FO}^2$.

**Definition 4.14** (The modal translation)**.** For fresh variables $s$ and $t$, the modal translation of formulae $\mathsf{tr}_{\mathrm{mod}} \colon \mathsf{LFD}(\tau, V) \to \mathsf{FO}(\sigma)$ is given by $\mathsf{tr}_{\mathrm{mod}}(\varphi) = \mathsf{tr}^s_{\mathrm{mod}}(\varphi)$ as well as

1. $\mathsf{tr}^s_{\mathrm{mod}}(R\mathbf{x}) = R_\mathbf{x} s$ for $R \in \tau$.

2. $\mathsf{tr}^s_{\mathrm{mod}}$ commutes with boolean connectives.

3. $\mathsf{tr}^s_{\mathrm{mod}}(\mathsf{D}_X\,\varphi) = \forall t(t \sim_X s \to \mathsf{tr}^t_{\mathrm{mod}}(\varphi))$.

4. $\mathsf{tr}^s_{\mathrm{mod}}(D_X y) = \forall t(t \sim_X s \to t \sim_y s)$.

Here $\mathsf{tr}^t_{\mathrm{mod}}$ is defined as $\mathsf{tr}^s_{\mathrm{mod}}$ with all $s$ and $t$ interchanged. Note that $\mathsf{tr}_{\mathrm{mod}}(\varphi) \in \mathsf{FO}^2$ has at most one free variable.

**Fact 4.15.** If $\varphi \in \mathsf{LFD}(\tau, V)$ and $\mathcal{M}, s \in \mathcal{SRM}[\tau, V]$ then

$$\mathcal{M}, s \models \varphi \qquad \text{iff} \qquad \mathsf{T}_{\mathrm{srm} \to \mathrm{eqd}}(\mathcal{M}, s) \models \mathsf{tr}_{\mathrm{mod}}(\varphi).$$

Similarly, if $\mathfrak{A}, s \in \mathcal{EQD}[\sigma]$, then

$$\mathfrak{A}, s \models \mathsf{tr}_{\mathrm{mod}}(\varphi) \qquad \text{iff} \qquad \mathsf{T}_{\mathrm{eqd} \to \mathrm{srm}}(\mathfrak{A}, s) \models \varphi.$$

*Proof.* The first claim follows from a straightforward induction on $\varphi$. For the second claim, use the first and the fact that the translations are inverses of each other. □

**Remark 4.16.** Composing the two translations discussed in this subsection yields

$$\begin{aligned} \mathsf{T}_{\mathrm{dep} \to \mathrm{eqd}} &: \mathcal{DEP}[\tau, V] \;\to\; \mathcal{EQD}[\sigma], \\ \mathsf{T}_{\mathrm{eqd} \to \mathrm{dep}} &: \mathcal{EQD}[\sigma] \;\to\; \mathcal{DEP}[\tau, V], \end{aligned}$$

which preserve $\mathsf{LFD}$ via $\mathsf{tr}_{\mathrm{mod}}$ in an analogous fashion to Facts 4.11 and 4.15.

Now one might wonder why we would even consider the class of equivalence structures, and not just translate formulae from $\mathsf{LFD}$ to $\mathsf{FO}$ by tagging on a sentence which states that all $\sim_x$ are equivalence relations. One reason is that it is quite convenient to be able to translate all considered structures in $\mathcal{EQD}$ back to dependence models in $\mathcal{DEP}$. In a larger class there would be structures on which translated $\mathsf{LFD}$-formulae have a well-defined semantics, but which cannot be transformed back to the models we evaluate $\mathsf{LFD}$ over. Furthermore, when translating $\mathsf{LFD}$-formulae in the above fashion, we evidently only need two variables, landing in $\mathsf{FO}^2$. This can be explained by the modal character of $\mathsf{LFD}$; once our viewpoint changes from one assignment to another via some modality $\mathsf{D}_X$ or $\mathsf{E}_X$, we lose all information about our initial position (except for $X$). On the other hand, stating transitivity of a relation requires at least three variables. Now full $\mathsf{FO}^3$ is well-known to be undecidable[2], whereas for $\mathsf{FO}^2$ on equivalence structures there is a lot of literature on the very fine line between decidability and undecidability and various complexity results. For example, it was shown in [29] that

[2]For example, $\mathsf{FO}^3$ contains the Kahr-Class we introduced in Section 2.3.

1. $\mathsf{FO}^2$ with only a single equivalence relation still has the finite model property (FMP), and its satisfiability problem is NEXPTIME-complete.

2. $\mathsf{FO}^2$ with two equivalence relations does not have the FMP any more, but is still decidable in 3-NEXPTIME. A lower bound of 2-NEXPTIME is shown in [28].

3. $\mathsf{FO}^2$ with three equivalences is already undecidable.

The last point actually holds already for a much smaller fragment of $\mathsf{FO}^2$:

**Theorem 4.17** ([29, Corollary 28])**.** Satisfiability and finite satisfiability on the class of structures with three equivalence relations $E_1, E_2, E_3$ are undecidable even for the following fragment of $\mathsf{FO}^2$ in vocabularies $\tau_0$ consisting only of unary predicates: conjunctions of sentences of the form

1. $\forall x \forall y (E_i xy \to \chi(x,y))$, and

2. $\forall x(\alpha(x) \to \exists y(E_i xy \wedge \alpha'(y)))$,

for quantifier-free and equality-free formulae $\chi$ and $\alpha$ and $\alpha'$. This fragment is in particular contained in the two-variable fragment of $\mathsf{GF}$.[3]

Interestingly enough, our translation $\mathsf{T}_{\text{dep} \to \text{eqd}}$ yields equivalence structures whose number of equivalence relations corresponds to the number of variables in the dependence model ($|\tau_{\text{eq}}| = |V|$), i.e. the number of equivalences we consider is unbounded. In view of the above result it seems quite remarkable that $\mathsf{LFD}$ is decidable.

So we know that the fragment outlined in Theorem 4.17 cannot be contained within the fragment one obtains via $\mathsf{tr}_{\text{mod}}$. Indeed, sentences of the first type $\forall x \forall y (E_i xy \to \chi(x,y))$ contain formulae $\chi(x,y)$ that may be arbitrary relational formulae of *two free variables.* Since in the context of the modal translation, variables represent assignments, this would correspond to having access to all relational facts of two distinct assignments at the same time. This is not possible in $\mathsf{LFD}$; there is a single current assignment, and we may change it via modalities $\mathsf{D}_X$ or $\mathsf{E}_X$, but once changed, we lose all information about the original assignment (apart from the $\mathsf{LFD}$-facts about the values of $X$). This also becomes apparent when we adapt our normal form from Proposition 4.1 to the current situation:

**Lemma 4.18.** Let $\psi \in \mathsf{LFD}(\tau, V)$ and $\varphi \in \mathsf{LFD}(\tau^+, V^+)$ be its normal form as defined in Proposition 4.1. Define $\sigma^+$ for $(\tau^+, V^+)$ as $\sigma$ for $(\tau, V)$ in Definition 4.12. Remember that $V^+ = V \uplus \{c\}$ for some fresh variable $c$. Then

[3] We introduce the guarded fragment $\mathsf{GF}$ in Section 4.2.

1. $\sigma \subseteq \sigma^+$.

2. $\mathsf{tr}_{\mathrm{mod}}(\varphi)$ with free variable $s$ has the form

$$\begin{aligned}\mathsf{tr}_{\mathrm{mod}}(\varphi)(s) = {} & \alpha(s) \wedge \forall s \beta(s) \\ & \wedge \bigwedge_j \forall s \exists t (t \sim_{X_j} s \wedge \gamma_j(t)) \\ & \wedge \bigwedge_k \forall s (R_k s \leftrightarrow \forall t (t \sim_{X_k} s \rightarrow t \sim_{y_k} s)) \\ & \wedge \forall t (t \sim_c s),\end{aligned}$$

where $\alpha, \beta$ and the $\gamma_j$ are boolean combinations of monadic relations.

3. $\mathsf{tr}_{\mathrm{mod}}(\varphi) \models \mathsf{tr}_{\mathrm{mod}}(\psi \wedge D_{\varnothing} c)$ over $\mathcal{EQD}$, meaning $\mathfrak{A}, s \models \mathsf{tr}_{\mathrm{mod}}(\varphi)$ implies $\mathfrak{A}, s \models \mathsf{tr}_{\mathrm{mod}}(\psi \wedge D_{\varnothing} c)$ for all $\mathfrak{A}, s \in \mathcal{EQD}[\sigma^+]$.

4. For all $\mathfrak{A}, s \in \mathcal{EQD}[\sigma]$ with $\mathfrak{A}, s \models \mathsf{tr}_{\mathrm{mod}}(\psi)$ we can find an expansion $\mathfrak{A}^+$ of $\mathfrak{A}$ to the vocabulary $\sigma^+$ such that $\mathfrak{A}^+, s \in \mathcal{EQD}[\sigma^+]$ and $\mathfrak{A}^+, s \models \mathsf{tr}_{\mathrm{mod}}(\varphi)$.

5. In particular, $\mathsf{tr}_{\mathrm{mod}}(\psi)$ and $\mathsf{tr}_{\mathrm{mod}}(\varphi)$ are satisfiable over the same universes of structures in $\mathcal{EQD}$.

*Proof.* Most points follow immediately from Proposition 4.1, Definitions 4.12 and 4.14, and Remark 4.16. For (4.), let $\mathbf{M}, s^{\sim} = \mathsf{T}_{\mathrm{eqd} \to \mathrm{dep}}(\mathfrak{A}, s)$. If $\mathfrak{A}, s \models \mathsf{tr}_{\mathrm{mod}}(\psi)$, then $\mathbf{M}, s^{\sim} \models \psi$, and hence we can extend $\mathbf{M}, s^{\sim}$ to $\mathbf{M}^+, s^+$ of type $(\tau^+, V^+)$ such that $\mathbf{M}^+, s^+ \models \varphi$. From the way we defined $\mathsf{T}_{\mathrm{eqd} \to \mathrm{dep}}$ it is clear that we can find a corresponding extension $\mathfrak{A}^+$ of $\mathfrak{A}$ to the vocabulary $\sigma^+$ such that $\mathsf{T}_{\mathrm{eqd} \to \mathrm{dep}}(\mathfrak{A}^+, s) = \mathbf{M}^+, s^+$, and hence $\mathfrak{A}^+, s \models \mathsf{tr}_{\mathrm{mod}}(\varphi)$. Indeed, $\sigma^+ \setminus \sigma$ contains only monadic relations $R_{\mathbf{x}}$ for $R \in \tau^+ \setminus \tau$, together with the new binary relation $\sim_c$, which we interpret to be the all-relation, since $c$ is constant in $\mathbf{M}^+$. □

This gives a normal form for the set of translated LFD-formulae. Inspecting the form of $\mathsf{tr}_{\mathrm{mod}}(\varphi)$ above, we see that arbitrary relational formulae of the form $\chi(s,t)$ never occur. Essentially, the only way in which subformulae of the form $\chi(s,t)$ occur is when $\chi(s,t) = \vartheta(s,t) \wedge \gamma(t)$ where $\vartheta$ is a boolean combination of formulae of the form $s \sim_x t$, and $\gamma$ evidently only has one free variable. We will come back to this at the end of Section 4.2, where we compare the expressive power of LFD and to that of guarded fragments of FO.

### 4.1.3 A Functional Translation

Another view of LFD combines the two previous perspectives by considering two-sorted structures, with one sort being assignments and the other one being their values. Variables $x \in V$ are then viewed as functions $f_x$ from assignments to values. The role of the $=_x$ (or the equivalence relations $\sim_x$ in the modal context) is then taken by agreement under $f_x$.

**Definition 4.19** (The class $\mathcal{FUN}$)**.** Define $\sigma := \tau_0 \uplus \tau_{\text{fun}} \uplus \{[\![T]\!]\}$ with

$$\tau_0 := \{R_{\mathbf{x}} \mid R \in \tau,\ \mathbf{x} \in V^{\text{ar}(R)}\} \qquad \text{and} \qquad \tau_{\text{fun}} := \{f_x \mid x \in V\}$$

where all the $f_x$ are unary functions, $[\![T]\!]$ is a monadic predicate used to distinguish the two sorts, and $\tau_0$ contains only monadic relations. Hence $\tau_0$ is the exact same as defined for $\mathcal{EQD}$ in Definition 4.12. For the rest of this subsection, this $\sigma$ will be fixed together with $(\tau, V)$. Similarly to the auxiliary notation $\sim_X$ of the modal translation we will use the notation

$$f_X(s) = f_X(t) \quad := \quad \bigwedge_{x \in X} f_x(s) = f_x(t), \qquad X \subseteq V.$$

Since assignments are now viewed as plain objects, we need to give the new way of stating agreement on variables its meaning, i.e. we adapt the sentence $\psi_\sigma$ from Definition 4.12 to the vocabulary $\sigma$ used here:

$$\psi_\sigma := \bigwedge_{R_{\mathbf{x}} \in \tau_0} \forall s \forall t((R_{\mathbf{x}} s \wedge f_{[\mathbf{x}]}(s) = f_{[\mathbf{x}]}(t)) \ \rightarrow\ R_{\mathbf{x}} t).$$

Now the class corresponding to $\mathcal{DEP}[\tau, V]$ in the current setting is simply the class of models of $\psi_\sigma$, pointed by elements in the respective interpretation of $[\![T]\!]$:

$$\mathcal{FUN}[\sigma] := \{\mathfrak{A}, s \mid \mathfrak{A} \text{ is a } \sigma\text{-structure, } s \in [\![T]\!]^{\mathfrak{A}}, \text{ and } \mathfrak{A} \models \psi_\sigma\}.$$

**Definition 4.20** (From $\mathcal{DEP}$ to $\mathcal{FUN}$)**.** We have the translations

$$\begin{aligned} \mathsf{T}_{\text{dep}\,\rightarrow\,\text{fun}} &: \mathcal{DEP}[\tau, V] \ \rightarrow\ \mathcal{FUN}[\sigma], \\ \mathsf{T}_{\text{fun}\,\rightarrow\,\text{dep}} &: \mathcal{FUN}[\sigma] \ \rightarrow\ \mathcal{DEP}[\tau, V]. \end{aligned}$$

Given $\mathbf{M}, s \in \mathcal{DEP}[\tau, V]$ with universe $M$, its translation $\mathfrak{A}, s = \mathsf{T}_{\text{dep}\,\rightarrow\,\text{fun}}(\mathbf{M}, s)$ has universe $A := T_{\mathbf{M}} \uplus M$ and is specified by $[\![T]\!]^{\mathfrak{A}} := T_{\mathbf{M}}$,

$$R_{\mathbf{x}}^{\mathfrak{A}} := \{s' \in T_{\mathbf{M}} \mid \mathbf{M}, s' \models R\mathbf{x}\},$$

as well as $f_x^{\mathfrak{A}}(s') := s'(x)$ for all assignments $s' \in T_{\mathbf{M}}$ and $f_x^{\mathfrak{A}}(m) := m$ for all other objects $m \in M$. Conversely, given $\mathfrak{B}, t \in \mathcal{FUN}[\sigma]$ we define the distinguished dependence model $\mathbf{N}, t^{\sim} = \mathsf{T}_{\text{fun}\,\rightarrow\,\text{dep}}(\mathfrak{B}, t)$ as having

1. the universe $N := \{(x, [s]_x) \mid s \in [\![T]\!]^{\mathfrak{B}}, x \in V\}$ where $[s]_x$ is the equivalence

class of $s$ w.r.t. agreement under $f_x^{\mathfrak{B}}$, so $[s]_x := \{s' \in [\![T]\!]^{\mathfrak{B}} \mid f_x^{\mathfrak{B}}(s) = f_x^{\mathfrak{B}}(s')\}$.

2. the team $T_{\mathbf{N}} := \{s^\sim \mid s \in [\![T]\!]^{\mathfrak{B}}\}$ where $s^\sim(x) := (x, [s]_x)$ for $x \in V$.

3. and $n$-ary relations $R \in \tau$ interpreted as

$$R^{\mathbf{N}} := \{((x_1, [s]_{x_1}), \ldots, (x_n, [s]_{x_n})) \mid (x_1, \ldots, x_n) = \mathbf{x} \in V^n \text{ and } s \in R_{\mathbf{x}}^{\mathfrak{B}}\}.$$

The modal translation $\mathsf{tr}_{\mathrm{mod}}$ is easily adapted to the current situation; we need to restrict quantification to the assignments (objects in $[\![T]\!]$) and instead of the equivalences $\sim_x$ use our functions $f_x$ to compare assignments.

**Definition 4.21** (The functional translation)**.** For fresh variables $s$ and $t$, the functional translation of formulae $\mathsf{tr}_{\mathrm{fun}}\colon \mathsf{LFD}(\tau, V) \to \mathsf{FO}(\sigma)$ is given by $\mathsf{tr}_{\mathrm{fun}}(\varphi) = \mathsf{tr}^s_{\mathrm{fun}}(\varphi)$ as well as

1. $\mathsf{tr}^s_{\mathrm{fun}}(R\mathbf{x}) = R_{\mathbf{x}}s$ for $R \in \tau$.

2. $\mathsf{tr}^s_{\mathrm{fun}}$ commutes with boolean connectives.

3. $\mathsf{tr}^s_{\mathrm{fun}}(\mathsf{D}_X\, \varphi) = \forall t(([\![T]\!]t \wedge f_X(t) = f_X(s)) \;\to\; \mathsf{tr}^t_{\mathrm{fun}}(\varphi))$.

4. $\mathsf{tr}^s_{\mathrm{fun}}(D_X y) = \forall t(([\![T]\!]t \wedge f_X(t) = f_X(s)) \;\to\; f_y(t) = f_y(s))$.

Here $\mathsf{tr}^t_{\mathrm{fun}}$ is defined as $\mathsf{tr}^s_{\mathrm{fun}}$ with all $s$ and $t$ interchanged. Note that $\mathsf{tr}_{\mathrm{fun}}(\varphi) \in \mathsf{FO}^2$ has at most one free variable.

**Fact 4.22.** If $\varphi \in \mathsf{LFD}(\tau, V)$ and $\mathbf{M}, s \in \mathcal{DEP}[\tau, V]$ then

$$\mathbf{M}, s \models \varphi \qquad \text{iff} \qquad \mathsf{T}_{\mathrm{dep}\,\to\,\mathrm{fun}}(\mathbf{M}, s) \models \mathsf{tr}_{\mathrm{fun}}(\varphi).$$

Similarly, if $\mathfrak{A}, s$ is a pointed $\sigma$-structure in $\mathcal{FUN}[\sigma]$, then

$$\mathfrak{A}, s \models \mathsf{tr}_{\mathrm{fun}}(\varphi) \qquad \text{iff} \qquad \mathsf{T}_{\mathrm{fun}\,\to\,\mathrm{dep}}(\mathfrak{A}, s) \models \varphi.$$

*Proof.* Both claims follow from a straightforward induction on $\varphi$, analogously to Fact 4.11. ☐

### 4.1.4 General First-Order Translations

With the three example translations of Sections 4.1.1 to 4.1.3 in mind, we will now distill their essence to define reasonable first-order translations of $\mathsf{LFD}$ in general.

**Definition 4.23** (Good first-order translation of $\mathsf{LFD}$)**.** Given a finite type $(\tau, V)$, assume we have

1. some $n \in \mathbb{N}$ and new finite signature $\sigma$.
2. an $\mathsf{FO}(\sigma)$-definable class of (pointed) $\sigma$-structures

   $$\mathcal{C}[\sigma] := \{\mathfrak{A}, \mathbf{a} \mid \mathfrak{A} \text{ is a } \sigma\text{-structure, } \mathbf{a} \text{ an } n\text{-tuple in } \mathfrak{A}, \text{ and } \mathfrak{A} \models \Theta(\mathbf{a})\}.$$

   for some fixed set $\Theta(\mathbf{x})$ of formulae $\varphi(\mathbf{x}) \in \mathsf{FO}(\sigma)$.
3. translations between the classes

   $$\mathsf{F}\colon \mathcal{DEP}[\tau, V] \to \mathcal{C}[\sigma] \qquad \text{and} \qquad \mathsf{G}\colon \mathcal{C}[\sigma] \to \mathcal{DEP}[\tau, V].$$

   If $\mathsf{G}(\mathfrak{A}, \mathbf{a}) = \mathbf{M}, s$, then we want $\mathbf{M}$ to only depend on $\mathfrak{A}$, and not on $\mathbf{a}$. We write $\mathsf{G}(\mathfrak{A}) := \mathbf{M}$. Similarly, if $\mathfrak{A}$ is clear from context, we just write $\mathsf{G}(\mathbf{a})$, so

   $$\mathsf{G}(\mathfrak{A}, \mathbf{a}) = \mathsf{G}(\mathfrak{A}), \mathsf{G}(\mathbf{a}).$$

   Likewise for $\mathsf{F}$. For $\mathfrak{A}, \mathbf{a} \in \mathcal{C}[\sigma]$, we define

   $$\mathrm{Team}(\mathfrak{A}) := \{\mathbf{b} \in A^n \mid \mathfrak{A}, \mathbf{b} \in \mathcal{C}[\sigma]\} = \{\mathbf{b} \in A^n \mid \mathfrak{A} \models \Theta(\mathbf{b})\}.$$

   From the above, we see that $\mathsf{G}$ induces a map $\mathsf{G}\colon \mathrm{Team}(\mathfrak{A}) \to T_{\mathsf{G}(\mathfrak{A})}$. We require this to be *surjective for every fixed* $\mathfrak{A}, \mathbf{a} \in \mathcal{C}[\sigma]$. We do not need an analogous requirement for $\mathsf{F}$.
4. a translation of formulae

   $$\mathsf{tr}\colon \mathsf{LFD}(\tau, V) \to \mathsf{FO}(\sigma)$$

   that commutes with boolean connectives and produces formulae with at most $n$ free variables (so that we can evaluate translated formulae on the structures of $\mathcal{C}[\sigma]$ that are pointed by $n$-tuples). Furthermore, we require that for every $\varphi \in \mathsf{LFD}(\tau, V)$ and $\mathbf{M}, s \in \mathcal{DEP}[\tau, V]$ we have

   $$\mathbf{M}, s \models \varphi \qquad \text{iff} \qquad \mathsf{F}(\mathbf{M}, s) \models \mathsf{tr}(\varphi)$$

   and dually for all $\mathfrak{A}, \mathbf{a} \in \mathcal{C}[\sigma]$ it holds that

   $$\mathfrak{A}, \mathbf{a} \models \mathsf{tr}(\varphi) \qquad \text{iff} \qquad \mathsf{G}(\mathfrak{A}, \mathbf{a}) \models \varphi.$$
5. for every $X \subseteq V$ a formula $\vartheta_X(\mathbf{x}, \mathbf{y}) \in \mathsf{FO}(\sigma)$ with the interpretation that for

all structures $\mathfrak{A}, \mathbf{a} \in \mathcal{C}[\sigma]$ and $n$-tuples $\mathbf{b}, \mathbf{c}$ in $\mathfrak{A}$

$$\mathfrak{A} \models \vartheta_X(\mathbf{b}, \mathbf{c}) \quad \text{iff} \quad \mathbf{b}, \mathbf{c} \in \text{Team}(\mathfrak{A}) \quad \text{and} \quad \mathsf{G}(\mathbf{b}) =_X \mathsf{G}(\mathbf{c}).$$

Then we say that $(\mathcal{C}[\sigma], \mathsf{F}, \mathsf{G}, \mathsf{tr}, (\vartheta_X)_{X \subseteq V})$ is a good translation of $\mathsf{LFD}$ over $\mathcal{DEP}[\tau, V]$ to $\mathsf{FO}$ over $\mathcal{C}[\sigma]$.

Again we could just as well use constants instead of considering pointed structures. A consequence of requiring the existence of a "back"-translation $\mathsf{G}$ is that good translations yield a reduction for the satisfiability problem in the following sense.

**Fact 4.24.** Let $(\mathcal{C}[\sigma], \mathsf{F}, \mathsf{G}, \mathsf{tr}, (\vartheta_X)_{X \subseteq V})$ be a good translation of $\mathsf{LFD}$ over $\mathcal{DEP}[\tau, V]$ to $\mathsf{FO}$ over $\mathcal{C}[\sigma]$. Then $\varphi$ is satisfiable over $\mathcal{DEP}[\tau, V]$ if and only if $\mathsf{tr}(\varphi)$ is satisfiable over $\mathcal{C}[\sigma]$.

Another assumption could be that $\mathsf{G}$ and $\mathsf{F}$ preserve finite models, so that a finite model property of the first-order fragment corresponding to $\mathsf{LFD}$ over $\mathcal{C}$ would immediately yield a finite model property of $\mathsf{LFD}$; we come back to this in Fact 6.3. Now we verify that the three previously discussed translations of Sections 4.1.1 to 4.1.3 are good in the sense of the above definition.

**Example 4.25** (The standard translation is good)**.** Remember that $\mathbf{v}$ denotes a fixed enumeration of $V$. Let $\mathbf{v}'$ be a fresh copy of $\mathbf{v}$. In the setting of Definition 4.23, the standard translation is specified by

1. $n = |V|$ and $\sigma = \tau \uplus \{[\![T]\!]\}$ where $[\![T]\!]$ is an $n$-ary relation symbol,
2. $\Theta(\mathbf{v}) = \{[\![T]\!]\mathbf{v}\}$, hence $\mathcal{C}[\sigma] = \mathcal{DFO}[\sigma]$,
3. $\mathsf{F} = \mathsf{T}_{\text{dep} \to \text{dfo}}$ and $\mathsf{G} = \mathsf{T}_{\text{dfo} \to \text{dep}}$,
4. $\mathsf{tr} = \mathsf{tr}_{\text{st}}$,
5. $\vartheta_X(\mathbf{v}, \mathbf{v}') = [\![T]\!]\mathbf{v} \wedge [\![T]\!]\mathbf{v}' \wedge \bigwedge_{v \in X} v = v'$.

It now follows from Definitions 4.3 to 4.5 and Fact 4.6 that all requirements for this to be a good translation are met.

**Example 4.26** (The modal translation is good)**.** In the setting of Definition 4.23, the modal translation is specified by

1. $n = 1$ and $\sigma = \tau_0 \uplus \tau_{\text{eq}}$ as defined in Definition 4.12,
2. $\Theta(s) = \{\psi_\sigma\} \cup \{$"$\sim_v$ is an equivalence relation" $\mid v \in V\}$, so $\mathcal{C}[\sigma] = \mathcal{EQD}[\sigma]$,

3. $\mathsf{F} = \mathsf{T}_{\text{dep} \to \text{eqd}}$ and $\mathsf{G} = \mathsf{T}_{\text{eqd} \to \text{dep}}$,

4. $\mathsf{tr} = \mathsf{tr}_{\text{mod}}$,

5. $\vartheta_X(s,t) = s \sim_X t = \bigwedge_{x \in X} s \sim_x t$.

It now follows from Fact 4.11, Definitions 4.12 to 4.14, Fact 4.15, and Remark 4.16 that all requirements for this to be a good translation are met.

**Example 4.27** (The functional translation is good)**.** In the setting of Definition 4.23, the functional translation is specified by

1. $n = 1$ and $\sigma = \tau_0 \uplus \tau_{\text{fun}} \uplus \{[\![T]\!]\}$ as defined in Definition 4.19, with $[\![T]\!]$ being a monadic predicate,

2. $\Theta(s) = \{\psi_\sigma,\ [\![T]\!]s\}$, so $\mathcal{C}[\sigma] = \mathcal{FUN}[\sigma]$,

3. $\mathsf{F} = \mathsf{T}_{\text{dep} \to \text{fun}}$ and $\mathsf{G} = \mathsf{T}_{\text{fun} \to \text{dep}}$,

4. $\mathsf{tr} = \mathsf{tr}_{\text{fun}}$,

5. $$\begin{aligned}\vartheta_X(s,t) &= [\![T]\!]s \wedge [\![T]\!]t \wedge (f_X(s) = f_X(t)) \\ &= [\![T]\!]s \wedge [\![T]\!]t \wedge \bigwedge_{x \in X} (f_x(s) = f_x(t)).\end{aligned}$$

It now follows from Definitions 4.19 to 4.21 and Fact 4.22 that all requirements for this to be a good translation are met.

**Remark 4.28.** Because of the way we defined the standard and functional translations, we can adapt our Scott normal form Proposition 4.1 to them in the same way we did for the modal translation in Lemma 4.18, although we will not go into the details here.

For our expressive completeness result in the next subsection we need to transfer some rudimentary logical notions of $\mathsf{LFD}$ via our good translations. Henceforth, until the end of this subsection, we fix some good translation $(\mathcal{C}[\sigma], \mathsf{F}, \mathsf{G}, \mathsf{tr}, (\vartheta_X)_{X \subseteq V})$ of $\mathsf{LFD}$ over $\mathcal{DEP}[\tau, V]$ to $\mathsf{FO}$ over $\mathcal{C}[\sigma]$. Note that $(\tau, V)$ must be finite.

**Definition 4.29** ($\mathsf{LFD}$-theory and $\mathsf{LFD}$-equivalence)**.** Define the rank-$k$ $\mathsf{LFD}$-theory of $\mathfrak{A}, \mathbf{a} \in \mathcal{C}[\sigma]$ as the set of translated $\mathsf{LFD}(\tau, V)$ formulae of quantifier rank at most $k$ that are satisfied in $\mathfrak{A}, \mathbf{a}$, so for $k \in \mathbb{N}$

$$\mathrm{Th}^k_{\mathsf{LFD}}(\mathfrak{A}, \mathbf{a}) := \{\mathsf{tr}(\varphi) \mid \varphi \in \mathsf{LFD}(\tau, V),\ \mathsf{qr}(\varphi) \leq k,\ \mathfrak{A}, \mathbf{a} \models \mathsf{tr}(\varphi)\},$$

as well as $\mathrm{Th}_{\mathsf{LFD}}(\mathfrak{A}, \mathbf{a})$ without the quantifier restriction. Furthermore, let

$$\mathfrak{A}, \mathbf{a} \equiv^k_{\mathsf{LFD}} \mathfrak{B}, \mathbf{b} \qquad \text{iff} \qquad \mathrm{Th}^k_{\mathsf{LFD}}(\mathfrak{A}, \mathbf{a}) = \mathrm{Th}^k_{\mathsf{LFD}}(\mathfrak{B}, \mathbf{b}).$$

Note that $\mathfrak{A}, \mathbf{a}$ and $\mathfrak{B}, \mathbf{b}$ have the same rank-$k$ $\mathsf{LFD}$-theory iff their corresponding dependence models are $\mathsf{LFD}$-equivalent up to rank $k$:

$$\mathfrak{A}, \mathbf{a} \equiv^k_{\mathsf{LFD}} \mathfrak{B}, \mathbf{b} \qquad \text{iff} \qquad \mathsf{G}(\mathfrak{A}, \mathbf{a}) \equiv^k_{\mathsf{LFD}} \mathsf{G}(\mathfrak{B}, \mathbf{b}).$$

Likewise $\mathbf{M}, s \equiv^k_{\mathsf{LFD}} \mathbf{N}, t$ iff $\mathsf{F}(\mathbf{M}, s) \equiv^k_{\mathsf{LFD}} \mathsf{F}(\mathbf{N}, t)$ for all $(\tau, V)$ dependence models $\mathbf{M}, s$ and $\mathbf{N}, t$. We make analogous definitions and observations for $\equiv_{\mathsf{LFD}}$.

**Notation 4.30.** Let $\varphi, \psi \in \mathsf{FO}(\sigma)$ and $\Psi \subseteq \mathsf{FO}(\sigma)$.

- We write $\varphi \equiv_{\mathcal{C}} \psi$ if $\varphi$ and $\psi$ are equivalent over $\mathcal{C}[\sigma]$, meaning that for all $\mathfrak{A}, \mathbf{a} \in \mathcal{C}[\sigma]$ we have $\mathfrak{A}, \mathbf{a} \models \varphi$ iff $\mathfrak{A}, \mathbf{a} \models \psi$.
- Similarly, we write $\Psi \models_{\mathcal{C}} \psi$ for entailment restricted to $\mathcal{C}[\sigma]$, namely that for all $\mathfrak{A}, \mathbf{a} \in \mathcal{C}[\sigma]$ with $\mathfrak{A}, \mathbf{a} \models \Psi$ we have $\mathfrak{A}, \mathbf{a} \models \psi$.

We adapt our notion of bisimulation and its approximations as defined in Definitions 3.1 and 3.4 to the class $\mathcal{C}[\sigma]$ as follows.

**Definition 4.31** (Translated bisimulation)**.** We define bisimulation between two $\sigma$-structures $\mathfrak{A}, \mathfrak{B}$ in $\mathcal{C}[\sigma]$ as a binary relation $Z \subseteq \mathrm{Team}(\mathfrak{A}) \times \mathrm{Team}(\mathfrak{B})$ such that for all $(\mathbf{a}, \mathbf{b}) \in Z$ we have

- $\mathfrak{A}, \mathbf{a} \equiv^0_{\mathsf{LFD}} \mathfrak{B}, \mathbf{b}$.
- (back) For all $\mathbf{b}' \in \mathrm{Team}(\mathfrak{B})$ and (finite) $X \subseteq V$ with $\mathfrak{B} \models \vartheta_X(\mathbf{b}', \mathbf{b})$, there is an $\mathbf{a}' \in \mathrm{Team}(\mathfrak{A})$ with $(\mathbf{a}', \mathbf{b}') \in Z$ and $\mathfrak{A} \models \vartheta_X(\mathbf{a}', \mathbf{a})$.
- (forth) For all $\mathbf{a}' \in \mathrm{Team}(\mathfrak{A})$ and (finite) $X \subseteq V$ with $\mathfrak{A} \models \vartheta_X(\mathbf{a}', \mathbf{a})$, there is a $\mathbf{b}' \in \mathrm{Team}(\mathfrak{B})$ with $(\mathbf{a}', \mathbf{b}') \in Z$ and $\mathfrak{B} \models \vartheta_X(\mathbf{b}', \mathbf{b})$.

We write $\mathfrak{A}, \mathbf{a} \sim \mathfrak{B}, \mathbf{b}$ if there is a bisimulation $Z$ between $\mathfrak{A}$ and $\mathfrak{B}$ with $(\mathbf{a}, \mathbf{b}) \in Z$.

**Definition 4.32.** We write $\mathfrak{A}, \mathbf{a} \sim^0 \mathfrak{B}, \mathbf{b}$ if $\mathfrak{A}, \mathbf{a} \equiv^0_{\mathsf{LFD}} \mathfrak{B}, \mathbf{b}$. Now let $k \in \mathbb{N}$. For $\mathfrak{A}, \mathbf{a} \sim^{k+1} \mathfrak{B}, \mathbf{b}$ we require

- $(k+1)$-back: For all $\mathbf{b}' \in \mathrm{Team}(\mathfrak{B})$ and (finite) $X \subseteq V$ with $\mathfrak{B} \models \vartheta_X(\mathbf{b}', \mathbf{b})$ there is an $\mathbf{a}' \in \mathrm{Team}(\mathfrak{A})$ with $\mathfrak{A}, \mathbf{a}' \sim^k \mathfrak{B}, \mathbf{b}'$ and $\mathfrak{A} \models \vartheta_X(\mathbf{a}', \mathbf{a})$.
- $(k+1)$-forth: For all $\mathbf{a}' \in \mathrm{Team}(\mathfrak{A})$ and (finite) $X \subseteq V$ with $\mathfrak{A} \models \vartheta_X(\mathbf{a}', \mathbf{a})$ there is a $\mathbf{b}' \in \mathrm{Team}(\mathfrak{B})$ with $\mathfrak{A}, \mathbf{a}' \sim^k \mathfrak{B}, \mathbf{b}'$ and $\mathfrak{B} \models \vartheta_X(\mathbf{b}', \mathbf{b})$.

Finally, we also say that $\mathfrak{A}, \mathbf{a} \sim^\omega \mathfrak{B}, \mathbf{b}$ iff $\mathfrak{A}, \mathbf{a} \sim^k \mathfrak{B}, \mathbf{b}$ for all $k \in \mathbb{N}$.

Since we only deal with LFD in the current setting, we will use $\sim$ instead of $\sim_{\mathsf{LFD}}$ for dependence models and LFD-bisimulation will just be called bisimulation. The main reason for the requirement that $\mathsf{G}$ induces a surjection $\mathsf{G}\colon \mathrm{Team}(\mathfrak{A}) \twoheadrightarrow T_{\mathsf{G}(\mathfrak{A})}$ for every fixed $\mathfrak{A}, \mathbf{a} \in \mathcal{C}[\sigma]$, is that we thereby obtain the following correspondence, which one may also view as a justification for the definition of translated bisimulations.

**Proposition 4.33.** For $k \in \mathbb{N}$

$$\mathfrak{A}, \mathbf{a} \sim^k \mathfrak{B}, \mathbf{b} \qquad \text{iff} \qquad \mathsf{G}(\mathfrak{A}, \mathbf{a}) \sim^k \mathsf{G}(\mathfrak{B}, \mathbf{b}),$$

as well as $\mathfrak{A}, \mathbf{a} \sim \mathfrak{B}, \mathbf{b}$ iff $\mathsf{G}(\mathfrak{A}, \mathbf{a}) \sim \mathsf{G}(\mathfrak{B}, \mathbf{b})$.

*Proof.* See Proposition C.4 in the appendix. □

**Remark 4.34.** Considering Definition 4.29, our Ehrenfeucht-Fraïssé analogue Theorem 3.12 carries over for finite types: for $k \in \mathbb{N}$

$$\begin{aligned} \mathfrak{A}, \mathbf{a} \sim^k \mathfrak{B}, \mathbf{b} \qquad &\text{iff} \qquad \mathsf{G}(\mathfrak{A}, \mathbf{a}) \sim^k \mathsf{G}(\mathfrak{B}, \mathbf{b}) \\ &\text{iff} \qquad \mathsf{G}(\mathfrak{A}, \mathbf{a}) \equiv^k_{\mathsf{LFD}} \mathsf{G}(\mathfrak{B}, \mathbf{b}) \\ &\text{iff} \qquad \mathfrak{A}, \mathbf{a} \equiv^k_{\mathsf{LFD}} \mathfrak{B}, \mathbf{b}. \end{aligned}$$

In particular, $\mathfrak{A}, \mathbf{a} \sim^\omega \mathfrak{B}, \mathbf{b}$ iff $\mathfrak{A}, \mathbf{a} \equiv_{\mathsf{LFD}} \mathfrak{B}, \mathbf{b}$.

**Remark 4.35.** Using the above correspondence, we have for $k \in \mathbb{N}$

$$\begin{aligned} \mathbf{M}, s \sim^k \mathbf{N}, t \qquad &\text{iff} \qquad \mathbf{M}, s \equiv^k_{\mathsf{LFD}} \mathbf{N}, t \\ &\text{iff} \qquad \mathsf{F}(\mathbf{M}, s) \equiv^k_{\mathsf{LFD}} \mathsf{F}(\mathbf{N}, t) \\ &\text{iff} \qquad \mathsf{F}(\mathbf{M}, s) \sim^k \mathsf{F}(\mathbf{N}, t). \end{aligned}$$

A correspondence for full bisimulation analogous to Proposition 4.33 seems to require that $\mathsf{F}$, analogously to $\mathsf{G}$, induces a surjection $\mathsf{F} : T_{\mathbf{M}} \twoheadrightarrow \mathrm{Team}(\mathsf{F}(\mathbf{M}))$ for every fixed $\mathbf{M}, s \in \mathcal{DEP}[\tau, V]$. We did not assume this in the definition of our good translations, because we do not need this correspondence for the proof of our main theorems. Note though that our three discussed example translations fulfill this requirement.

**Remark 4.36.** We can adapt the characteristic formulae from Lemma 3.10 by defining that for any $k \in \mathbb{N}$

$$\chi^k_{\mathfrak{A},\mathbf{a}} \coloneqq \mathsf{tr}(\chi^k_{\mathsf{G}(\mathfrak{A},\mathbf{a})}), \qquad \mathfrak{A}, \mathbf{a} \in \mathcal{C}[\sigma].$$

By Proposition 4.33 and Lemma 3.10 we obtain that for $(\mathfrak{A}, \mathbf{a}), (\mathfrak{B}, \mathbf{b}) \in \mathcal{C}[\sigma]$

$$\mathfrak{B}, \mathbf{b} \models \chi^k_{\mathfrak{A},\mathbf{a}} \qquad \text{iff} \qquad \mathfrak{B}, \mathbf{b} \sim^k \mathfrak{A}, \mathbf{a}.$$

### 4.1.5 A Characterisation Theorem

In this subsection we will prove a characterisation theorem for good first-order translations of LFD with respect to their translated bisimulations. The theorem is an analogue of van Benthem's Theorem, which states that ML, via its standard translation into FO, is precisely the bisimulation-invariant fragment of FO over the class of pointed Kripke structures. It was first formulated in [7] and [8]. We adapt a well known proof using saturated structures by mainly following the exposition in [11, Chapter 2.6]. First, we briefly recall some standard model-theoretic notions.

**Definition 4.37.** Let $\sigma$ be a signature and $\mathfrak{A}$ a $\sigma$-structure with universe $A$.

1. The theory $\mathrm{Th}(\mathfrak{A})$ of $\mathfrak{A}$ is the set of all $\mathsf{FO}(\sigma)$ sentences satisfied in $\mathfrak{A}$.

2. A substructure $\mathfrak{B} \subseteq \mathfrak{A}$ is called an elementary substructure of $\mathfrak{A}$, or equivalently $\mathfrak{A}$ an elementary extension of $\mathfrak{B}$, denoted $\mathfrak{B} \preceq \mathfrak{A}$, if for all $\varphi(\mathbf{x}) \in \mathsf{FO}(\sigma)$

$$\mathfrak{B} \models \varphi(\mathbf{b}) \quad \text{iff} \quad \mathfrak{A} \models \varphi(\mathbf{b}), \qquad \text{for all fitting tuples } \mathbf{b} \text{ in } \mathfrak{B}.$$

3. For a set $B \subseteq A$ we let $\mathfrak{A}_B$ be the expansion of $\mathfrak{A}$ by constants $c_b$ for each $b \in B$ which are interpreted as $c_b^{\mathfrak{A}} := b$. We also write $\tau \cup B$ for this signature.

4. An $n$-type of $\mathfrak{A}$ over $B \subseteq A$ is a set $p(\mathbf{x})$ of formulae $\varphi(x_1, \ldots, x_n) \in \mathsf{FO}(\tau \cup B)$ such that $p \cup \mathrm{Th}(\mathfrak{A}_B)$ is satisfiable. We call $B$ the set of parameters for $p$.

5. We say $\mathfrak{A}$ is $\omega$-saturated if every type $p(\mathbf{x})$ of $\mathfrak{A}$ with *finitely many parameters* is realized in $\mathfrak{A}$, i.e. there is some tuple $\mathbf{a}$ in $\mathfrak{A}$ with $\mathfrak{A} \models p(\mathbf{a})$.

**Theorem 4.38.** Every structure has an $\omega$-saturated elementary extension.

*Proof.* See [13, Chapter 5] or [25, Chapter 8]. □

**Lemma 4.39.** Let $(\tau, V)$ be a finite type and $(\mathcal{C}[\sigma], \mathsf{F}, \mathsf{G}, \mathsf{tr}, (\vartheta_X)_{X \subseteq V})$ a good translation from LFD over $\mathcal{DEP}[\tau, V]$ to FO over $\mathcal{C}[\sigma]$. Assume $\mathcal{C}[\sigma]$ contains two (pointed) $\omega$-saturated models $\mathfrak{A}, \mathbf{a}$ and $\mathfrak{B}, \mathbf{b}$. If $\mathfrak{A}, \mathbf{a} \sim^\omega \mathfrak{B}, \mathbf{b}$, then already $\mathfrak{A}, \mathbf{a} \sim \mathfrak{B}, \mathbf{b}$.

*Proof.* Let $\mathfrak{A}, \mathbf{a}$ and $\mathfrak{B}, \mathbf{b}$ be as described above, and assume that $\mathfrak{A}, \mathbf{a} \sim^\omega \mathfrak{B}, \mathbf{b}$. Define

$$Z := \{(\mathbf{s}, \mathbf{t}) \in \mathrm{Team}(\mathfrak{A}) \times \mathrm{Team}(\mathfrak{B}) \mid \mathfrak{A}, \mathbf{s} \equiv_{\mathsf{LFD}} \mathfrak{B}, \mathbf{t}\}.$$

By Remark 4.34 we have $(\mathbf{a},\mathbf{b}) \in Z$. We now show that $Z$ is a bisimulation according to Definition 4.31, so let $(\mathbf{s},\mathbf{t}) \in Z$. Clearly $\mathfrak{A},\mathbf{s} \equiv^0_{\mathsf{LFD}} \mathfrak{B},\mathbf{t}$. We proceed by checking the forth condition.

Let $\mathbf{s}' \in \mathrm{Team}(\mathfrak{A})$ and $X \subseteq V$ with $\mathfrak{A} \models \vartheta_X(\mathbf{s}',\mathbf{s})$. Set

$$p(\mathbf{x}) := \{\vartheta_X(\mathbf{x},\mathbf{t})\} \cup \mathrm{Th}_{\mathsf{LFD}}(\mathfrak{A},\mathbf{s}').$$

We want to show that $p$ is a type of $\mathfrak{B}$ over $[\mathbf{t}]$, i.e. that $p$ together with the first-order theory of $\mathfrak{B}_{[\mathbf{t}]}$ is satisfiable. Assume this is not the case, so by compactness there exists a finite $\Phi_0 \subseteq \mathrm{Th}_{\mathsf{LFD}}(\mathfrak{A},\mathbf{s}')$ with $\mathfrak{B},\mathbf{t} \models \varphi$, where

$$\varphi(\mathbf{x}) := \forall \mathbf{t}'(\vartheta_X(\mathbf{t}',\mathbf{x}) \to \neg \bigwedge \Phi_0(\mathbf{t}')).$$

Per definition of $\mathrm{Th}_{\mathsf{LFD}}$, we know that there is some finite $\Psi_0 \subseteq \mathsf{LFD}$ such that $\Phi_0 = \{\mathsf{tr}(\psi) \mid \psi \in \Psi_0\}$. Since $\mathsf{tr}$ commutes with boolean connectives, it follows that $\neg \bigwedge \Phi_0 = \mathsf{tr}\,(\neg \bigwedge \Psi_0)$. Now $\varphi$ is equivalent (over $\mathcal{C}[\sigma]$) to the translation of the $\mathsf{LFD}$-formula $\mathsf{D}_X \neg \bigwedge \Psi_0$. Indeed, for an arbitrary $\mathfrak{C},\mathbf{c} \in \mathcal{C}[\sigma]$, we know that $\mathsf{G}$ induces a surjection of $\mathrm{Team}(\mathfrak{C})$ onto $T_{\mathsf{G}(\mathfrak{C})}$, so we get

$$\begin{aligned}
&\mathfrak{C},\mathbf{c} \models \varphi\\
\text{iff}\quad &\text{for all } \mathbf{t}' \in \mathrm{Team}(\mathfrak{C}) \text{ with } \mathfrak{C} \models \vartheta_X(\mathbf{t}',\mathbf{c}) \text{ we have } \mathfrak{C},\mathbf{t}' \models \neg \textstyle\bigwedge \Phi_0\\
\text{iff}\quad &\text{for all } \mathbf{t}' \in \mathrm{Team}(\mathfrak{C}) \text{ with } \mathsf{G}(\mathbf{t}') =_X \mathsf{G}(\mathbf{c}) \text{ we have } \mathsf{G}(\mathfrak{C},\mathbf{t}') \models \neg \textstyle\bigwedge \Psi_0\\
\text{iff}\quad &\text{for all } t' \in T_{\mathsf{G}(\mathfrak{C})} \text{ with } t' =_X \mathsf{G}(\mathbf{c}) \text{ we have } \mathsf{G}(\mathfrak{C}),t' \models \neg \textstyle\bigwedge \Psi_0\\
\text{iff}\quad &\mathsf{G}(\mathfrak{C},\mathbf{c}) \models \mathsf{D}_X \neg \textstyle\bigwedge \Psi_0\\
\text{iff}\quad &\mathfrak{C},\mathbf{c} \models \mathsf{tr}\left(\mathsf{D}_X \neg \textstyle\bigwedge \Psi_0\right).
\end{aligned}$$

Since $\mathfrak{A},\mathbf{s} \equiv_{\mathsf{LFD}} \mathfrak{B},\mathbf{t}$ and $\mathfrak{B},\mathbf{t} \models \varphi$, we therefore obtain $\mathfrak{A},\mathbf{s} \models \varphi$. In particular $\mathfrak{A},\mathbf{s}' \models \neg \bigwedge \Phi_0$, which contradicts $\Phi_0 \subseteq \mathrm{Th}_{\mathsf{LFD}}(\mathfrak{A},\mathbf{s}')$.

Thus $p$ is a type with finitely many parameters (namely $[\mathbf{t}]$) over $\mathfrak{B}$. By $\omega$-saturatedness we obtain some $\mathbf{t}'$ in $\mathfrak{B}$ with $\mathfrak{B} \models p(\mathbf{t}')$. It follows that $\mathfrak{B} \models \vartheta_X(\mathbf{t}',\mathbf{t})$ and $(\mathbf{s}',\mathbf{t}') \in Z$, which proves the forth condition. The back condition is shown analogously. We conclude that $Z$ is a bisimulation and hence $\mathfrak{A},\mathbf{a} \sim \mathfrak{B},\mathbf{b}$. □

**Theorem 4.40** (Expressive Completeness)**.** Let $(\tau, V)$ be a finite type and consider a good translation $(\mathcal{C}[\sigma], \mathsf{F}, \mathsf{G}, \mathsf{tr}, (\vartheta_X)_{X\subseteq V})$ of $\mathsf{LFD}$ over $\mathcal{DEP}[\tau, V]$ to $\mathsf{FO}$ over $\mathcal{C}[\sigma]$. For any $\varphi \in \mathsf{FO}(\sigma)$ the following are equivalent

1. $\varphi$ is bisimulation-invariant over $\mathcal{C}[\sigma]$, i.e. for all $(\mathfrak{A},\mathbf{a}),(\mathfrak{B},\mathbf{b}) \in \mathcal{C}[\sigma]$ we have

$$\mathfrak{A},\mathbf{a} \sim \mathfrak{B},\mathbf{b} \qquad \Longrightarrow \qquad \mathfrak{A},\mathbf{a} \models \varphi \quad \text{iff} \quad \mathfrak{B},\mathbf{b} \models \varphi.$$

2. $\varphi$ is equivalent to an $\mathsf{LFD}$-formula over $\mathcal{C}[\sigma]$, so there is a $\psi \in \mathsf{LFD}$ with

$$\mathfrak{A}, \mathbf{a} \models \varphi \qquad \text{iff} \qquad \mathfrak{A}, \mathbf{a} \models \mathsf{tr}(\psi), \qquad\qquad \mathfrak{A}, \mathbf{a} \in \mathcal{C}[\sigma].$$

Hence we write

$$\mathsf{FO}/{\sim} \equiv \mathsf{LFD} \text{ over } \mathcal{C}[\sigma] \text{ via } \mathsf{tr}.$$

*Proof.* We know from Remark 4.34 that $\mathfrak{A}, \mathbf{a} \sim \mathfrak{B}, \mathbf{b}$ entails $\mathfrak{A}, \mathbf{a} \equiv_{\mathsf{LFD}} \mathfrak{B}, \mathbf{b}$. Hence the implication "(2) $\Rightarrow$ (1)" is clear.

For the converse, let $\Theta \subseteq \mathsf{FO}(\sigma)$ be the set of formulae that defines $\mathcal{C}[\sigma]$, as required in Definition 4.23. Recall the notation $\equiv_{\mathcal{C}}$ and $\models_{\mathcal{C}}$ from Notation 4.30. Note that $\Psi \models_{\mathcal{C}} \psi$ if and only if $\Theta \cup \Psi \models \psi$.

Now assume (1), that $\varphi$ is bisimulation-invariant over $\mathcal{C}[\sigma]$, and consider the set of $\mathsf{LFD}$-consequences of $\varphi$ over $\mathcal{C}$:

$$C(\varphi) := \{\mathsf{tr}(\psi) \mid \psi \in \mathsf{LFD}(\tau, V) \text{ and } \varphi \models_{\mathcal{C}} \mathsf{tr}(\psi)\}.$$

We claim that it suffices to show $C(\varphi) \models_{\mathcal{C}} \varphi$. Indeed, using compactness this yields a finite subset $C_0 \subseteq C(\varphi)$ with $C_0 \models_{\mathcal{C}} \varphi$ and therefore $\bigwedge C_0 \equiv_{\mathcal{C}} \varphi$. But then we can write $C_0 = \{\mathsf{tr}(\psi_1), \dots, \mathsf{tr}(\psi_m)\}$ for $\psi_k \in \mathsf{LFD}$ and come to the conclusion

$$\varphi \equiv_{\mathcal{C}} \bigwedge C_0 = \mathsf{tr}\left(\bigwedge_{k=1}^{m} \psi_k\right)$$

which shows that $\varphi$ is equivalent to an $\mathsf{LFD}$-formula over $\mathcal{C}[\sigma]$, so we are done.

Thus we only need to prove $C(\varphi) \models_{\mathcal{C}} \varphi$. If $C(\varphi)$ is unsatisfiable over $\mathcal{C}[\sigma]$, this holds vacuously. Hence let $\mathfrak{A}, \mathbf{a} \in \mathcal{C}[\sigma]$ with $\mathfrak{A}, \mathbf{a} \models C(\varphi)$, so we need to show $\mathfrak{A}, \mathbf{a} \models \varphi$. We claim that $\mathrm{Th}_{\mathsf{LFD}}(\mathfrak{A}, \mathbf{a}) \cup \{\varphi\}$ is satisfiable over $\mathcal{C}[\sigma]$. Otherwise, compactness gives us some finite $\Phi_0 \subseteq \mathrm{Th}_{\mathsf{LFD}}(\mathfrak{A}, \mathbf{a})$ such that $\Theta \cup \Phi_0 \cup \{\varphi\}$ is unsatisfiable, meaning

$$\Theta \cup \{\varphi\} \models \neg \bigwedge \Phi_0 \qquad \text{and thus} \qquad \varphi \models_{\mathcal{C}} \neg \bigwedge \Phi_0.$$

But then $\neg \bigwedge \Phi_0$ is an $\mathsf{LFD}$-consequence of $\varphi$ and hence contained in $C(\varphi)$. This contradicts $\mathfrak{A}, \mathbf{a} \models C(\varphi)$ and $\Phi_0 \subseteq \mathrm{Th}_{\mathsf{LFD}}(\mathfrak{A}, \mathbf{a})$.

Therefore $\mathrm{Th}_{\mathsf{LFD}}(\mathfrak{A}, \mathbf{a}) \cup \{\varphi\}$ has some model $\mathfrak{B}, \mathbf{b} \in \mathcal{C}[\sigma]$. Note that $\mathfrak{B}, \mathbf{b} \models \mathrm{Th}_{\mathsf{LFD}}(\mathfrak{A}, \mathbf{a})$ implies $\mathfrak{A}, \mathbf{a} \equiv_{\mathsf{LFD}} \mathfrak{B}, \mathbf{b}$. Now take $\omega$-saturated elementary extensions $\mathfrak{A} \preceq \mathfrak{A}^+$ and $\mathfrak{B} \preceq \mathfrak{B}^+$. By elementary extension we have $\mathfrak{A}^+, \mathbf{a} \models \Theta$, so $\mathfrak{A}^+, \mathbf{a} \in \mathcal{C}[\sigma]$, as well as $\mathfrak{A}^+, \mathbf{a} \equiv_{\mathsf{LFD}} \mathfrak{A}, \mathbf{a}$; likewise for $\mathfrak{B}^+, \mathbf{b}$. It follows that

$$\mathfrak{A}^+, \mathbf{a} \equiv_{\mathsf{LFD}} \mathfrak{A}, \mathbf{a} \equiv_{\mathsf{LFD}} \mathfrak{B}, \mathbf{b} \equiv_{\mathsf{LFD}} \mathfrak{B}^+, \mathbf{b}.$$

The Ehrenfeucht-Fraïssé correspondence from Remark 4.34 yields $\mathfrak{A}^+, \mathbf{a} \sim^{\omega} \mathfrak{B}^+, \mathbf{b}$.

But now we can apply Lemma 4.39 to find $\mathfrak{A}^+, \mathbf{a} \sim \mathfrak{B}^+, \mathbf{b}$. The following diagram depicts the current situation.

$$\begin{array}{ccc} \mathfrak{A}, \mathbf{a} & & \mathfrak{B}, \mathbf{b} \\ \curlyeqprec & & \curlyeqprec \\ \mathfrak{A}^+, \mathbf{a} & \sim & \mathfrak{B}^+, \mathbf{b} \end{array}$$

We know $\varphi$ is bisimulation-invariant over $\mathcal{C}[\sigma]$. Moreover, as $\varphi \in \mathsf{FO}$, it is also invariant under passage to elementary extensions and substructures. Based on $\mathfrak{B}, \mathbf{b} \models \varphi$ we can then conclude $\mathfrak{A}, \mathbf{a} \models \varphi$ via diagram chasing. With this we inferred $\mathfrak{A}, \mathbf{a} \models \varphi$ from $\mathfrak{A}, \mathbf{a} \models C(\varphi)$ for an arbitrary $\mathfrak{A}, \mathbf{a} \in \mathcal{C}[\sigma]$, which shows that $C(\varphi) \models_{\mathcal{C}} \varphi$. We discussed above how this concludes the proof of the theorem. ☐

**Remark 4.41.** In the setting of this theorem, if $\varphi$ is bisimulation-invariant over $\mathcal{C}[\sigma]$, it is already invariant under $k$-bisimulation over $\mathcal{C}[\sigma]$ for some $k \in \mathbb{N}$. Indeed, just take $k$ to be the quantifier rank of an $\mathsf{LFD}$-formula to which $\varphi$ is equivalent over $\mathcal{C}[\sigma]$. As a consequence, we obtain an analogue of Corollary 3.13, namely that

$$\varphi \equiv_{\mathcal{C}} \bigvee_{\mathfrak{A},\mathbf{a} \models \varphi} \chi^k_{\mathfrak{A},\mathbf{a}} = \mathsf{tr}\left( \bigvee_{\mathfrak{A},\mathbf{a} \models \varphi} \chi^k_{\mathsf{G}(\mathfrak{A},\mathbf{a})} \right),$$

where the disjunctions range over all $\mathfrak{A}, \mathbf{a} \in \mathcal{C}[\sigma]$, but are essentially finite, since up to $\mathsf{LFD}$-equivalence there are only finitely many such characteristic formulae, see Lemma 3.10 and Remark 4.36.

The three translations we discussed in Sections 4.1.1 to 4.1.3 work the same for all finite types $(\tau, V)$ and corresponding $\sigma$. In this case, we write

$$\mathsf{FO}/{\sim} \equiv \mathsf{LFD} \text{ over } \mathcal{C} \text{ via } \mathsf{tr}.$$

Note though that for different good translations the adapted notions of bisimulation as defined in Definition 4.31 are also formally different. Although we use the same symbol $\sim$ for all of them, it should always be clear from the context which translation and corresponding notion of bisimulation we are referring to. In particular, the relations $\sim$ in the three points of the following corollary are not the same relations, and depend on which translation $\mathsf{tr}$ and class of structures $\mathcal{C}$ we consider.

**Corollary 4.42.** Applying Theorem 4.40 to the three translations we discussed in Sections 4.1.1 to 4.1.3 (or Examples 4.25 to 4.27 for short), we obtain

1. $\mathsf{FO}/{\sim} \equiv \mathsf{LFD}$ over $\mathcal{DFO}$ via the standard translation $\mathsf{tr}_{\mathrm{st}}$.

2. $\mathsf{FO}/{\sim} \equiv \mathsf{FO}^2/{\sim} \equiv \mathsf{LFD}$ over $\mathcal{EQD}$ via the modal translation $\mathsf{tr}_{\mathrm{mod}}$, since it embeds $\mathsf{LFD}$ into $\mathsf{FO}^2$.

3. $\mathsf{FO}/{\sim} \equiv \mathsf{FO}^2/{\sim} \equiv \mathsf{LFD}$ over $\mathcal{FUN}$ via the functional translation $\mathsf{tr}_{\mathrm{fun}}$, since it also embeds $\mathsf{LFD}$ into $\mathsf{FO}^2$.

**Remark 4.43.** Remember that the standard translation can easily be extended to $\mathsf{LFD}^=$. In this case, one can create an analogue of Section 4.1.4 for $\mathsf{LFD}^=$. Then, the theorems of this section hold with minimally adapted proofs, since the equality atoms are simply hidden in the "agreement on atoms" condition for bisimulation. Denote the adapted $\mathsf{LFD}^=$-bisimulation by $\sim_=$. Then $\mathsf{FO}/{\sim_=} \equiv \mathsf{LFD}^=$ over $\mathcal{DFO}$ via the adapted standard translation $\mathsf{tr}_{\mathrm{st}}$, see Remark 4.7.

## 4.2 Guarded Fragments

Guarded logics arise as a natural generalization of modal logics. Considering the standard translation $ST$ of ordinary modal logic $\mathsf{ML}$ into $\mathsf{FO}$, we see that all quantifiers in the resulting $\mathsf{FO}$-formulae are relativized to some accessibility relation $E$:

$$ST_x(\Box\varphi) = \forall y(Exy \to ST_y(\varphi)) \qquad \text{and} \qquad ST_x(\Diamond\varphi) = \exists y(Exy \land ST_y(\varphi)).$$

The guarded fragment, denoted $\mathsf{GF}$, is a fragment of relational $\mathsf{FO}$ in which all quantifiers must be relativized (guarded) by some positive atom in this sense. More formally, $\mathsf{GF}$ is the smallest fragment of relational $\mathsf{FO}$ generated from atomic formulae by boolean connectives and guarded quantification, which is defined as follows: if $\alpha(\mathbf{xy})$ is a positive atomic formula and $\varphi(\mathbf{xy})$ a formula in $\mathsf{GF}$ with $\mathrm{Free}(\psi) \subseteq \mathrm{Free}(\alpha) = [\mathbf{x}] \cup [\mathbf{y}]$, then

$$\forall \mathbf{y}(\alpha(\mathbf{xy}) \to \varphi(\mathbf{xy})) \in \mathsf{GF} \qquad \text{and} \qquad \exists \mathbf{y}(\alpha(\mathbf{xy}) \land \varphi(\mathbf{xy})) \in \mathsf{GF}.$$

$\mathsf{GF}$ was first introduced in [2], and today many different generalizations have been considered. For example, weakening the conditions of guarded quantifications leads to the loosely guarded fragment $\mathsf{LGF}$, introduced in [9], where we can guard quantification by conjunctions of atomic formulae satisfying a certain condition, instead of single atomic formula. For example, the formula $\forall z((x \leq z \land z < y) \to \psi(z))$ is often found in the translations of formulae of the form $(\psi \,\mathrm{until}\, \varphi)$ from temporal logics such as $\mathsf{LTL}$. This quantification is not guarded, but it is loosely guarded (cf. [23]). On a different note, one can extend the notion of guardedness to fixed point logics, see e.g. [22].

It was found that many of such guarded logics share much of the nice decidability and model-theoretic properties of modal logics, see e.g. [2, 17, 23, 10, 20], and it is now believed that this guardedness is one of the reasons that modal logics are so well-behaved [18].

Arguably one of the most natural generalizations of GF within FO was introduced in [23] as the clique-guarded fragment CGF. It was observed that (loosely) guarded tuples of some relational structure $\mathfrak{A}$ induce a clique in the Gaifman graph[4] of $\mathfrak{A}$. Moreover, for each finite relational $\tau$ and $k \in \mathbb{N}$, there is a positive, existential first-order formula $\mathit{clique}(x_1, \ldots, x_k)$ which is satisfied at a tuple $\mathbf{a}$ of some $\tau$-structure $\mathfrak{A}$ if and only if $\mathbf{a}$ induces a clique in the Gaifman graph of $\mathfrak{A}$. CGF is then defined in an analogous way to GF or LGF, but always uses *clique* of the right arity as a guard, in the sense of

$$\forall \mathbf{y}(\mathit{clique}(\mathbf{xy}) \rightarrow \varphi(\mathbf{xy})) \qquad \text{and} \qquad \exists \mathbf{y}(\mathit{clique}(\mathbf{xy}) \wedge \varphi(\mathbf{xy})).$$

Overall the following hierarchy is well known (see e.g. [2, 23]):

$$\mathsf{ML} \subsetneq \mathsf{GF} \subsetneq \mathsf{LGF} \subsetneq \mathsf{CGF} \subsetneq \mathsf{FO}.$$

Guarded bisimulations between structures $\mathfrak{A}, \mathbf{a}$ and $\mathfrak{B}, \mathbf{b}$ are defined as sets $I$ of partial isomorphisms between $\mathfrak{A}$ and $\mathfrak{B}$ such that $(\mathbf{a} \mapsto \mathbf{b}) \in I$ and $I$ is closed under suitable back and forth conditions. This notion naturally extends to clique-guarded bisimulation for CGF, as described in [23]. For a survey of various notions of bisimulation and their uses for understanding expressive power, model-theoretic and algorithmic properties of modal and guarded logics, we refer the reader to [20].

**Fact 4.44.** The relevant bisimulations for GF, LGF, CGF are all compatible with disjoint unions of bisimilar models. More specifically, assume we have clique-guarded bisimulations $I$ between $\mathfrak{A}$ and $\mathfrak{B}$ as well as $I'$ between $\mathfrak{A}$ and $\mathfrak{C}$. Then we obtain that $I \uplus I'$ is a clique-guarded bisimulation between $\mathfrak{A}$ and $\mathfrak{B} \uplus \mathfrak{C}$. In particular, from $\mathfrak{A}, \mathbf{a} \sim_{\mathsf{CGF}} \mathfrak{B}, \mathbf{b}$ and $\mathfrak{A}, \mathbf{a} \sim_{\mathsf{CGF}} \mathfrak{C}, \mathbf{c}$ we infer

$$\mathfrak{A}, \mathbf{a} \sim_{\mathsf{CGF}} (\mathfrak{B} \uplus \mathfrak{C}), \mathbf{b} \qquad \text{and} \qquad \mathfrak{A}, \mathbf{a} \sim_{\mathsf{CGF}} (\mathfrak{B} \uplus \mathfrak{C}), \mathbf{c}.$$

This also holds for GF or LGF, since clique-guarded bisimilarity is finer than guarded and loosely-guarded bisimilarity, see also [2, 23].

This highlights a small but important difference between LFD and the above guarded fragments. Namely, as a consequence of the above fact and the invariance of these guarded fragments under their respective bisimulation, they are *invariant under duplications* in the following sense.

[4] The Gaifman graph of a relational $\tau$-structure $\mathfrak{A}$ has as its universe the universe of $\mathfrak{A}$, and an edge between two unequal elements if they coexist in some relational fact of $\mathfrak{A}$, i.e. they occur together in some $\mathbf{c} \in R^{\mathfrak{A}}$ for some $R \in \tau$.

**Definition 4.45.** Let $\sigma$ be a vocabulary and $\mathcal{C}[\sigma]$ a class of pointed $\sigma$-structures. Denote disjoint copies of $\mathfrak{A}, \mathbf{a} \in \mathcal{C}[\sigma]$ by $\mathfrak{A}', \mathbf{a}'$.

1. We say $\mathcal{C}[\sigma]$ is *closed under duplications* if for all such $\mathfrak{A}, \mathbf{a} \in \mathcal{C}[\sigma]$ we have $(\mathfrak{A} \uplus \mathfrak{A}'), \mathbf{a} \in \mathcal{C}[\sigma]$ and $(\mathfrak{A} \uplus \mathfrak{A}'), \mathbf{a}' \in \mathcal{C}[\sigma]$.

2. Furthermore, we call a fragment $\mathcal{L}(\sigma) \subseteq \mathsf{FO}(\sigma)$ *invariant under duplications* if for all $\mathfrak{A}, \mathbf{a} \in \mathcal{C}[\sigma]$ and $\psi(\mathbf{x}) \in \mathcal{L}(\sigma)$ we have

$$\mathfrak{A} \models \psi(\mathbf{a}) \qquad \text{iff} \qquad (\mathfrak{A} \uplus \mathfrak{A}') \models \psi(\mathbf{a}) \wedge \psi(\mathbf{a}').$$

Examples of duplication-closed classes are the class of all pointed $\sigma$-structures, or classes defined by some fragment of FO which is invariant under duplications. Fact 4.44 implies that GF, LGF, CGF are examples of such duplication-invariant fragments of first-order logic.

However, we discussed in Section 3.2 that an analogue for invariance under duplication does not hold for LFD. This is caused by dependence atoms of the form $D_\varnothing x$, allowing LFD to state that some variable is constant, which is clearly not invariant under duplications. Conversely, we saw in Proposition 3.17 that LFD restricted to (distinguished) dependence models without constant variables is duplication-invariant, which shows that those constancy atoms are really the only reason that LFD is not duplication-invariant. Nevertheless, this difference between LFD and the guarded fragments comes to light for good first-order translations as follows.

Let $(\tau, V)$ be a finite type and $(\mathcal{C}[\sigma], \mathsf{F}, \mathsf{G}, \mathsf{tr}, (\vartheta_X)_{X \subseteq V})$ a good translation of LFD over $\mathcal{DEP}[\tau, V]$ into some $\mathcal{L} \subseteq \mathsf{FO}$ over $\mathcal{C}[\sigma]$. Note that for $X \neq \varnothing$, the $\vartheta_X$ can be defined as $\bigwedge_{x \in X} \vartheta_x$, so the structure of the $\vartheta_x$ essentially decides the structure of $\vartheta_X$ for $X \neq \varnothing$. Regarding these formulae in our example translations, we observe that the $\vartheta_x(\mathbf{x}, \mathbf{y})$ are just conjunctions of positive atoms, where at least one of the atoms contains variables from both $\mathbf{x}$ and $\mathbf{y}$ (cf. Examples 4.25 to 4.27). For our case, the crucial consequence of this is that they require some connection between $\mathbf{x}$ and $\mathbf{y}$ in the form of said atom. More formally, they are *local* in the following sense.

**Definition 4.46.** Assume we have some $\mathfrak{A}, \mathbf{a} \in \mathcal{C}[\sigma]$ and another tuple $\mathbf{b} \in \mathrm{Team}(\mathfrak{A})$ disjoint from $\mathbf{a}$ such that no elements of $\mathbf{a}$ and $\mathbf{b}$ occur together in any relational fact of $\mathfrak{A}$. This means that for all $R \in \sigma$ and $\mathbf{c} \in R^{\mathfrak{A}}$ we have $[\mathbf{c}] \cap [\mathbf{a}] = \varnothing$ or $[\mathbf{c}] \cap [\mathbf{b}] = \varnothing$. If it follows for any such triple $(\mathfrak{A}, \mathbf{a}, \mathbf{b})$ that $\mathfrak{A} \not\models \vartheta_x(\mathbf{a}, \mathbf{b})$, then we say that $\vartheta_x$ is local.

We now show that under reasonable assumptions towards good first-order translations of LFD, we can use the reason that LFD is not invariant under duplications (namely that it can define constants) to show that such translations cannot embed LFD into duplication-invariant fragments of FO.

**Proposition 4.47.** Let $(\mathcal{C}[\sigma], \mathsf{F}, \mathsf{G}, \mathsf{tr}, (\vartheta_X)_{X\subseteq V})$ be a good translation of LFD over $\mathcal{DEP}[\tau, V]$ to some $\mathcal{L}(\sigma) \subseteq \mathsf{FO}(\sigma)$ over $\mathcal{C}[\sigma]$ via $\mathsf{tr}$, such that

1. $\mathcal{C}[\sigma]$ is closed under duplications, and
2. there is an $x \in V$ so that $\vartheta_x$ is local.

Then $\mathcal{L}(\sigma)$ cannot be invariant under duplications, because $\mathsf{tr}(D_\varnothing x) \in \mathcal{L}(\sigma)$ is not.

*Proof.* First, observe that for all $\mathfrak{A}, \mathbf{a} \in \mathcal{C}[\sigma]$ we have

$$\begin{aligned}\mathfrak{A}, \mathbf{a} \models \mathsf{tr}(D_\varnothing x) \quad &\text{iff} \quad \mathfrak{A} \models \forall \mathbf{z}(\vartheta_\varnothing(\mathbf{z}, \mathbf{a}) \to \vartheta_x(\mathbf{z}, \mathbf{a}))\\ &\text{iff} \quad \mathfrak{A} \models \forall \mathbf{z} \forall \mathbf{z}'(\vartheta_\varnothing(\mathbf{z}, \mathbf{z}') \to \vartheta_x(\mathbf{z}, \mathbf{z}')).\end{aligned}$$

Now, assume for the sake of contradiction that $\mathcal{L}(\sigma)$ is duplication-invariant. Consider some $(\tau, V)$ dependence model $\mathbf{M}, s$ with $\mathbf{M}, s \models D_\varnothing x$. Set $\mathfrak{A}, \mathbf{a} := \mathsf{F}(\mathbf{M}, s)$, so that $\mathfrak{A}, \mathbf{a} \models \mathsf{tr}(D_\varnothing x)$. Let $\mathfrak{A}', \mathbf{a}'$ be a disjoint copy of $\mathfrak{A}, \mathbf{a}$ and $\mathfrak{A}^+ := \mathfrak{A} \uplus \mathfrak{A}'$ their disjoint union. Since $\mathcal{C}[\sigma]$ is closed under duplications, we have $(\mathfrak{A}^+, \mathbf{a}), (\mathfrak{A}^+, \mathbf{a}') \in \mathcal{C}[\sigma]$, so in particular $\mathbf{a}, \mathbf{a}' \in \mathrm{Team}(\mathfrak{A}^+)$, which in turn yields $\mathfrak{A}^+ \models \vartheta_\varnothing(\mathbf{a}, \mathbf{a}')$. Clearly $\mathbf{a}$ and $\mathbf{a}'$ are disjoint and none of their elements appear together in any relational fact of $\mathfrak{A}^+$ (since $R^{\mathfrak{A}^+} = R^{\mathfrak{A}} \uplus R^{\mathfrak{A}'}$). Therefore $\mathfrak{A}^+ \models \neg\vartheta_x(\mathbf{a}, \mathbf{a}')$ per locality of $\vartheta_x$. Finally, since $\mathsf{tr}(D_\varnothing x) \in \mathcal{L}(\sigma)$ is invariant under duplication and $\mathfrak{A}, \mathbf{a} \models \mathsf{tr}(D_\varnothing x)$, we must also have $\mathfrak{A}^+, \mathbf{a} \models \mathsf{tr}(D_\varnothing x)$. Thus

$$\mathfrak{A}^+ \models \vartheta_\varnothing(\mathbf{a}, \mathbf{a}') \wedge \neg\vartheta_x(\mathbf{a}, \mathbf{a}') \qquad \text{and} \qquad \mathfrak{A}^+, \mathbf{a} \models \mathsf{tr}(D_\varnothing x),$$

which contradicts the equivalence at the beginning of this proof. □

**Corollary 4.48.** The standard and modal translation do not embed LFD into the clique-guarded fragment CGF.

*Proof.* All $\vartheta_x$ are local for both these translations. Indeed, in the modal case we have that $\vartheta_x(s, t) = s \sim_x t$ is just a binary relation, so clearly local. For the standard translation, we have $\vartheta_x(\mathbf{v}, \mathbf{v}') = [\![T]\!]\mathbf{v} \wedge [\![T]\!]\mathbf{v}' \wedge x = x'$, which contains the equality $x = x'$ that is obviously local, so $\vartheta_x$ is local too. Furthermore, it is easy to see that the classes $\mathcal{DFO}$ and $\mathcal{EQD}$ are duplication-closed. Since we already mentioned that CGF is invariant under duplication, the claim follows from Proposition 4.47. □

There certainly also exist notions expressible in GF but not in LFD. For one, in the setting of the standard translation it becomes obvious that LFD cannot make statements about assignments and values outside of a team; within some $\mathfrak{A}, \mathbf{a} \in \mathcal{DFO}$ a sentence such as $\exists x Rx$ may be true whereas its team-restriction $\mathsf{tr}_{\mathrm{st}}(\mathsf{E}\, Rx) = \exists \mathbf{v}([\![T]\!]\mathbf{v} \wedge Rx)$ is not. We also know from Fact 2.12 that LFD cannot define explicit equality $x = y$, whereas GF clearly can. But both these examples are mostly a consequence of the fact that dependence models, and the class $\mathcal{DFO}$ corresponding to them in the context of the standard translation, offer a lot more information than plain LFD can access. For example, we have seen in the modal context that as long as the $=_X$-classes are known for all $X \subseteq V$, the actual values of assignments are irrelevant for LFD-semantics. Indeed, standard relational models or the structures in $\mathcal{EQD}$ view the assignments as atomic objects, while still offering well-defined LFD-semantics. It is easy to see that both examples from above fail in this context, as the required information to express them in GF (or even FO) is simply not present in structures of $\mathcal{SRM}$ or $\mathcal{EQD}$; it is impossible to define a formula $\varphi \in \mathsf{FO}$ such that $\mathsf{T}_{\mathrm{dep}\,\to\,\mathrm{eqd}}(\mathbf{M}, s) \models \varphi$ iff $s(x) = s(y)$ for all suitable $\mathbf{M}, s$. For this reason, we argue that the standard relational models and the modal perspective in general are the more natural way to view LFD, and that when comparing the expressive power of LFD to fragments of FO, the modal translation (or similar ones) should be used, since it abstracts away the "excess-information" that is not needed for LFD-semantics but still present in dependence models.

But even in the modal setting we can easily find formulae in GF that are not equivalent to any LFD formula. Indeed, the modal character of LFD emerges in the fact that apart from dependence atoms, once we change our viewpoint from some current assignment $s$ to some other assignment $t$ with $t =_X s$ via one of the modalities $\mathsf{D}_X$ or $\mathsf{E}_X$, we cannot refer back to any information about $t$ (except about the values of $X$). As a consequence, the modal translation $\mathsf{tr}_{\mathrm{mod}}$ does not produce arbitrary relational formulae in two variables.[5] In particular, we have the following.

**Proposition 4.49.** Let $(\tau, V)$ be a finite type with $x, y, z \in V$ and define $\sigma$ for $(\tau, V)$ as in the context of the modal translation, see Definition 4.12. Let

$$\varphi(s) := \forall t(t \sim_x s \;\to\; (t \sim_y s \vee t \sim_z s)) \in \mathsf{GF}^2(\sigma).$$

Then there exists no $\psi \in \mathsf{LFD}$ with $\mathsf{tr}_{\mathrm{mod}}(\psi) \equiv_{\mathcal{EQD}} \varphi$.

*Proof.* It is not hard to find an example of two $(\varnothing, \{x, y, z\})$ dependence models that are LFD-bisimilar, but where their corresponding $\mathcal{EQD}$-structures disagree on $\varphi$. The example is then easily extended to arbitrary signatures containing the variables $x, y, z$. For the details, see Proposition A.5 in the appendix. □

[5] This was discussed at the end of Section 4.1.2 when comparing the fragment of FO obtained via $\mathsf{tr}_{\mathrm{mod}}$ to the undecidable fragment of $\mathsf{GF}^2$ with three equivalences defined in Theorem 4.17.

**Corollary 4.50.** In the context of the standard and modal translations, LFD is expressively incomparable to the guarded fragment GF and even the clique-guarded fragment CGF.

*Proof.* One direction of this is already shown in Corollary 4.48. We also mentioned above that in the context of the standard translation, GF (and hence CGF) can define explicit equality whereas LFD cannot. For the modal translation, we have the above proposition. □

It seems likely that one can find analogues of Proposition 4.49 or trivial examples like explicit equality for many more good translations of LFD into relational FO. Together with the generality of Proposition 4.47, this suggests that LFD is intrinsically expressively incomparable to GF and CGF.

## 4.3 Logics with Team Semantics

Team semantics is the basis for many modern logics of dependence, independence and imperfect information (see e.g. [1]). Similarly to LFD, a set of assignments (the team) is relevant for evaluating formulae. But unlike LFD, there is not a designated "current" assignment, and evaluation happens on the whole team, leading to non-classical semantics. Furthermore, the team is not bound to the structure we evaluate a formula on, and may change during evaluation, i.e. subformulae may be evaluated on different teams than the original formula. As mentioned in the introduction, team semantics originated in [26] as a means of providing a compositional, model-theoretic semantics for Independence-Friendly Logic [24].

**Notation 4.51.** As the teams are often extended and split during evaluation, we will use the following standard notations for a team $T \subseteq A^V$:

- If $x$ is some variable (not necessarily in $V$), $s \in T$ and $a \in A$, then $s[x \mapsto a]$ is the unique assignment that agrees with $s$ on $V \setminus \{x\}$ and sends $x$ to $a$. Likewise with tuples $s[\mathbf{x} \mapsto \mathbf{a}]$.
- $T[x \mapsto A] := \{s[x \mapsto a] \mid s \in T, a \in A\}$ and $T[\mathbf{x} \mapsto \mathbf{a}] := \{s[\mathbf{x} \mapsto \mathbf{a}] \mid s \in T\}$.
- Given a function $F\colon T \to \mathcal{P}(A) \setminus \{\varnothing\}$ from the team $T$ into the power set of $A$ excluding the empty set, we set $T[x \mapsto F] := \{s[x \mapsto a] \mid s \in T, a \in F(s)\}$.
- Let $T \restriction X := \{s \restriction X \mid s \in T\}$ be the restriction of $T$ to the domain $X \subseteq V$.

**Definition 4.52.** The (lax) team semantics for $\mathsf{FO}(\tau)$ over $\tau$-structures $\mathfrak{A}$ with universe $A$ is defined recursively via the satisfaction relation $\mathfrak{A} \models_T \psi$ which states that the team $T$ satisfies $\psi$ in $\mathfrak{A}$. We assume that $\psi \in \mathsf{FO}$ is in negation normal form and that $T$ fits to $\mathfrak{A}$ and $\psi$, i.e. that $T \subseteq A^V$ for $V$ containing the free variables of $\psi$.

- If $\psi$ is a relational literal, then $\mathfrak{A} \models_T \psi$ iff $\mathfrak{A}, s \models \psi$ in the usual Tarski semantics, for all $s \in T$.
- $\mathfrak{A} \models_T \psi_1 \wedge \psi_2$ iff $\mathfrak{A} \models_T \psi_1$ and $\mathfrak{A} \models_T \psi_2$.
- $\mathfrak{A} \models_T \psi_1 \vee \psi_2$ iff $T = T_1 \cup T_2$ for subteams $T_i$ with with $\mathfrak{A} \models_{T_i} \psi_i$.
- $\mathfrak{A} \models_T \forall x \psi$ iff $\mathfrak{A} \models_{T[x \mapsto A]} \psi$.
- $\mathfrak{A} \models_T \exists x \psi$ iff $\mathfrak{A} \models_{T[x \mapsto F]} \psi$ for some suitable function $F \colon T \to \mathcal{P}(A) \setminus \{\varnothing\}$.

The idea to treat (functional) dependencies as atomic properties of teams was introduced by Väänänen for Dependence Logic in [34]. One extends first-order logic by atoms $\mathrm{dep}(\mathbf{x}, y)$ with the following semantics:

$$\mathfrak{A} \models_T \mathrm{dep}(\mathbf{x}, y) \qquad \text{iff} \qquad \text{for all } s, t \in T\text{: } s =_{\mathbf{x}} t \text{ implies } s =_y t.$$

One crucial difference to dependence atoms $D_X y$ in $\mathsf{LFD}$ is that $\mathrm{dep}(\mathbf{x}, y)$ represents *global* dependence whereas $D_X y$ is *local*, see also Examples 2.3 and 2.4. Hence, for a structure $\mathfrak{M}$ with universe $M$ and a team $T \subseteq M^V$ we can consider the dependence model $\mathbf{M} = (\mathfrak{M}, T)$ and obtain the equivalence

$$\mathfrak{M} \models_T \mathrm{dep}(\mathbf{x}, y) \qquad \text{iff} \qquad \mathbf{M} \models \mathsf{A}\, D_{\mathbf{x}} y.$$

On the other hand, it is not clear how one would retrieve a notion of local dependence from the atoms of dependence logic. Other atoms commonly added to $\mathsf{FO}$ with team semantics include

- Inclusion: $\mathfrak{A} \models_T \mathbf{x} \subseteq \mathbf{y}$ iff $T(\mathbf{x}) \subseteq T(\mathbf{y})$.
- Exclusion: $\mathfrak{A} \models_T \mathbf{x} \mid \mathbf{y}$ iff $T(\mathbf{x}) \cap T(\mathbf{y}) = \varnothing$.
- Conditional independence: $\mathfrak{A} \models_T \mathbf{x} \bot_{\mathbf{z}} \mathbf{y}$ iff for all $s, t \in T$ with $s(\mathbf{z}) = t(\mathbf{z})$ there exists $u \in T$ with $u(\mathbf{z}) = s(\mathbf{z}) = t(\mathbf{z})$ and $u(\mathbf{xy}) = s(\mathbf{x})t(\mathbf{y})$.

For some $\mathscr{C} \subseteq \{\mathrm{dep}, \subseteq, |, \bot\}$ we write $\mathsf{FO}(\mathscr{C})$ to denote the logic one obtains by adding the atoms in $\mathscr{C}$ to $\mathsf{FO}$ with team semantics. Inclusion and exclusion were introduced in [16], and independence logic $\mathsf{FO}(\bot)$ was proposed in [21]. The following hierarchy with respect to all formulae was shown in [16]:

$$\mathsf{FO} \subsetneq \mathsf{FO}(\mathrm{dep}) = \mathsf{FO}(|) \subsetneq \mathsf{FO}(\subseteq, |) = \mathsf{FO}(\bot).$$

Hence we will freely use dependence atoms within $\mathsf{FO}(\subseteq, |) = \mathsf{FO}(\perp)$. Apart from atoms we may also add new connectives such as

- Classical negation: $\mathfrak{A} \models_T \mathsf{not}\, \psi$ iff $\mathfrak{A} \not\models_T \psi$.
- Classical disjunction: $\mathfrak{A} \models_T \psi_1 \sqcup \psi_2$ iff $\mathfrak{A} \models_T \psi_1$ or $\mathfrak{A} \models_T \psi_2$.

Classical disjunction can be defined in team semantics by constancy atoms; in restriction to models with at least two elements, it holds for $\psi_1, \psi_2 \in \mathsf{FO}(\subseteq, |)$ that

$$\psi_1 \sqcup \psi_2 \quad \equiv \quad \exists c \exists d(\mathrm{dep}(c) \land \mathrm{dep}(d) \land [(c = d \land \psi_1) \lor (c \neq d \land \psi_2)]).$$

Here $c$ and $d$ are fresh variables not occurring in $\psi_1, \psi_2$ or the domain of $T$, and $\mathrm{dep}(c) \coloneqq \mathrm{dep}((), c)$ denotes dependence on the empty tuple of variables, i.e. that $c$ is constant in the team, like $D_\varnothing c$ would do for $\mathsf{LFD}$. Note that if we only have one element, we can also only have a single assignment, and then team disjunction $\lor$ coincides with classical disjunction $\sqcup$. Hence we can freely use classical disjunction $\sqcup$ for formulae in $\mathsf{FO}(\subseteq, |)$.

The following properties are often used to compare logics with team semantics.

**Definition 4.53** (Downward closure and flatness)**.** The logic $\mathsf{FO}(\mathscr{C})$ satisfies downward closure if for all $\psi \in \mathsf{FO}(\mathscr{C})$ and fitting $\mathfrak{A}$ with team $T$ we have

$$\mathfrak{A} \models_T \psi \qquad \text{implies} \qquad \mathfrak{A} \models_{T'} \psi, \quad T' \subseteq T.$$

The following stronger property is called flatness:

$$\mathfrak{A} \models_T \psi \qquad \text{iff} \qquad \mathfrak{A} \models_{\{s\}} \psi, \quad s \in T,$$

Here $\mathfrak{A} \models_{\{s\}} \psi$ is equivalent to $\mathfrak{A}, s \models \psi$ in the classical Tarski semantics of $\mathsf{FO}$ with a single assignment $s$.

**Definition 4.54.** We say that $\mathsf{FO}(\mathscr{C})$ is local if for all $\psi \in \mathsf{FO}(\mathscr{C})$ and every set of variables $X$ with $\mathrm{Free}(\psi) \subseteq X$ we have

$$\mathfrak{A} \models_T \psi \qquad \text{iff} \qquad \mathfrak{A} \models_{(T \upharpoonright X)} \psi$$

for all suitable $\mathfrak{A}$ and $T$.

**Lemma 4.55.** $\mathsf{FO}$ satisfies flatness, and $\mathsf{FO}(\mathrm{dep})$ satisfies downward closure.

*Proof.* See [34, Proposition 3.35] and [34, Proposition 3.10]. □

**Lemma 4.56.** In our setting of lax team semantics, $\mathsf{FO}(\subseteq, |) = \mathsf{FO}(\perp)$ is local.

*Proof.* See [16, Theorem 4.22]. ☐

For a more complete overview of the mentioned extensions of FO with team semantics, we refer the reader to [34, 1, 16].

We now want to give a translation of LFD into a logic with team semantics. The desired form is as follows, where $(\tau, V)$ is a finite type as usual. Given $\mathbf{M}, s \in \mathcal{DEP}[\tau, V]$ and $\varphi \in \mathsf{LFD}(\tau, V)$ we want to define $\mathsf{tr}_{\text{team}}(\varphi)$ and a fitting team $T_{\mathbf{M},s}$ over $\mathfrak{M}$ such that

$$\mathbf{M}, s \models \varphi \qquad \text{iff} \qquad \mathfrak{M} \models_{T_{\mathbf{M},s}} \mathsf{tr}_{\text{team}}(\varphi).$$

Let $\mathbf{v}$ be a tuple which enumerates $V$ and $\tilde{\mathbf{v}}$ a fresh copy of $\mathbf{v}$. These will have the fixed meaning that assignments of the team $T_\mathbf{M}$ are stored within $T_{\mathbf{M},s}(\mathbf{v}) = T_\mathbf{M}(\mathbf{v})$ and the current assignment $s$ is stored within $T_{\mathbf{M},s}(\tilde{\mathbf{v}}) = \{s(\mathbf{v})\}$, so $\tilde{\mathbf{v}}$ will be constant throughout $T_{\mathbf{M},s}$. Formally

$$T_{\mathbf{M},s} \coloneqq T_\mathbf{M}[\tilde{\mathbf{v}} \mapsto s(\mathbf{v})] = \{\mathbf{v}\tilde{\mathbf{v}} \mapsto t(\mathbf{v})s(\mathbf{v}) \mid t \in T_\mathbf{M}\}.$$

**Lemma 4.57.** Under the above constraints of how the translation should look like, it is not possible to embed LFD into dependence logic FO(dep).

*Proof.* The crucial difference is that by Lemma 4.55 dependence logic is downwards-closed, whereas LFD is not (in a suitably adapted sense). Indeed it is generally not the case that $(\mathfrak{M}, T) \models \varphi$ entails $(\mathfrak{M}, T') \models \varphi$ for sentences $\varphi \in \mathsf{LFD}$ and subteams $T' \subseteq T$. As an example, consider the negated global dependence $\varphi = \neg \mathsf{A}\, D_x y$. Clearly a full dependence model with universe $\{0, 1\}$ satisfies $\varphi$, whereas the model with its team restricted to a single assignment does not satisfy $\varphi$.

With how we defined $T_{\mathbf{M},s}$, the downwards-closure of $\mathsf{tr}_{\text{team}}(\varphi)$ would yield the downwards-closure of $\varphi$ in the above sense, a contradiction. Since we know from Lemma 4.55 that dependence logic is downwards-closed, the claim follows. ☐

We will instead give a translation into $\mathsf{FO}(\subseteq, |) = \mathsf{FO}(\perp)$ that was communicated to the author by Richard Wilke.[6] For the translation of quantifiers $\mathsf{D}_X$ and $\mathsf{E}_X$ we will need another copy $\mathbf{v}'$ of $\mathbf{v}$, which will be considered fresh every time we use it, meaning all its variables have not been used before. Other tuples of variables are assumed to be contained within one of $\mathbf{v}, \tilde{\mathbf{v}}, \mathbf{v}'$ and will be annotated accordingly, e.g. $\mathbf{x}, \tilde{\mathbf{x}}, \mathbf{x}'$ are pairwise disjoint copies of another with $[\mathbf{x}] \subseteq [\mathbf{v}]$, $[\tilde{\mathbf{x}}] \subseteq [\tilde{\mathbf{v}}]$ and $[\mathbf{x}'] \subseteq [\mathbf{v}']$. Similarly to $\mathbf{v}'$, we will also require fresh variables $\widehat{x}, \widehat{y}$ on a few occasions (and these should not be considered part of $\mathbf{v}$).

[6] Mathematical Foundations of Computer Science, RWTH Aachen University, Aachen, Germany. E-mail address: wilke@logic.rwth-aachen.de

Define for $\beta \in \mathsf{FO}(\subseteq, |)$ and $\mathsf{FO}$-literals $\alpha, \gamma$ the auxiliary notation

$$\alpha \to \beta := \neg\alpha \vee (\alpha \wedge \beta) \qquad \text{and} \qquad \alpha \leftrightarrow \gamma := (\alpha \wedge \gamma) \vee (\neg\alpha \wedge \neg\gamma).$$

Note that we can define non-constancy in $\mathsf{FO}(\subseteq)$ by stating that a variable $z$ takes at least two different values: for fresh variables $\widehat{x}, \widehat{y}$ we have

$$\mathsf{not}\,\mathrm{dep}(z) \equiv \exists\widehat{x}\exists\widehat{y}(\widehat{x} \neq \widehat{y} \wedge \widehat{x} \subseteq z \wedge \widehat{y} \subseteq z).$$

Given $\varphi \in \mathsf{LFD}(\tau, V)$ in negation normal form we construct $\mathsf{tr}_{\mathrm{team}}(\varphi)$ as follows. The rules for literals, boolean connectives, and dependence atoms are straightforward:

- $\mathsf{tr}_{\mathrm{team}}(\varphi) = \varphi[\mathbf{v} \mapsto \widetilde{\mathbf{v}}]$ whenever $\varphi$ is a relational literal. Here $\varphi[\mathbf{v} \mapsto \widetilde{\mathbf{v}}]$ denotes the formula one obtains from $\varphi$ by replacing each variable of $\mathbf{v}$ with its counterpart in $\widetilde{\mathbf{v}}$.
- $\mathsf{tr}_{\mathrm{team}}(\varphi_1 \wedge \varphi_2) = \mathsf{tr}_{\mathrm{team}}(\varphi_1) \wedge \mathsf{tr}_{\mathrm{team}}(\varphi_2)$.
- $\mathsf{tr}_{\mathrm{team}}(\varphi_1 \vee \varphi_2) = \mathsf{tr}_{\mathrm{team}}(\varphi_1) \sqcup \mathsf{tr}_{\mathrm{team}}(\varphi_2)$.
- $\mathsf{tr}_{\mathrm{team}}(D_{\mathbf{x}}y) = (\mathbf{x} = \widetilde{\mathbf{x}}) \to (y = \widetilde{y})$.
- $\mathsf{tr}_{\mathrm{team}}(\neg D_{\mathbf{x}}y) = (\mathbf{x} = \widetilde{\mathbf{x}}) \to \mathsf{not}\,\mathrm{dep}(y)$.

For translating $\mathsf{E}_{\mathbf{x}}\,\varphi$ we need to choose an assignment $t$ from our team that agrees with the current assignment $s$ on $\mathbf{x}$, and then evaluate $\mathsf{tr}_{\mathrm{team}}(\varphi)$ on something like $T_{\mathbf{M},t}$, where the values of the current assignment are updated to $t(\mathbf{v})$.

Let $\mathbf{v}'$ be a fresh copy of $\mathbf{v}$. The idea is that we quantify $\mathbf{v}'$ suitably to represent our next assignment $t \in T_{\mathbf{M}}$ with $t =_{\mathbf{x}} s$. For this, we state that $\mathbf{v}'$ is constant, and that its tuple of values occurs in our team, via $\mathbf{v}' \subseteq \mathbf{v}$. The requirement $t =_{\mathbf{x}} s$ is then expressed as $\mathbf{x}' = \widetilde{\mathbf{x}}$, since $s$ is represented by $\widetilde{\mathbf{v}}$. Then, we want to evaluate $\mathsf{tr}_{\mathrm{team}}(\varphi)$ with the new current assignment stored in $\mathbf{v}'$. Overall, we define

$$\mathsf{tr}_{\mathrm{team}}(\mathsf{E}_{\mathbf{x}}\,\varphi) = \exists\mathbf{v}'\Big(\bigwedge_{v' \in [\mathbf{v}']} \mathrm{dep}(v') \wedge \mathbf{v}' \subseteq \mathbf{v} \wedge \mathbf{x}' = \widetilde{\mathbf{x}} \wedge \mathsf{tr}_{\mathrm{team}}(\varphi)[\widetilde{\mathbf{v}} \mapsto \mathbf{v}']\Big).$$

As before, $\mathsf{tr}_{\mathrm{team}}(\varphi)[\widetilde{\mathbf{v}} \mapsto \mathbf{v}']$ represents the formula one obtains from $\mathsf{tr}_{\mathrm{team}}(\varphi)$ by replacing all variables in $\widetilde{\mathbf{v}}$ with their copies in $\mathbf{v}'$.

Formulae of the form $\mathsf{D}_{\mathbf{x}}\,\varphi \in \mathsf{LFD}$ usually need to be evaluated at multiple assignments, namely at those in the $=_{\mathbf{x}}$-class of the current assignment. Hence, for translating $\mathsf{D}_{\mathbf{x}}\,\varphi$, we construct a team that is analogous to the union of all $T_{\mathbf{M},t}$ for $t$ in the $=_{\mathbf{x}}$-class of $s$. To preserve our intended interpretation of the team, we then want to evaluate $\mathsf{tr}_{\mathrm{team}}(\varphi)$ separately on each such $T_{\mathbf{M},t}$. For this we introduce the following auxiliary concept.

**Definition 4.58.** Consider a $\psi \in \mathsf{FO}(\subseteq, |)$ and fitting structure $\mathfrak{A}$ with team $T$. For some tuple $\mathbf{x}$ over the domain of $T$, let

$$\mathfrak{A} \models_T \Gamma\mathbf{x}.\psi \qquad \text{iff} \qquad \mathfrak{A} \models_{(T \restriction \mathbf{x}=\mathbf{a})} \psi, \quad \mathbf{a} \in T(\mathbf{x}),$$

where $(T \restriction \mathbf{x} = \mathbf{a}) \coloneqq \{t \in T \mid t(\mathbf{x}) = \mathbf{a}\}$.

**Lemma 4.59.** In the above setting, if none of the variables $x \in [\mathbf{x}]$ are quantified within $\psi$, then $\Gamma\mathbf{x}.\psi$ is equivalent to some formula in $\mathsf{FO}(\subseteq, |)$.

*Proof.* First note that for all (tuples of) variables $\mathbf{x}, \mathbf{y}, \mathbf{z}$ and variables $y$ in the domain of $T$ we have

- $\Gamma\mathbf{x}.\operatorname{dep}(\mathbf{z}, y) \equiv \operatorname{dep}(\mathbf{xz}, y)$.
- $\Gamma\mathbf{x}.(\mathbf{z} \subseteq \mathbf{y}) \equiv (\mathbf{xz} \subseteq \mathbf{xy})$.
- $\Gamma\mathbf{x}.(\mathbf{z} \mid \mathbf{y}) \equiv (\mathbf{xz} \mid \mathbf{xy})$.
- If $\psi \in \mathsf{FO}$, then $\Gamma\mathbf{x}.\psi \equiv \psi$ by flatness of $\mathsf{FO}$ (cf. Definition 4.53).

Furthermore, for $\psi_1, \psi_2 \in \mathsf{FO}(\subseteq, |)$ it is easy to see that

$$\Gamma\mathbf{x}.(\psi_1 \wedge \psi_2) \quad \equiv \quad (\Gamma\mathbf{x}.\psi_1) \wedge (\Gamma\mathbf{x}.\psi_2).$$

This also holds for (team) disjunction:

If $\mathfrak{A} \models_T \Gamma\mathbf{x}.(\psi_1 \vee \psi_2)$, then for every $\mathbf{a} \in T(\mathbf{x})$ we can split $(T \restriction \mathbf{x} = \mathbf{a}) = T^1_{\mathbf{a}} \cup T^2_{\mathbf{a}}$ such that $\mathfrak{A} \models_{T^i_{\mathbf{a}}} \psi_i$. If we define $T^i \coloneqq \bigcup_{\mathbf{a} \in T(\mathbf{x})} T^i_{\mathbf{a}}$, then clearly $(T^i \restriction \mathbf{x} = \mathbf{a}) = T^i_{\mathbf{a}}$ and hence $\mathfrak{A} \models_{T^i} \Gamma\mathbf{x}.\psi_i$. Since also $T^1 \cup T^2 = T$, we get $\mathfrak{A} \models_T (\Gamma\mathbf{x}.\psi_1) \vee (\Gamma\mathbf{x}.\psi_2)$.

Conversely, if $\mathfrak{A} \models_T (\Gamma\mathbf{x}.\psi_1) \vee (\Gamma\mathbf{x}.\psi_2)$, then we find $T^1 \cup T^2 = T$ with $\mathfrak{A} \models_{T^i} \Gamma\mathbf{x}.\psi_i$. Define $T^i_{\mathbf{a}} \coloneqq (T^i \restriction \mathbf{x} = \mathbf{a})$. Then $\mathfrak{A} \models_{T^i_{\mathbf{a}}} \psi_i$ and $T^1_{\mathbf{a}} \cup T^2_{\mathbf{a}} = (T \restriction \mathbf{x} = \mathbf{a})$, so we have $\mathfrak{A} \models_{(T \restriction \mathbf{x}=\mathbf{a})} \psi_1 \vee \psi_2$, for all $\mathbf{a} \in T(\mathbf{x})$. Thus $\mathfrak{A} \models_T \Gamma\mathbf{x}.(\psi_1 \vee \psi_2)$.

This proves

$$\Gamma\mathbf{x}.(\psi_1 \vee \psi_2) \quad \equiv \quad (\Gamma\mathbf{x}.\psi_1) \vee (\Gamma\mathbf{x}.\psi_2).$$

Moreover, if $y \notin [\mathbf{x}]$, then

$$(T \restriction \mathbf{x} = \mathbf{a})[y \mapsto A] = (T[y \mapsto A] \restriction \mathbf{x} = \mathbf{a})$$

and analogously for suitable Skolem extensions $F \colon T \to \mathcal{P}(A) \setminus \{\varnothing\}$ it holds that

$$(T \restriction \mathbf{x} = \mathbf{a})[y \mapsto F] = (T[y \mapsto F] \restriction \mathbf{x} = \mathbf{a}).$$

This shows that whenever $y \notin [\mathbf{x}]$, we have

$$\Gamma\mathbf{x}.\forall y\psi \equiv \forall y\Gamma\mathbf{x}.\psi \qquad \text{and} \qquad \Gamma\mathbf{x}.\exists y\psi \equiv \exists y\Gamma\mathbf{x}.\psi.$$

Therefore, since $\psi$ does not quantify any variables in $[\mathbf{x}]$, we can transform $\Gamma\mathbf{x}.\psi$ to an equivalent formula in $\mathsf{FO}(\subseteq, |)$ by iteratively distributing $\Gamma\mathbf{x}$ over connectives and quantifiers down to first-order formulae or dependence / inclusion / exclusion atoms, where we then use the equivalences given at the start of this proof. ☐

With this we can give the last definition for completely specifying $\mathsf{tr}_{\text{team}}$. Again let $\mathbf{v}'$ be a fresh copy of $\mathbf{v}$. We universally quantify $\mathbf{v}'$, and consider only the subteam where $\mathbf{v}'$ represents assignments in our team $T_{\mathbf{M},s}(\mathbf{v}) = T_{\mathbf{M}}(\mathbf{v})$, i.e. where $\mathbf{v}' \subseteq \mathbf{v}$. As for $\mathsf{E}_{\mathbf{x}}\, \varphi$, we encode agreement on $\mathbf{x}$ with the current assignment by stating $\mathbf{x}' = \widetilde{\mathbf{x}}$. Finally, we let $\mathbf{v}'$ replace the role of $\widetilde{\mathbf{v}}$ as the new current assignment in $\mathsf{tr}_{\text{team}}(\varphi)$, and use $\Gamma\mathbf{v}'$ to evaluate $\mathsf{tr}_{\text{team}}(\varphi)$ at each assignment in the $=_{\mathbf{x}}$-class of $s$:

$$\mathsf{tr}_{\text{team}}(\mathsf{D}_{\mathbf{x}}\, \varphi) = \forall \mathbf{v}'((\mathbf{v}' \mid \mathbf{v}) \vee (\mathbf{v}' \subseteq \mathbf{v} \wedge (\mathbf{x}' = \widetilde{\mathbf{x}} \to \Gamma\mathbf{v}'.(\mathsf{tr}_{\text{team}}(\varphi)[\widetilde{\mathbf{v}} \mapsto \mathbf{v}'])))).$$

**Proposition 4.60.** Our translation of $\mathsf{LFD}$ into $\mathsf{FO}(\subseteq, |)$ works as intended. Namely, for every $\varphi \in \mathsf{LFD}$ and fitting dependence model $\mathbf{M}, s \in \mathcal{DEP}$ with $\mathbf{M} = (\mathfrak{M}, T_{\mathbf{M}})$ we have $\mathsf{tr}_{\text{team}}(\varphi) \in \mathsf{FO}(\subseteq, |)$ and

$$\mathbf{M}, s \models \varphi \qquad \text{iff} \qquad \mathfrak{M} \models_{T_{\mathbf{M},s}} \mathsf{tr}_{\text{team}}(\varphi).$$

*Proof.* This is shown via an induction on $\varphi$. The case for literals and boolean connectives is clear, whereas the idea of the translation of dependence quantifiers is as explained in this section. The details can be found in Proposition C.5 in the appendix. ☐

**Remark 4.61.** As for $\mathsf{tr}_{\text{st}}$ it is trivial to extend $\mathsf{tr}_{\text{team}}$ to $\mathsf{LFD}^{=}$ by stating $\mathsf{tr}_{\text{team}}(x = y) = (\widetilde{x} = \widetilde{y})$.

# Chapter 5

# Complexity Results

## 5.1 Satisfiability

The satisfiability problem Sat($\mathsf{LFD}$) is defined as usual; given $\varphi \in \mathsf{LFD}$, decide whether $\varphi$ is satisfiable, and measure the complexity with respect to the length $|\varphi|$ of $\varphi$. The proof of decidability of $\mathsf{LFD}$ in [6, Section 4] already yields an elementary upper bound which we specify below.

We assume familiarity with big-O-notation; for our purposes it suffices to say that for some non-decreasing $f\colon \mathbb{N} \to \mathbb{N}$ the notation $\mathcal{O}(f(n))$ describes the set of non-decreasing functions $\mathbb{N} \to \mathbb{N}$ that are asymptotically bounded by $f$ (as $n \to \infty$). Moreover, we use the common notation

$$2^{\mathcal{O}(n)} := \bigcup_{k\in\mathbb{N}} \mathcal{O}(2^{kn}) \qquad \text{and} \qquad 2^{2^{\mathcal{O}(n)}} := \bigcup_{k\in\mathbb{N}} \mathcal{O}(2^{2^{kn}}).$$

Note that if $p$ is some polynomial and $f \in 2^{\mathcal{O}(n)}$, then $n \mapsto p(f(n))$ is still in $2^{\mathcal{O}(n)}$. This is because $f \in \mathcal{O}(2^{kn})$ and $p \in \mathcal{O}(n^\ell)$ for some fixed $k, \ell \in \mathbb{N}$, and hence $p(f) \in \mathcal{O}((2^{kn})^\ell) = \mathcal{O}(2^{\ell kn}) \subseteq 2^{\mathcal{O}(n)}$. An analogous result holds for $2^{2^{\mathcal{O}(n)}}$. We will also use

$$2^{2^{\mathcal{O}(n)}} \cdot 2^{\mathcal{O}(n)} := \bigcup \{\mathcal{O}(f(n)g(n)) \mid f \in 2^{2^{\mathcal{O}(n)}}, g \in 2^{\mathcal{O}(n)}\} = 2^{2^{\mathcal{O}(n)}}$$

which is clear by $2^{2^{kn}} \leq 2^{2^{kn}}2^{\ell n} = 2^{\ell n + 2^{kn}} \leq 2^{2^{(k+\ell)n}}$ for large enough $n$ and fixed $k, \ell$. We define and prove $2^{\mathcal{O}(n)}n = 2^{\mathcal{O}(n)}$ in the same fashion.

**Proposition 5.1.** Sat($\mathsf{LFD}$) $\in$ 2-NEXPTIME.

*Proof.* In Appendix B we cite the relevant definitions from [6, Section 4] and give most complexity considerations used here.

Assume we have the input $\varphi \in \mathsf{LFD}$ with $n = |\varphi|$ and $V_\varphi$ being the set of variables that occur in $\varphi$. Let $\Phi$ be a set that contains all subformulae of $\varphi$ all dependence atoms $D_X Y$[1] for $X, Y \subseteq V_\varphi$, and is closed under one round of negations, where explicit negations themselves are left as they are. Remark B.1 gives us the bound $|\Phi| \in 2^{\mathcal{O}(n)}$. Type models $\mathfrak{M}$ for $\Phi$ (cf. Definition B.6) are certain sets of subsets of $\Phi$. Remark B.7 gives us the bound $|\mathfrak{M}| \in 2^{2^{\mathcal{O}(n)}}$ for these. [6, Theorem 4.9] shows that if there exists a type model $\mathfrak{M}$ for $\Phi$ such that there is some $\Delta \in \mathfrak{M}$ with $\varphi \in \Delta$, then $\varphi$ is satisfiable.

Now we decide whether $\varphi$ is satisfiable via the following steps:

1. Nondeterministically guess a set $\mathfrak{M}$ of subsets of $\Phi$. This takes nondeterministic time $|\mathfrak{M}| \in 2^{2^{\mathcal{O}(n)}}$.
2. Check that there is some $\Delta \in \mathfrak{M}$ with $\varphi \in \Delta$. This can be done in time that is linear in $|\mathfrak{M}|$, so in deterministic time $2^{2^{\mathcal{O}(n)}}$.
3. Lastly we can verify that $\mathfrak{M}$ is a type model for $\Phi$ in deterministic time $2^{2^{\mathcal{O}(n)}}$ (cf. Corollary B.9).

Overall, we can check satisfiability of $\varphi$ in nondeterministic time $2^{2^{\mathcal{O}(n)}}$, so we obtain $\mathrm{Sat}(\mathsf{LFD}) \in 2\text{-}\textsc{Nexptime} = \bigcup_{k\in\mathbb{N}} \textsc{Ntime}(2^{2^{n^k}})$. ☐

**Proposition 5.2.** $\mathrm{Sat}(\mathsf{LFD})$ is $\textsc{Pspace}$-hard.

*Proof.* We give a polynomial reduction from the totally quantified boolean formula problem TQBF, also known as QBF or QSAT. The input of this problem is of the form

$$\psi = Q_1 x_1 Q_2 x_2 \cdots Q_n x_n \varphi(x_1, \dots, x_n)$$

where $Q_i \in \{\forall, \exists\}$ and $\varphi(x_1, \dots, x_n)$ is a boolean formula. The semantics should be clear and the decision problem is whether this input evaluates to true. It is well known that this problem is $\textsc{Pspace}$-complete (cf. [4, Theorem 3.9] or [33, Theorem 19.1]). We will encode boolean values by means of a predicate $R$, i.e. if $Rx$ holds, we consider $x$ to be 1, and otherwise 0. The intended model has universe $\{0, 1\}$, $R$ interpreted as $\{1\}$, and a full team $\{0, 1\}^V$. To enforce the whole space of $2^n$ possible boolean assignments on arbitrary models $\mathbf{M}$, we state that for each assignment $s$ and each variable $x_i$ we can find assignments $t_0, t_1$ which agree with $s$ on everything *except* $x_i$, and such that $\mathbf{M}, t_0 \models \neg Rx_i$ and $\mathbf{M}, t_1 \models Rx_i$. This is expressed by the sentence

$$\vartheta := \mathsf{A} \bigwedge_{i=1}^{n} (\mathsf{E}_{V_i}\, Rx_i \wedge \mathsf{E}_{V_i} \neg Rx_i)$$

[1]Remember that $D_X Y$ is shorthand for $\bigwedge_{y\in Y} D_X y$, see Notation 2.6.

where we denote $V_i \coloneqq V \setminus \{x_i\}$. Clearly $\vartheta$ can be constructed in polynomial time. Now we transform $\psi$ into an $\mathsf{LFD}$-sentence via the following steps:

1. If $Q_1 = \forall$, then replace $Q_1 x_1$ by $\mathsf{A}$, otherwise by $\mathsf{E}$.
2. Replace the subsequent $Q_i x_i$ by $\mathsf{D}_{V_i}$ if $Q_i = \forall$, and otherwise by $\mathsf{E}_{V_i}$.
3. Finally replace each occurrence of $x_i$ in $\varphi$ by $Rx_i$.

The result is an $\mathsf{LFD}$-sentence $\psi^*$, and our reduction sends $\psi$ to $\vartheta \wedge \psi^* \in \mathsf{LFD}$. As an example, consider $\psi = \forall x \exists y \forall z (x \vee y \vee z) \in$ TQBF. Then

$$\vartheta = \mathsf{A}(\mathsf{E}_{yz}\, Rx \wedge \mathsf{E}_{yz}\, \neg Rx \quad \wedge \quad \mathsf{E}_{xz}\, Ry \wedge \mathsf{E}_{xz}\, \neg Ry \quad \wedge \quad \mathsf{E}_{xy}\, Rz \wedge \mathsf{E}_{xy}\, \neg Rz)$$

and

$$\psi^* = \mathsf{A}\, \mathsf{E}_{xz}\, \mathsf{D}_{xy}(Rx \vee Ry \vee Rz).$$

It is straightforward to verify the correctness of this reduction. Therefore we obtain TQBF $\leq_p$ Sat($\mathsf{LFD}$), and thus the PSPACE-hardness of Sat($\mathsf{LFD}$). ☐

**Remark 5.3.** We attempted but were not able to adapt a proof for the NEXPTIME-hardness of Sat($\mathsf{FO}^2$). The proof we considered is due to Fürer [15] and Lewis [30], and uses a reduction from domino tilings on exponentially large grids. As we have seen at the start of Section 2.3, the fact that we can assume without loss of generality that our dependence models are distinguished (cf. Fact 2.12), creates some difficulties when we try to enforce confluence within dependence models. Thus, one might try this at the level of the assignments instead, from the modal perspective. But here, it is precisely this modal character which causes difficulties. As was already discussed on various occasions throughout this thesis, we cannot access information about two unrelated assignments (in the sense that they do not agree on any variable) simultaneously. To guarantee a valid tiling, one needs to state that all objects (assignments) sharing neighbouring coordinates are tiled by compatible dominos. But the coordinates are usually encoded as monadic predicates $X_0, Y_0, \ldots, X_n, Y_n$, and such objects (assignments) with neighbouring coordinates may not be related by any $\sim_x$ (i.e. the assignments may be disjoint). Hence a quantification of the form

$$\forall s \forall t (\text{neighbour}(s,t) \rightarrow \text{compatibleTiles}(s,t))$$

is necessary, which must allow $s$ and $t$ to be disjoint, and sentences of this form are generally not equivalent to any (translated) $\mathsf{LFD}$-sentences, see e.g. the end of Sections 4.1.2 and 4.2.

Overall, this begs the question whether there is a suitable variant of tiling arguments that can be expressed within $\mathsf{LFD}$.

## 5.2 Model Checking

The model checking problem for $\mathsf{FO}$, denoted $\mathrm{MC}(\mathsf{FO})$, is defined as follows: Given a formula $\psi(\mathbf{x}) \in \mathsf{FO}$, a fitting finite structure $\mathfrak{A}$, and constants $\mathbf{a}$ in $\mathfrak{A}$ interpreting the free variables $\mathbf{x}$ of $\psi$, determine whether $\mathfrak{A} \models \psi(\mathbf{a})$. The complexity is measured in the size of all inputs, i.e. the length $|\psi|$ of $\psi$ and the size $|\mathfrak{A}|$ of the structure $\mathfrak{A}$. The problems $\mathrm{MC}(\mathsf{ML}), \mathrm{MC}(\mathsf{GF})$ or $\mathrm{MC}(\mathsf{FO}^k)$ for $k \in \mathbb{N}$ are defined analogously to $\mathrm{MC}(\mathsf{FO})$.

**Remark 5.4.** Analogously to Remark B.1 which we used for the proof of Proposition 5.1, we want to remark that we can consider $|\mathfrak{A}|$ to be the cardinality of the universe $A$ of $\mathfrak{A}$. This is because $\tau$ is finite, and hence the actual length of a reasonable encoding of $\mathfrak{A}$ with all relations is still polynomial in the cardinality of $A$. Since we are only interested in complexity classes, a polynomially larger input makes no difference; consider for example that $\log(n^k) = k \log n \in \mathcal{O}(\log n)$ for fixed $k$, meaning that if something is logarithmic in $n^k$, then it also logarithmic in $n$. Likewise, if something is polynomial in $n^k$, it is obviously still polynomial in $n$.

**Theorem 5.5.** The problems $\mathrm{MC}(\mathsf{ML})$ and $\mathrm{MC}(\mathsf{FO}^k)$ for $k \geq 2$ are PTIME-complete whereas $\mathrm{MC}(\mathsf{FO})$ is PSPACE-complete.

*Proof.* See [19, Section 4]. □

Later these results were extended to guarded first-order and fixed point logics in [10], showing in particular that $\mathrm{MC}(\mathsf{GF})$ is PTIME-complete.

Henceforth let $\mathcal{L}$ denote $\mathsf{LFD}$ or $\mathsf{LFD}^=$. The model checking problem for $\mathcal{L}$ is essentially the same; as input we are given some $\psi \in \mathcal{L}(\tau, V)$ and finite $\mathbf{M}, s \in \mathcal{DEP}[\tau, V]$ for some finite type $(\tau, V)$. The task is to decide whether $\mathbf{M}, s \models \psi$. We let $\mathfrak{M}$ denote the underlying $\tau$-structure of $\mathbf{M}$ with finite universe $M$. It turns out however that the complexity is largely influenced by how we encode the team $T_{\mathbf{M}}$ of $\mathbf{M}$. Before we discuss these technicalities, we will consider the special case of full models.

**Remark 5.6.** $\mathsf{LFD}$ over full dependence models (of finite type) corresponds to relational first-order logic without equality; the quantifiers $\exists x$ and $\forall x$ respectively have the same semantics as $\mathsf{E}_{V_x}$ and $\mathsf{D}_{V_x}$ over full dependence models, where $V_x := V \setminus \{x\}$. A version of this was already used in the proof of Proposition 5.2.

It is therefore not surprising that for this special case, we obtain the same lower bounds as for first-order model checking.

**Definition 5.7.** The decision problem $\mathrm{MC}_{\mathrm{full}}(\mathcal{L})$ is the restriction of model checking for $\mathcal{L}$ to instances where the team is always the full team. More formally:

1. The inputs are tuples $(\psi, \mathfrak{M}, \mathbf{v}, s)$ where
   (a) $\psi \in \mathcal{L}(\tau, V)$ for some finite type $(\tau, V)$.
   (b) $\mathfrak{M}$ is a $\tau$-structure with finite universe $M$.
   (c) $\mathbf{v}$ is an enumeration of $V$.
   (d) $s \in T$ is the current assignment, encoded as the tuple $s(\mathbf{v})$.
2. The task is to decide whether $(\mathfrak{M}, M^V), s \models \psi$.
3. The complexity is measured in the size of the input, so essentially with respect to $|\psi| + |\mathfrak{M}| + |V|$.

For $k \in \mathbb{N}$ we also consider the variant $\mathrm{MC}^k_{\mathrm{full}}(\mathcal{L})$ where always $|V| \leq k$.

**Proposition 5.8.** $\mathrm{MC}_{\mathrm{full}}(\mathcal{L})$ is PSPACE-hard.

*Proof.* We can adapt our reduction from TQBF to Sat(LFD) in the proof of Proposition 5.2 to obtain a reduction from TQBF to $\mathrm{MC}_{\mathrm{full}}(\mathcal{L})$. For a given input

$$\psi = Q_1 x_1 \cdots Q_n x_n \varphi(x_1, \ldots, x_n)$$

with $Q_i \in \{\forall, \exists\}$, construct the input $(\psi^*, \mathfrak{M}, \mathbf{v}, s)$ for $\mathrm{MC}_{\mathrm{full}}(\mathcal{L})$, where

- $\psi^*$ is defined in the exact same way as in the proof of Proposition 5.2, namely by replacing $Q_1 x_1$ with $\mathsf{A}$ or $\mathsf{E}$, the $Q_i x_i$ with $\mathsf{D}_{V_i}$ or $\mathsf{E}_{V_i}$, and all $x_i$ in $\varphi$ by $Rx_i$. Here $V_i := V \setminus \{x_i\}$.
- $\mathfrak{M}$ is the intended model from the proof of Proposition 5.2. It consists of the universe $M = \{0, 1\}$ and a monadic predicate $R^{\mathfrak{M}} = \{1\}$.
- $\mathbf{v} = x_1, \ldots, x_n$.
- $s = (0, \ldots, 0)$ is irrelevant since $\psi^*$ is a sentence.

In the reduction TQBF $\leq_p$ Sat(LFD) of Proposition 5.2 we had to add another formula $\vartheta$ to enforce all possible boolean assignments on the variables to exist, but in the context of $\mathrm{MC}_{\mathrm{full}}(\mathcal{L})$ this is guaranteed by the full team. Now $\psi^*, \mathbf{v}$ and $s$ are clearly computable in polynomial time from the input, and $\mathfrak{M}$ is always the same, i.e. constant. This shows TQBF $\leq_p \mathrm{MC}_{\mathrm{full}}(\mathcal{L})$. □

**Proposition 5.9.** $\mathrm{MC}^k_{\mathrm{full}}(\mathcal{L})$ is PTIME-hard for $k \geq 2$.

*Proof.* The PTIME-hardness of $\mathrm{MC}(\mathsf{FO}^k)$ for $k \geq 2$ is shown in [19] by proving the PTIME-hardness for $\mathrm{MC}(\mathsf{ML})$ via a logspace-reduction from the GAME-problem[2] and noting that $\mathsf{ML}$ is a sublogic of $\mathsf{FO}^2$. For our case, note that $\mathsf{ML}$ (via the same standard translation) is in particular a sublogic of *relational* $\mathsf{FO}^2$ *without equality*. Hence the model checking problem for said fragment of $\mathsf{FO}^2$ must be PTIME-hard as well. We already discussed above that $\mathsf{LFD}$ over full models (of finite type) corresponds to relational $\mathsf{FO}$ without equality. Indeed, for $V = \{x, y\}$, it is trivial to give a logspace-computable translation from relational equality-free formulae $\psi \in \mathsf{FO}^2$ to equivalent formulae $\psi^* \in \mathsf{LFD}$, in the sense that for all suitable structures $\mathfrak{A}$ and assignments $s$

$$\mathfrak{A}, s \models \psi \qquad \text{iff} \qquad (\mathfrak{A}, A^V), s \models \psi^*.$$

Indeed, simply replace $\exists x$ by $\mathsf{E}_y$ and $\forall x$ by $\mathsf{D}_y$, likewise with $x$ and $y$ interchanged. This proves the PTIME-hardness of $\mathrm{MC}^k_{\mathrm{full}}(\mathcal{L})$ for $k \geq 2$. □

**Solving the model checking problem for $\mathcal{L}$.** First, we present a general method for solving the model checking problem for $\mathcal{L}$ that abstracts away the specifics on how the team is encoded. Without loss of generality we only deal with formulae in negation normal form, where negations are pushed inwards as far as possible (down to the atoms) by using the duality between $\mathsf{D}_X$ and $\mathsf{E}_X$ as well as de Morgan's laws.

We assume familiarity with alternating complexity classes.[3] In the proof of Theorem 5.5 in [19], an alternating algorithm for solving the first-order model checking problem was given. The following results were obtained.

**Fact 5.10** ([19])**.** There exists an alternating algorithm that solves instances $(\psi, \mathfrak{A}, \mathbf{a})$ of the first-order model checking problem requiring only

1. alternating space $\mathcal{O}(\log|\psi| + r \log|\mathfrak{A}|)$, where $r$ is the maximal number of free variables in any subformula of $\psi$, and

2. alternating time $\mathcal{O}(|\psi| \log|\mathfrak{A}|)$.

Together with the well-known results

$$\text{ALOGSPACE} = \text{PTIME} \qquad \text{and} \qquad \text{APTIME} = \text{PSPACE}$$

one then obtains that $\mathrm{MC}(\mathsf{FO}^k) \in$ PTIME, $k \in \mathbb{N}$ and $\mathrm{MC}(\mathsf{FO}) \in$ PSPACE. We adapted this algorithm to work for $\mathcal{L}$ in Algorithm 1.

---

[2]The GAME-problem is defined and shown to be PTIME-complete in [27].

[3]For a background in alternating complexity classes, see [5, Chapter 3] or [33, Chapters 16.2 & 19.1].

---

**Algorithm 1:** Alternating model checking for $\mathcal{L} \in \{\mathsf{LFD}, \mathsf{LFD}^=\}$

---

**ModelCheck($\psi, \mathbf{M}, s$)**
**Input:** a formula $\psi \in \mathcal{L}(\tau, V)$ in negation normal form where $(\tau, V)$ is finite, and a finite pointed dependence model $\mathbf{M}, s \in \mathcal{DEP}[\tau, V]$.

**if** $\psi$ is a relational literal or (in)equality **then**
  **if** $\mathbf{M}, s \models \psi$ **then** ACCEPT **else** REJECT

**if** $\psi = D_X y$ **then**
  **universally choose** $t \in T_\mathbf{M}$ with $t =_X s$
  **if** $t =_y s$ **then** ACCEPT **else** REJECT

**if** $\psi = \neg D_X y$ **then**
  **existentially guess** $t \in T_\mathbf{M}$ with $t =_X s$
  **if** $t \neq_y s$ **then** ACCEPT **else** REJECT

**if** $\psi = \eta \wedge \vartheta$ **then**
  **universally choose** $\varphi \in \{\eta, \vartheta\}$
  **ModelCheck($\varphi, \mathbf{M}, s$)**

**if** $\psi = \eta \vee \vartheta$ **then**
  **existentially guess** $\varphi \in \{\eta, \vartheta\}$
  **ModelCheck($\varphi, \mathbf{M}, s$)**

**if** $\psi = \mathsf{D}_X\, \varphi$ **then**
  **universally choose** $t \in T_\mathbf{M}$ with $t =_X s$
  **ModelCheck($\varphi, \mathbf{M}, t$)**

**if** $\psi = \mathsf{E}_X\, \varphi$ **then**
  **existentially guess** $t \in T_\mathbf{M}$ with $t =_X s$
  **ModelCheck($\varphi, \mathbf{M}, t$)**

---

It encodes the usual model checking game between an existential and universal player, played on positions $(\varphi, t)$ where $\varphi$ is some subformula of $\psi$ and $t \in T_\mathbf{M}$ an assignment. The initial position is $(\psi, s)$, and the players take turns according to the structure of the subformula at the current position. Algorithm 1 is run with these positions as inputs and implements the rules of this game; at positions where we existentially guess in the algorithm, the existential player, who is trying to show that $\mathbf{M}, s \models \psi$, makes a move. So he chooses disjuncts at disjunctions, assignments at positions $\mathsf{E}_X\, \varphi$ or $\neg D_X y$ and wins whenever the algorithm ACCEPTS, i.e. when we arrive at some $(\varphi, t)$ where $\varphi$ is a literal with $\mathbf{M}, t \models \varphi$. Dually, the universal player tries to show that $\mathbf{M}, s \not\models \psi$, so she tries to find counterexamples in the form of choosing conjuncts at conjunctions and assignments at positions $\mathsf{D}_X\, \varphi$ or $D_X y$. She wins at positions the algorithm REJECTS, so at positions $(\varphi, t)$ where $\varphi$ is a literal and $\mathbf{M}, t \not\models \varphi$. It is straightforward to verify the correctness of Algorithm 1. Essentially, the algorithm accepts if and only if the existential player has a winning strategy in this game (which happens if and only if $\mathbf{M}, s \models \psi$).

**How to encode the team.** Now we come back to the discussion on how to encode the team. A first idea might be the following.

**Definition 5.11.** We encode the team by listing all contained assignments. More specifically, define the problem $\mathrm{MC}_{\mathrm{list}}(\mathcal{L})$ as follows.

1. The inputs are tuples $(\psi, \mathfrak{M}, \mathbf{v}, s, T)$ where
   (a) $(\psi, \mathfrak{M}, \mathbf{v}, s)$ have the same interpretation as in Definition 5.7 for $\mathrm{MC}_{\mathrm{full}}(\mathcal{L})$.
   (b) $T \subseteq M^V$ is a team with $s \in T$, encoded as the list of tuples $T(\mathbf{v})$.
2. The task is to decide whether $(\mathfrak{M}, T), s \models \psi$.
3. The complexity is measured in the input size, so essentially with respect to $|\psi|+|\mathfrak{M}|+|V|+|T|$. Analogously to Remark 5.4 it makes no difference whether we consider $|T|$ as the cardinality of $T$ or the actual length of a reasonable encoding of $T$, since the latter is still polynomial in the input size.

It is easy to give examples that demonstrate a core problem with this approach.

**Example 5.12.** Given $\mathfrak{M}$ with universe $M = \{1, \ldots, 9\}$ and variables $V = \{x, y, z\}$, it is clear that a team such as

$$T = \{s \in M^V \mid s(x) \in \{1, 2, 3\},\ s(y) \in \{4, 5, 6\},\ s(z) \in \{7, 8, 9\}\}$$

can be encoded in more efficient ways than simply listing all assignments. For example, we could specify the rule set that was used in the set-builder notation. An even more drastic example is the full team $M^V$; an algorithm deciding membership of this team is trivial, whereas encoding the full team as a list leads to an exponential blowup, as it has size $|M|^{|V|}$ and $V$ is independent of $\mathfrak{M}$.

So given $\mathfrak{M}, \psi, V$ and a current assignment $s$, encoding a team $T \subseteq M^V$ as a list of all its assignments generally leads to an exponentially longer input. As a consequence of this, one obtains a deceptively low complexity for $\mathrm{MC}_{\mathrm{list}}(\mathcal{L})$.

**Proposition 5.13.** We can implement Algorithm 1 to decide instances $(\psi, \mathfrak{M}, \mathbf{v}, s, T)$ of $\mathrm{MC}_{\mathrm{list}}(\mathcal{L})$ with alternating workspace $\mathcal{O}(\log |\psi| + \log |T|)$.

*Proof.* We implement picking[4] assignments $t \in T$ by picking a pointer to some tuple in the list representing $T$. Such as pointer requires $\log |T|$ space.

[4] Here, "picking" refers to either existentially guessing or universally choosing.

The algorithm only needs to keep track of the current position $(\varphi, t)$, since $\psi, \mathfrak{M}$ and $T$ are never modified. A pointer of length $\log|\psi|$ suffices to specify the current subformula $\varphi$ of $\psi$. As mentioned above, we represent assignments by pointers of length $\log|T|$. We only ever need a constant number of assignments in the workspace, which proves the claim. □

**Corollary 5.14.** $\mathrm{MC}_{\mathrm{list}}(\mathcal{L}) \in$ ALOGSPACE $=$ PTIME.

**Remark 5.15.** Comparing the above to our result that $\mathrm{MC}_{\mathrm{full}}(\mathcal{L})$ is PSPACE-hard, these results seem to contradict the common belief that PTIME $\neq$ PSPACE. This is however not the case, as the two problems differ on the usual length of their inputs. Indeed, there cannot exist a polynomial-time reduction from $\mathrm{MC}_{\mathrm{full}}(\mathcal{L})$ to $\mathrm{MC}_{\mathrm{list}}(\mathcal{L})$, because the size of the full team $M^V$ is exponential in the size $|\psi| + |\mathfrak{M}| + |V|$ of the input of $\mathrm{MC}_{\mathrm{full}}(\mathcal{L})$.

Nevertheless, this disparity shows that the approach of encoding $T$ as a list is unsatisfying. We need some way to specify the team in a more compact way. A common approach is to encode the team as a first-order formula $\varphi_T$ over the vocabulary $\tau \cup M$ where all elements of $\mathfrak{M}$ are added as constants (cf. Definition 4.37). The idea is that for all tuples $\mathbf{t} \in M^{|V|}$ we have

$$\mathfrak{M}_M \models \varphi_T(\mathbf{t}) \qquad \text{iff} \qquad \text{there is some } t \in T \text{ with } t(\mathbf{v}) = \mathbf{t}.$$

As an example, consider the team $T$ stated in Example 5.12, which can be encoded as the formula

$$\varphi_T(x, y, z) = \left( \bigvee_{i \in \{1,2,3\}} x = i \right) \wedge \left( \bigvee_{i \in \{4,5,6\}} y = i \right) \wedge \left( \bigvee_{i \in \{7,8,9\}} z = i \right).$$

Listing all assignments would correspond to the disjunctive normal form of this $\varphi_T$, which is a lot longer than this conjunctive normal form.

**Definition 5.16.** We encode the team by a first-order formula as described above. More specifically, define the problem $\mathrm{MC}_{\mathrm{formula}}(\mathcal{L})$ as follows.

1. The inputs are tuples $(\psi, \mathfrak{M}, \mathbf{v}, s, \varphi_T)$ where
   (a) $(\psi, \mathfrak{M}, \mathbf{v}, s)$ have the same interpretation as in Definition 5.7 for $\mathrm{MC}_{\mathrm{full}}(\mathcal{L})$.
   (b) $\varphi_T \in \mathsf{FO}(\tau \cup M)$ encodes $T \subseteq M^V$ as described above; for all $\mathbf{t} \in M^{|V|}$:

$$\mathfrak{M}_M \models \varphi_T(\mathbf{t}) \qquad \text{iff} \qquad \text{there is some } t \in T \text{ with } t(\mathbf{v}) = \mathbf{t}.$$

2. The task is to decide whether $(\mathfrak{M}, T), s \models \psi$.

3. The complexity is measured in the input size, so essentially with respect to $|\psi| + |\mathfrak{M}| + |V| + |\varphi_T|$.

**Lemma 5.17.** We can implement Algorithm 1 so that a given instance $(\psi, \mathfrak{M}, \mathbf{v}, s, \varphi_T)$ for $\mathrm{MC}_{\mathrm{formula}}(\mathcal{L})$ is solved requiring only

1. alternating space $\mathcal{O}(\log|\psi| + (|V|+r)\log|\mathfrak{M}| + \log|\varphi_T|)$, where $r$ is the maximal number of free variables in any subformula of $\varphi_T$, and
2. alternating time $\mathcal{O}(|\psi| \cdot (|\psi| + |\mathfrak{M}| + (|V| + |\varphi_T|)\log|\mathfrak{M}|))$.

*Proof.* Assignments $t$ are encoded by their values $t(\mathbf{v})$, thus taking $\mathcal{O}(|V|\log|\mathfrak{M}|)$ space. We implement picking[5] assignments $t \in T$ by picking a tuple $\mathbf{t} \in M^{|V|}$ and then performing first-order model checking on $(\varphi_T, \mathfrak{M}_M, \mathbf{t})$. We know that $|\mathfrak{M}_M|$ is polynomial in $|\mathfrak{M}|$ and hence $\log|\mathfrak{M}_M| \in \mathcal{O}(\log|\mathfrak{M}|)$, see Remark 5.4. It follows from Fact 5.10 that picking an assignment in this way requires alternating space $\mathcal{O}((|V| + r)\log|\mathfrak{M}| + \log|\varphi_T|)$ and alternating time $\mathcal{O}((|V| + |\varphi_T|)\log|\mathfrak{M}|)$.

As in the proof of Proposition 5.13, we only need to keep track of the current position $(\varphi, t)$, and the current subformula $\varphi$ is specified via a pointer of length $\log|\psi|$. We only ever need to store a constant number of assignments, taking overall space $\mathcal{O}(|V|\log|\mathfrak{M}|)$. Because we can always reuse the space for the first-order model checking, this yields the claimed bound on the alternating space-complexity.

For the time analysis, note that testing whether $t =_X s$ for two assignments $s, t$ in the workspace is possible within alternating time $\mathcal{O}(|V|\log|\mathfrak{M}|)$. Furthermore:

1. If $\varphi$ is a relational literal or an (in)equality, we can evaluate whether $\mathbf{M}, s \models \varphi$ in alternating time $\mathcal{O}(|\psi| + |\mathfrak{M}| + |V|\log|\mathfrak{M}|)$.
2. In the case of dependence atoms or dependence quantifiers, the algorithm picks a new assignment $t \in T$, does a constant number of checks of the form $t =_X s$, and possibly updates the current assignment from $s$ to $t$ and the current subformula from $\mathsf{E}_X\,\varphi$ or $\mathsf{D}_X\,\varphi$ to $\varphi$. With the bounds we gave above this takes alternating time $\mathcal{O}(|\psi| + (|V| + |\varphi_T|)\log|\mathfrak{M}|)$.
3. Choosing some subformula at conjunctions and disjunctions takes only $\mathcal{O}(|\psi|)$ time, since we essentially just have to move our pointer within $\psi$.

At each recursive call we move to some subformula of the currently considered formula, so we have at most $|\psi|$ recursive calls. From the above points, we see that in each call the algorithm takes at most $\mathcal{O}(|\psi| + |\mathfrak{M}| + (|V| + |\varphi_T|)\log|\mathfrak{M}|)$ alternating time until it either terminates or invokes the next recursive call. This proves the claimed bound on the alternating time-complexity. □

[5]Here, "picking" refers to either existentially guessing or universally choosing.

**Corollary 5.18.** $\mathrm{MC}_{\mathrm{formula}}(\mathcal{L}) \in \textsc{Aptime} = \textsc{Pspace}$.

We want to show an analogue of $\mathrm{MC}(\mathsf{FO}^k) \in \textsc{Alogspace} = \textsc{Ptime}$. The idea is to define $\mathrm{MC}^k_{\mathrm{formula}}(\mathcal{L})$ as the restriction of $\mathrm{MC}_{\mathrm{formula}}(\mathcal{L})$ to instances where always $|V| \leq k$. So we need to prove that when $|V| \leq k$, the space-complexity given in Lemma 5.17 is logarithmic in the input. The problem is the occurrence of $r$, which describes the maximum number of free variables in any subformula of $\varphi_T$. It comes from the space-complexity of first-order model checking $(\varphi_T, \mathfrak{M}_M, \mathbf{t})$ for $\mathbf{t} \in M^{|V|}$. Currently, we allow arbitrary $\varphi_T \in \mathsf{FO}(\tau \cup M)$ to represent the team $T$ in the input. To obtain our wanted analogue, we need to bound $r$ by some constant for all instances of $\mathrm{MC}^k_{\mathrm{formula}}(\mathcal{L})$.

Every team $T \subseteq M^V$ can be encoded by an $\mathsf{FO}(\tau \cup M)$-formula $\varphi_T$ that uses only the variables in $V$. Indeed, although rather inefficient, we can just set

$$\varphi_T(\mathbf{v}) = \bigvee_{t \in T} (\mathbf{v} = t(\mathbf{v})).$$

This shows that it is very lenient to assume that there exists some global bound for $r$ in all instances of $\mathrm{MC}^k_{\mathrm{formula}}(\mathcal{L})$.

**Definition 5.19.** For $k \in \mathbb{N}$ and $B \geq k$, define $\mathrm{MC}^{B,k}_{\mathrm{formula}}(\mathcal{L})$ as the restriction of $\mathrm{MC}_{\mathrm{formula}}(\mathcal{L})$ to instances $(\psi, \mathfrak{M}, \mathbf{v}, s, \varphi_T)$ where:

1. $|V| \leq k$, and
2. every subformula of $\varphi_T$ has at most $B$ free variables.

**Corollary 5.20.** $\mathrm{MC}^{B,k}_{\mathrm{formula}}(\mathcal{L}) \in \textsc{Alogspace} = \textsc{Ptime}$ for all $B \geq k \in \mathbb{N}$.

*Proof.* From Lemma 5.17 and the above definition of $\mathrm{MC}^{B,k}_{\mathrm{formula}}(\mathcal{L})$ we see that we can solve instances $(\psi, \mathfrak{M}, \mathbf{v}, s, \varphi_T)$ of $\mathrm{MC}^{B,k}_{\mathrm{formula}}(\mathcal{L})$ with the algorithm from Lemma 5.17 requiring only alternating space $\mathcal{O}(\log|\psi| + \log|\mathfrak{M}| + \log|\varphi_T|)$. □

**Proposition 5.21.**

1. $\mathrm{MC}_{\mathrm{formula}}(\mathcal{L})$ is $\textsc{Pspace}$-complete.
2. $\mathrm{MC}^{B,k}_{\mathrm{formula}}(\mathcal{L})$ is $\textsc{Ptime}$-complete for all $B \geq k \geq 2$.

*Proof.* One part is already known from Corollaries 5.18 and 5.20. Since the full team is specified by $\varphi_{M^V}(\mathbf{v}) = \mathrm{True}$, we obtain the following logspace-computable reduction showing that $\mathrm{MC}_{\mathrm{full}}(\mathcal{L}) \leq_{\log} \mathrm{MC}_{\mathrm{formula}}(\mathcal{L})$:

$$\mathrm{MC}_{\mathrm{full}}(\mathcal{L}) \to \mathrm{MC}_{\mathrm{formula}}(\mathcal{L}), \quad (\psi, \mathfrak{M}, \mathbf{v}, s) \mapsto (\psi, \mathfrak{M}, \mathbf{v}, s, \mathrm{True}).$$

Via the same reduction we can show that $\mathrm{MC}^{k}_{\mathrm{full}}(\mathcal{L}) \leq_{\log} \mathrm{MC}^{B,k}_{\mathrm{formula}}(\mathcal{L})$ for all $B \geq k \in \mathbb{N}$. The claim then follows from Propositions 5.8 and 5.9. □

This shows that under reasonable assumptions, the complexity of model checking $\mathcal{L}$ is analogous to the complexity of first-order model checking; PSPACE-complete in general, but PTIME-complete in restriction to $k$ variables, for $k \geq 2$.

# Chapter 6

# Remarks on the Finite Model Property

Although deciding whether LFD has the finite model property (FMP) has certainly been the main focus of the author on numerous occasions over the period of writing this thesis, an answer was unfortunately not found. In this chapter we wish to mention various interesting observations and the approaches that have been tried to tackle this problem.

For the rest of this chapter we fix some finite type $(\tau, V)$. A dependence model is finite if it has a finite universe. A logic has the FMP if every satisfiable sentence in it has some finite model. Since $V$ is finite, a finite universe always entails a finite team. The converse holds up to LFD-equivalence:

**Remark 6.1.** Let $\mathbf{M}, s \in \mathcal{DEP}[\tau, V]$ be a dependence model with a finite team $T$. Considering the definitions of the translations to and from standard relational models (cf. Definition 4.10), we see that the universe of

$$\mathbf{N}, t \coloneqq \mathsf{T}_{\mathrm{srm} \to \mathrm{dep}}(\mathsf{T}_{\mathrm{dep} \to \mathrm{srm}}(\mathbf{M}, s))$$

consists of equivalence classes of assignments in $T$, indexed by variables from $V$. Therefore it must be finite. Moreover, we know from Fact 4.11 that $\mathbf{N}, t \equiv_{\mathsf{LFD}} \mathbf{M}, s$, which shows that only the objects which appear in some assignment of the team are relevant for evaluating LFD-formulae on dependence models. In particular, $\mathbf{M}$ is LFD-equivalent to a finite dependence model, hence in the context of LFD we can consider dependence models with finite teams as finite.

We want to mention that [6, Section 6] defined a more general version of the standard relational models, now just called relational models, for which a finite model property was shown via filtration in [6, Proposition 6.5]. However, this does not imply a finite model property for $\mathsf{LFD}$ on $\mathcal{SRM}$ or $\mathcal{DEP}$.

**Proposition 6.2.** The following are equivalent.

1. $\mathsf{LFD}$ has the finite model property.
2. All characteristic formulae $\chi^2_{\mathbf{M},s}$ of the 2-$\mathsf{LFD}$-bisimulation classes of dependence models have a finite model.
3. Every dependence model is 2-$\mathsf{LFD}$-bisimilar to a finite dependence model.

*Proof.* Based on Lemma 3.10, the equivalence of (2) and (3) is obvious. As each $\chi^2_{\mathbf{M},s}$ is satisfiable, the implication "(1) $\Longrightarrow$ (2)" is clear as well.

Now assume (2) and let $\psi \in \mathsf{LFD}$ be satisfiable. Using our Scott normal form from Proposition 4.1, we see that there exists some $\varphi \in \mathsf{LFD}$ with $\mathsf{qr}(\varphi) \leq 2$ so that $\psi$ and $\varphi$ are satisfiable over the same universes. In particular $\varphi$ has some model $\mathbf{M}, s \models \varphi$. Per assumption there exists a finite dependence model $\mathbf{N}, t$ with $\mathbf{N}, t \models \chi^2_{\mathbf{M},s}$. Then Lemma 3.10 yields

$$\mathbf{N}, t \sim^2_{\mathsf{LFD}} \mathbf{M}, s \qquad \text{and thus} \qquad \mathbf{N}, t \equiv^2_{\mathsf{LFD}} \mathbf{M}, s$$

by our Ehrenfeucht-Fraïssé analogue Theorem 3.12. Because of $\mathsf{qr}(\varphi) \leq 2$ we have $\mathbf{N}, t \models \varphi$. Furthermore, since $\varphi \models \psi$ (cf. Proposition 4.1), we can infer that $\mathbf{N}, t \models \psi$, i.e. that $\psi$ has a finite model. This proves the last implication "(2) $\Longrightarrow$ (1)". □

We remind the reader of the discussion below Definition 3.7 where we mentioned that unlike for propositional modal logic, the quantifier rank does not capture the notion of *how far we look into the model*, because of the global modality $\mathsf{A}$. In the same vein, remember that all $\mathsf{LFD}$-bisimulations are global (except $\sim^0$). Hence we cannot simply "cut off" models $\mathbf{M}, s$ after some notion of distance around $s$ to obtain a finite 2-bisimilar model, as one would do for $\mathsf{ML}$.

Another way to transform the problem is to use certain finiteness-preserving good first-order translations and try to reason about the FMP of the resulting fragment of $\mathsf{FO}$ over the relevant class of structures.

**Fact 6.3.** Let $(\mathcal{C}[\sigma], \mathsf{F}, \mathsf{G}, \mathsf{tr}, (\vartheta_X)_{X \subseteq V})$ be a good translation of $\mathsf{LFD}$ over $\mathcal{DEP}[\tau, V]$ into $\mathsf{FO}$ over $\mathcal{C}[\sigma]$ such that:

1. If $\mathbf{M}, s \in \mathcal{DEP}[\tau, V]$ is finite, then $\mathsf{F}(\mathbf{M}, s)$ is finite as well.
2. If $\mathfrak{A}, \mathbf{a} \in \mathcal{C}[\sigma]$ is finite, then $\mathsf{G}(\mathfrak{A}, \mathbf{a})$ is finite as well.

Then $\mathsf{LFD}(\tau, V)$ has the FMP if and only if $\mathsf{FO}(\sigma)/\!\sim$ has it over $\mathcal{C}[\sigma]$, meaning that for every satisfiable $\varphi \in \mathsf{FO}(\sigma)$ which is equivalent to a translated $\mathsf{LFD}$-formula over $\mathcal{C}[\sigma]$ (cf. Theorem 4.40), there exists some finite $\mathfrak{A}, \mathbf{a} \in \mathcal{C}[\sigma]$ with $\mathfrak{A}, \mathbf{a} \models \varphi$.

Note that the standard, modal, and functional translations we discussed in Sections 4.1.1 to 4.1.3 preserve the finiteness of models as described above. One of the main motivations for the functional translation $\mathsf{tr}_{\text{fun}}$ was that it comes quite close to embedding $\mathsf{LFD}$ into a fragment of the monadic class $[all, (\omega), (\omega)]$, which consists of all first-order formulae that only contain monadic predicates and unary functions but no equality atoms. This class, also called the Löb-Gurevich class, is well known to have the finite model property (see e.g. [12, Chapter 6.2.1]), but becomes undecidable when we allow equality, similar to $\mathsf{LFD}$ and $\mathsf{LFD}^{=}$. Formulae translated by $\mathsf{tr}_{\text{fun}}$ do indeed only use monadic predicates and unary functions, but they also require equality atoms of the form $f_x(s) = f_x(t)$ to encode agreement on variables for assignments $s, t$. Therefore, $\mathsf{tr}_{\text{fun}}$ does not embed $\mathsf{LFD}$ into the monadic class. To the author's knowledge, none of the three discussed example translations embed $\mathsf{LFD}$ into a fragment of $\mathsf{FO}$ that is known to have the FMP over the relevant class of structures.

Attacking the problem from the other direction, we also tried to use the above fact to find an infinity axiom for $\mathsf{LFD}$ via the modal translation $\mathsf{tr}_{\text{mod}}$, by adapting existing work about $\mathsf{FO}^2$ and $\mathsf{GF}^2$ on equivalence structures, but unfortunately came to no conclusion either, as explained below.

**Remark 6.4.** [29, Example 5.1.1] describes an infinity axiom of $\mathsf{FO}^2$ with two equivalence relations $E_1, E_2$ and three unary relations $P, Q, S$ by the following conditions:

1. $P$ and $Q$ are disjoint and $S$ and $Q$ are disjoint and not related by $E_2$-links.
2. Every element of $P$ is $E_1$-equivalent to one in $Q$; every element of $Q$ is $E_2$-equivalent to one in $P$.
3. $S \cap P \neq \varnothing$.
4. Every pair of elements both of which belong to $P$ or to $Q$ is either connected by both $E_1$ and $E_2$ or by neither.

We tried to adapt this to $\mathsf{LFD}$ over four variables $x, y, z, v$, using $=_x$ for $E_1$ and $=_y$ for $E_2$, and evaluating the predicates $P, Q$ only on $z$, and $S$ only on $v$ (essentially, we take the modal perspective, viewing assignments as our elements in $S, P, Q$). This is straightforward for the first three conditions:

1. $\psi_1 := \mathsf{A}(\neg(Pz \wedge Qz) \wedge \neg(Sv \wedge Qz) \wedge (Sv \to \mathsf{D}_y(\neg Qz)) \wedge (Qz \to \mathsf{D}_y(\neg Sv)))$.

2. $\psi_2 := \mathsf{A}((Pz \to \mathsf{E}_x\, Qz) \wedge (Qz \to \mathsf{E}_y\, Pz))$.

3. $\psi_3 := \mathsf{E}(Sv \wedge Pz)$.

The problem lies in the fourth condition, which in our case translates to the requirement that up to $=_{xy}$, every $=_x$-class and $=_y$-class contains at most one element in $P$ and one in $Q$. As a consequence of Corollary 3.18 and Fact 3.19 it seems highly unlikely that this condition can be expressed in plain $\mathsf{LFD}$.

The role of Corollary 3.18 is crucial; we mentioned below it that if it would not hold, so for example we could define some $\varphi_2 \in \mathsf{LFD}$ which reasonably enforces $|T_{\mathbf{M}}(z)| \leq 2$, then we could already define an infinity axiom in $\mathsf{LFD}$. This is formalized in the following proposition.

**Proposition 6.5.** Assume we have a formula $\varphi_2$ such that for all dependence models $\mathbf{M}, s$ with $\mathbf{M}, s \models \varphi_2$ we have $|T_{\mathbf{M}}(z)| \leq 2$. Define

$$\psi_4 := \varphi_2 \wedge \mathsf{E}(Pz \wedge \neg Qz) \wedge \mathsf{E}(\neg Pz \wedge Qz) \wedge \mathsf{A}(D_{xz}y \wedge D_{yz}x). \tag{6.1}$$

Then let $\psi := \bigwedge_{i=1}^{4} \psi_i$ where the $\psi_1, \psi_2, \psi_3$ were defined in Remark 6.4. If $\psi$ is satisfiable, then it is an infinity axiom.

*Proof.* Let $\mathbf{M}, s \models \psi$ and assume for the sake of contradiction that $\mathbf{M}$ is finite. Via $\psi_3$ there is some $t_0 \in T_{\mathbf{M}}$ with $\mathbf{M}, t_0 \models Sv \wedge Pz$. Using $\psi_2$, we inductively define $(t_k)_{k\in\mathbb{N}}$ with $t_{2k} =_x t_{2k+1} =_y t_{2k+2}$ and $\mathbf{M}, t_{2k} \models Pz$ and $\mathbf{M}, t_{2k+1} \models Qz$ for all $k \in \mathbb{N}$. Since there is exactly one value of $z$ in $P$ and exactly one in $Q$, we have $t_{2k} =_z t_{2\ell}$ and $t_{2k+1} =_z t_{2\ell+1}$ for all $k, \ell \in \mathbb{N}$. Therefore, using $\mathbf{M} \models \mathsf{A}\, D_{xz}y$ and $\mathbf{M} \models \mathsf{A}\, D_{yz}x$ from $\psi_4$, we obtain that $t_{2k} =_x t_{2\ell}$ or $t_{2k} =_y t_{2\ell}$ already implies $t_{2k} =_{xyz} t_{2\ell}$, and likewise for $t_{2k+1}$ and $t_{2\ell+1}$, for all $k, \ell \in \mathbb{N}$.

Because $\mathbf{M}$ is finite, there must exist some smallest $k \in \mathbb{N}$ such that there is some $n > k$ with $t_k =_{xyz} t_n$. If $k > 0$ is even, then $n$ must be even too, and $t_{k-1} =_y t_k =_{xyz} t_n =_y t_{n-1}$ implies $t_{k-1} =_{xyz} t_{n-1}$. But then $t_{k-2} =_x t_{k-1} =_{xyz} t_{n-1} =_x t_{n-2}$ and hence $t_{k-2} =_{xyz} t_{n-2}$. Via induction we arrive at $t_0 =_{xyz} t_{n-k} =_y t_{n-k-1}$. Since $n-k-1$ is odd, we have $\mathbf{M}, t_{n-k-1} \models Qz$. Together with $\mathbf{M}, t_0 \models Sv$ and $t_0 =_y t_{n-k-1}$ we obtain our contradiction to $\psi_1$, since $S$ and $Q$ are now related by $=_y$. The case for $k = 0$ is included in the above argumentation, and the case for odd $k > 0$ is shown analogously. □

Although we already know that $\mathsf{LFD}^=$ must contain infinity axioms because of its undecidability, we can use the above proposition to give an explicit one.

**Example 6.6.** Using the $\mathsf{LFD}^=$-sentence

$$\varphi_2 := D_\varnothing z_1 \wedge D_\varnothing z_2 \wedge \mathsf{A}(z = z_1 \vee z = z_2)$$

in Proposition 6.5, we obtain an explicit infinity axiom for $\mathsf{LFD}^=$.

*Proof.* Clearly $\varphi_2$ enforces $|T_\mathbf{M}(z)| \leq 2$. We only need to show that $\psi$ as defined in Proposition 6.5 with the above $\varphi_2$ is satisfiable. Define $\mathbf{M}$ as having

1. The universe $\mathbb{N}$.
2. The relations $P^\mathbf{M} := \{0\}$, $Q^\mathbf{M} := \{1\}$, and $S^\mathbf{M} := \{0\}$.
3. The following team, with the fixed order of variables $x, y, z, v$:

   $$T_\mathbf{M} := \{(0,0,0,0)\} \cup \{(k,k,0,1) \mid k \geq 1\} \cup \{(k,k+1,1,1) \mid k \geq 0\}.$$

   The auxiliary variables $z_1, z_2$ used in $\varphi_2$ were left out here for ease of presentation, as they are constant anyways. Specifically, let $T_\mathbf{M}(z_1) = \{0\}$ and $T_\mathbf{M}(z_2) = \{1\}$.

Now it is trivial to verify that $\mathbf{M} \models \psi$. □

We can also adapt this infinity axiom to another extension of $\mathsf{LFD}$. In the subsection **Learning new facts** of [6, Section 7.4] a dynamic extension of $\mathsf{LFD}$ was introduced. It allows restricting the team to those assignments satisfying some formula $\varphi$ via an update modality $[\varphi]$:

$$\mathbf{M}, s \models [\varphi]\psi \qquad \text{iff} \qquad \mathbf{M}, s \models \varphi \text{ implies } \mathbf{M}|\varphi, s \models \psi$$

where $\mathbf{M}|\varphi$ differs from $\mathbf{M}$ only on its team $T_{\mathbf{M}|\varphi} = \{t \in T_\mathbf{M} \mid \mathbf{M}, t \models \varphi\}$. This follows the common theme of viewing the process of gaining knowledge as elimination of possibilities. Logics with such update modalities are sometimes also referred to as "public announcement logics".

It was remarked that $\mathsf{LFD}$ with this update modality can be reduced to $\mathsf{LFD}$ with new conditional dependence atoms $D_X^\varphi y$ that have the semantics

$$\mathbf{M}, s \models D_X^\varphi y \qquad \text{iff} \qquad \forall t \in T_\mathbf{M}(\mathbf{M}, t \models \varphi \text{ and } t =_X s \text{ imply } t =_y s).$$

[6, Fact 7.14] proves that such conditional dependencies cannot be expressed within plain $\mathsf{LFD}$. Since no explicit name was given for this logic, we shall call it $\mathsf{LCFD}$ for the "Logic of Conditional Functional Dependence".

**Definition 6.7.** Let $(\tau, V)$ be a type. Then $\mathsf{LCFD}(\tau, V)$ is the collection of formulae obtained by the following rules:

1. $\mathsf{LFD}(\tau, V) \subseteq \mathsf{LCFD}(\tau, V)$.
2. $\mathsf{LCFD}(\tau, V)$ is closed under boolean connectives $\wedge, \neg$ and dependence quantifiers $\mathsf{D}_X$ for finite $X \subseteq V$.
3. For $\varphi \in \mathsf{LCFD}(\tau, V)$, finite $X \subseteq V$ and $y \in V$, let $D_X^{\varphi} y \in \mathsf{LCFD}(\tau, V)$.

Moreover, denote by $\mathsf{LCFD}^{-}$ the fragment of $\mathsf{LCFD}$ which only allows conditional dependence atoms $D_X^{\varphi} y$ where $\varphi \in \mathsf{LFD}$.

To the author's knowledge, it is still an open question whether $\mathsf{LCFD}$ or even $\mathsf{LCFD}^{-}$ is decidable at the time of writing this thesis. We show below that the conditional dependence atoms allow us to adapt the infinity axiom we discussed above.

**Proposition 6.8.** $\mathsf{LCFD}^{-}$ (and hence $\mathsf{LCFD}$) does not have the FMP.

*Proof.* We proceed analogously to Example 6.6 and use Proposition 6.5. This time, define

$$\varphi_2 := D_{\varnothing}^{Pz} z \wedge D_{\varnothing}^{\neg Pz} z.$$

Define $\psi$ as in Proposition 6.5 with the $\varphi_2$ from here. Let $\mathbf{M}$ be the same model as in Example 6.6, without the auxiliary variables $z_1, z_2$ used there. It remains to show that $\mathbf{M} \models \varphi_2$. Let $t \in T_{\mathbf{M}}$. If $\mathbf{M}, t \models Pz$, then $t = (0, 0, 0, 0)$ or $t \in \{(k, k, 0, 1) \mid k \geq 1\}$. Either way we have $t(z) = 0$. This shows $\mathbf{M} \models D_{\varnothing}^{Pz} z$. Analogously, if $\mathbf{M}, t \models \neg Pz$, then $t \in \{(k, k + 1, 1, 1) \mid k \geq 0\}$, and $t(z) = 1$. This shows $\mathbf{M} \models D_{\varnothing}^{\neg Pz} z$, so overall $\mathbf{M} \models \varphi_2$. □

# Chapter 7

# Conclusion and Further Research

Let us recall what we have shown in this thesis. In Section 2.3 we proved the rather surprising fact that $\mathsf{LFD}^=$ is a conservative reduction class, negatively answering the question about its decidability that was posed in [6, Section 7.2].

We showed how to define a notion of bisimulation for both $\mathsf{LFD}$ and $\mathsf{LFD}^=$, and related it to equivalence within (the infinitary versions of) $\mathsf{LFD}$ and $\mathsf{LFD}^=$ via our Ehrenfeucht-Fraïssé analogue Theorem 3.12. We used said theorem to give various examples that demonstrate the limits of expressiveness of both these logics, in particular showcasing $\mathsf{LFD}$'s inability to deal with quantities.

We formally defined what constitutes a reasonable first-order translation of $\mathsf{LFD}$ and gave three examples in Sections 4.1.1 to 4.1.4, illustrating the various perspectives of $\mathsf{LFD}$. Solidifying the modal character, we proved an analogue of van Benthem's Theorem in Theorem 4.40, showing that $\mathsf{LFD}$ corresponds to the $\mathsf{LFD}$-bisimulation-invariant fragment of $\mathsf{FO}$ under all good first-order translations. In this spirit of relating $\mathsf{LFD}$ to other logics, we continued in Section 4.2 by proving that $\mathsf{LFD}$ is expressively incomparable to the (clique) guarded fragment in the context of the modal and standard translation, suggesting that this not caused by the specific first-order translation of $\mathsf{LFD}$, but rather an intrinsic fact about the disparities between $\mathsf{LFD}$ and guarded fragments of $\mathsf{FO}$. Furthermore, we show how to relate $\mathsf{LFD}$ to logics with team semantics by embedding it into inclusion-exclusion logic (i.e. independence logic) in Section 4.3.

In Section 5.1 the complexity of the satisfiability problem of $\mathsf{LFD}$ was narrowed town to lie between PSPACE and 2-NEXPTIME. The complexity of the model checking problem for $\mathsf{LFD}$ and $\mathsf{LFD}^=$ is shown to to largely depend on the way we encode teams. Using a compact encoding via first-order formulae, we showed the complexity to be analogous to that of model checking for $\mathsf{FO}$, namely PSPACE-complete, but PTIME-complete in restriction to $k$ variables.

We provided techniques and simplifications that may help to settle the question of whether LFD has the finite model property. We show how the ability to define upper bounds on the number of values some variable leads to infinity axioms for logics of the LFD-family. Using our examples from earlier, we demonstrate exactly why this is impossible in LFD, but also show how it works for $\mathsf{LFD}^=$ and the extension LCFD with conditional functional dependencies.

Although technically not part of this thesis, we also want to mention the python library written by the author[1], which was used to find and minimize most of the examples in this thesis, and may be helpful to others.

Apart from the open questions from [6] that we did not investigate, the following further questions emerged, some of them in discussion with Alexandru Baltag, Johan van Benthem, and Erich Grädel:

1. Does LFD have the finite model property (FMP)?
2. What is the exact complexity of the satisfiability problem for LFD?
3. Related to (2.), is there a version of tiling arguments that is possible for LFD and related logics? (See Remark 5.3).
4. What is the complexity of testing whether two given pointed dependence models of the same type are bisimilar?
5. What can be said about the extension of LFD by inclusion-atoms $x \subseteq y$ with the semantics
$$\mathbf{M}, s \models x \subseteq y \qquad \text{iff} \qquad T_{\mathbf{M}}(x) \subseteq T_{\mathbf{M}}(y).$$
Can the proof of undecidability for $\mathsf{LFD}^=$ be adapted?
6. Is $\mathsf{LCFD}^-$ or even LCFD decidable, despite not having the FMP?

[1]https://git.rwth-aachen.de/philpuetzstueck/lfd-sat.

# Appendix A

# Undefinability

The following and most other examples were found using a python library written by the author.[1] We always present the smallest possible counterexample, in the sense that we first minimized the number of entries in the bisimulation, then the size of the teams, and finally the number of elements in the structures.

**Example A.1.** We give an example to show that $\mathsf{LFD}^=$ cannot define cartesian products within binary relations. More specifically; we prove that there cannot exist a formula $\psi \in \mathsf{LFD}^=$ over a relational vocabulary $\tau \uplus \{R\}$, such that for every $(\tau \uplus \{R\}, V)$ dependence model $\mathbf{M}$ and $s \in T_\mathbf{M}$ we have

$$\mathbf{M}, s \models \psi \qquad \text{iff} \qquad \{(a,b) \mid \exists \mathbf{c}\colon (a,b,\mathbf{c}) \in R^\mathbf{M}\} \text{ is a cartesian product.}$$

It suffices to show this for $\tau = \varnothing$, $R$ binary and $V = \{x, y\}$, because the example below can be adapted accordingly, e.g. by setting $K^\mathbf{M} = K^\mathbf{N} = \varnothing$ for all other relations $K \in \tau$, letting variables apart from $x, y$ be constant in both teams while ensuring agreement on equality atoms, and letting all positions except the first two of $R$ be constant within both models. Consider

$$\mathbf{M} = (\{a,b\}, R^\mathbf{M}, T_\mathbf{M}) \qquad \text{and} \qquad \mathbf{N} = (\{0,1,2\}, R^\mathbf{N}, T_\mathbf{N})$$

where relations and teams as well as $Z \subseteq T_\mathbf{M} \times T_\mathbf{N}$ are given by

| $R^\mathbf{M}$ | $R^\mathbf{N}$ | $T_\mathbf{M}$ | $T_\mathbf{N}$ | $Z$ | |
|---|---|---|---|---|---|
| $(b,b)$ | $(0,0)$ | $(a,a)$ | $(0,0)$ | $(a,a)$ | $(1,1)$ |
| | $(2,2)$ | $(b,b)$ | $(1,1)$ | $(b,b)$ | $(0,0)$ |
| | | | $(2,2)$ | $(b,b)$ | $(2,2)$ |

[1] https://git.rwth-aachen.de/philpuetzstueck/lfd-sat.

Then it is trivial to verify that $Z$ is an $\mathsf{LFD}^=$-bisimulation. Moreover

$$\{b\} \times \{b\} = R^{\mathbf{M}} \subseteq T_{\mathbf{M}}$$

whereas for $\mathbf{N}$ we have

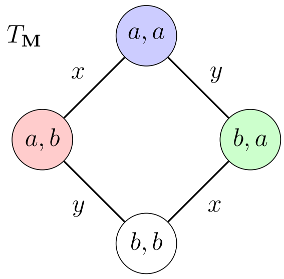
$$(0,0),(2,2) \in R^{\mathbf{N}} \qquad \text{and} \qquad R^{\mathbf{N}} \subseteq T_{\mathbf{N}} \qquad \text{but} \qquad (0,2) \notin R^{\mathbf{N}}.$$

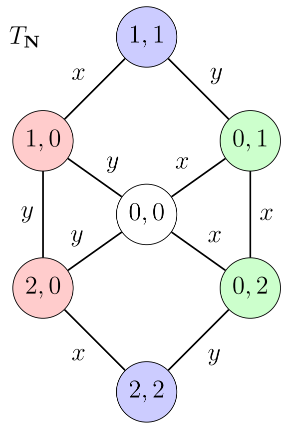
Hence $R^{\mathbf{M}}$ is a cartesian product while $R^{\mathbf{N}}$ is not, but $\mathbf{M}, s \equiv^{\infty}_{\mathsf{LFD}^=} \mathbf{N}, t$.

**Example A.2.** We give an example to show that $\mathsf{LFD}^=$ cannot define cartesian products within the assignment space. More specifically; we prove that there cannot exist a formula $\psi \in \mathsf{LFD}^=$ such that for every pointed dependence model $\mathbf{M}, s$ we have $\mathbf{M}, s \models \psi$ iff $T_{\mathbf{M}}(x, y)$ is a cartesian product. Similarly to Example A.1 above, we can restrict ourselves to an empty vocabulary $\tau = \varnothing$ and $V = \{x, y\}$, since the example can be adapted accordingly. Consider

$$\mathbf{M} = (\{a, b\}, T_{\mathbf{M}}) \qquad \text{and} \qquad \mathbf{N} = (\{0, 1, 2\}, T_{\mathbf{N}})$$

where $\mathbf{M}$ is full, so $T_{\mathbf{M}} = \{a, b\}^2$ and $T_{\mathbf{N}} = \{0, 1, 2\}^2 \setminus \{(1, 2), (2, 1)\}$. Below we depict $T_{\mathbf{M}}$ and $T_{\mathbf{N}}$. The colors represent a bisimulation $Z \subseteq T_{\mathbf{M}} \times T_{\mathbf{N}}$ (e.g. since $(a, a), (1, 1), (2, 2)$ are blue, we have $(a, a)Z(1, 1)$ and $(a, a)Z(2, 2)$) and the labels of the edges denote on which variables the conjoined assignments agree, i.e. they represent the relations $=_x$ and $=_y$.




It is straightforward to verify that $Z$ is indeed a bisimulation between $\mathbf{M}$ and $\mathbf{N}$. Moreover, we can see that $T_{\mathbf{M}}$ is a cartesian product, while $T_{\mathbf{N}}$ is not, even though there is a (necessarily global) $\mathsf{LFD}^=$-bisimulation between $\mathbf{M}$ and $\mathbf{N}$, and hence $\mathbf{M}, s \equiv^{\infty}_{\mathsf{LFD}^=} \mathbf{N}, t$ for all $(s, t) \in Z$ by Theorem 3.12.

**Proposition A.3.** Consider some distinguished dependence model $\mathbf{M}, s$ and a copy $\mathbf{M}'$ of $\mathbf{M}$ such that $\mathbf{M}$ and $\mathbf{M}'$ are disjoint except for values which are taken by a variable that is constant throughout the whole team. Then $(\mathbf{M} \cup \mathbf{M}'), s$ is still distinguished, and $\mathbf{M}, s \sim_{\mathsf{LFD}} (\mathbf{M} \cup \mathbf{M}'), s$.

*Proof.* So let $\mathbf{M}, s = (\mathfrak{M}, T), s$ be a distinguished $(\tau, V)$ dependence model with universe $M$. Denote by $C \subseteq V$ the set of variables that are constant in $T$, and define for $m \in M$

$$f(m) \coloneqq \begin{cases} m, & m \text{ is the value of some } x \in C \\ m', & \text{otherwise} \end{cases}$$

where the $m'$ are fresh copies of $m$, not occurring in $M$. To construct $\mathbf{M}'$, we set its universe to $M' \coloneqq f(M)$ and use the team $T' \coloneqq \{f(t) \mid t \in T\}$ where $f(t)$ represents the assignment with $f(t)(x) = f(t(x))$ for all $x \in V$. Finally we let

$$R^{\mathbf{M}'} \coloneqq \{(f(m_1), \ldots, f(m_n)) \mid (m_1, \ldots, m_n) \in R^{\mathbf{M}}\}.$$

Now $\mathbf{M}'$ is the copy of $\mathbf{M}$ that is disjoint to $\mathbf{M}$ except for the constants. Write $\mathbf{M}^+$ for the union of $\mathbf{M}$ and $\mathbf{M}'$, having objects $M^+ \coloneqq M \cup M'$, the team $T^+ \coloneqq T \cup T'$ and relations $R^{\mathbf{M}^+} \coloneqq R^{\mathbf{M}} \cup R^{\mathbf{M}'}$. Obviously $\mathbf{M}^+$ is still distinguished.

Note that since $f\colon M \to M'$ is bijective, we have for $t, u \in T$ that

$$t =_X u \qquad \text{iff} \qquad f(t) =_X f(u), \qquad\qquad X \subseteq V.$$

Moreover, $t =_X f(u)$ already implies that $X \subseteq C$. Indeed, per definition of $f$ we see that that $f(u)(x) = f(u(x)) \in M$ iff $u(x)$ is the value of some constant variable, but since $\mathbf{M}$ is distinguished, that variable must be $x$ itself. Note that the converse holds as well, i.e. $t, u, f(t), f(u)$ agree on $C$, for all $t, u \in T$. Now define the relation

$$Z = \{(t, t) \mid t \in T\} \cup \{(t, f(t)) \mid t \in T\} \subseteq T \times T^+.$$

We verify that $Z$ is an $\mathsf{LFD}$-bisimulation between $\mathbf{M}, s$ and $\mathbf{M}^+, s$. Clearly $(s, s) \in Z$ and pairs in $Z$ agree on relational atoms.

Consider a pair $(t, t) \in Z$. We assume $\mathbf{M}, t \models D_X y$ and show $\mathbf{M}^+, t \models D_X y$. Let $u \in T^+ = T \cup T'$ with $u =_X t$. If $u \in T$, then the assumption immediately yields $u =_y t$, and we are done. If $u = f(q) \in T' \setminus T$ for some $q \in T$, then as explained above, $f(q) = u =_X t$ implies $X \subseteq C$. But then $\mathbf{M}, t \models D_X y$ means that $y \in C$ as well, and hence $u =_y t$. We conclude $\mathbf{M}^+, t \models D_X y$. The converse implication "$\mathbf{M}^+, t \models D_X y$ implies $\mathbf{M}, t \models D_X y$" is clear since $T \subseteq T^+$.

Now consider a pair $(t, f(t)) \in Z$ instead. Again we assume $\mathbf{M}, t \models D_X y$ and show $\mathbf{M}^+, f(t) \models D_X y$. Let $u \in T^+ = T \cup T'$ with $u =_X f(t)$. If $u = f(q) \in T' \setminus T$ for some $q \in T$, then $f(q) =_X f(t)$, so $q =_X t$ and hence $q =_y t$. But then

$u = f(q) =_y f(t)$, and we are done. If $u \in T$, then $u =_X f(t)$ implies $X \subseteq C$, so $y \in C$ and hence $u =_y f(t)$, as before. Overall we conclude $\mathbf{M}^+, f(t) \models D_X y$. For the converse implication, assume $\mathbf{M}^+, f(t) \models D_X y$ and let $u \in T$ with $u =_X t$. Then $f(u) =_X f(t)$, so $f(u) =_y f(t)$, and thereby $u =_y t$, which shows $\mathbf{M}, t \models D_X y$.

Hence pairs in $Z$ agree on LFD-atoms. Now let $t \in T$. We verify the back and forth conditions as defined in Definition 3.1, although it actually looks more like bisimulation for finite types as defined in Fact 3.2, because each assignment in $T^+$ only appears in a single pair in $Z$.

1. Forth condition at $(t, t)$: For some $u \in T$ we choose $u \in T^+$ with $(u, u) \in Z$, regardless of the set $X \subseteq u \overline{\sqcap} t$.
2. Back condition at $(t, t)$: Let $u \in T^+ = T \cup T'$.
   (a) If $u \in T$ we again correspond with $u$ to $(u, u) \in Z$.
   (b) If $u = f(q) \in T' \setminus T$, we choose $q \in T$ to arrive at $(q, f(q)) \in Z$. Since $f(q) \overline{\sqcap} t = C$ and clearly $q =_C t$, this always works out.
3. Forth condition at $(t, f(t))$: For some $u \in T$ we can choose $f(u) \in T'$ with $(u, f(u)) \in Z$, since $(u \overline{\sqcap} t) = (f(u) \overline{\sqcap} f(t))$.
4. Back condition at $(t, f(t))$: Let $u \in T^+ = T \cup T'$.
   (a) If $u = f(q) \in T' \setminus T$, then we again correspond with $q \in T$ to land at $(q, f(q)) \in Z$.
   (b) If $u \in T$, then we can choose $u \in T$ to arrive at $(u, u) \in Z$, since $u \overline{\sqcap} f(t) = C$ and clearly $u =_C t$.

We conclude that $Z$ is an LFD-bisimulation and hence $\mathbf{M}, s \sim_{\mathsf{LFD}} \mathbf{M}^+, s$. □

**Example A.4.** We give an example to show that $\mathsf{LFD}^=$ cannot define any non-trivial upper bounds on the number of assignments in some $=_X$-class that satisfy some relational atom. More specifically, given:

1. some $n \geq 1$,
2. a relational vocabulary $\{R\} \cup \tau$,
3. a set of variables $V \neq \varnothing$ with a finite proper subset $X \subsetneq V$,
4. and a tuple $\mathbf{x} \in V^{\mathrm{ar}(R)}$,

there exist two $\mathsf{LFD}^{=}$-bisimilar pointed $(\{R\} \cup \tau, V)$ dependence models $\mathbf{M}, s$ and $\mathbf{N}, t$ such that the $=_X$-class of $s$ in $T_{\mathbf{M}}$ contains at most $n$ assignments $s'$ for which $\mathbf{M}, s' \models R\mathbf{x}$, whereas the $=_X$-class of $t$ in $T_{\mathbf{N}}$ contains more than $n$ such assignments. It suffices to show this for the case where $\tau = \varnothing, V = \{x\}, X = \varnothing, \mathbf{x} = x$ and $R$ is a monadic predicate. Indeed, as described in Example A.1 we can set $K^{\mathbf{M}} = K^{\mathbf{N}} = \varnothing$ for all other $K \in \tau$ to ensure bisimilar assignments from below will agree on all new relational atoms. Moreover, if we want larger $V$ and $X$, then we simply choose some $x \in V \setminus X$ and make all other variables constant, with two distinct variables sharing the same constant in one team iff the same holds in the other, thus guaranteeing that all bisimilar assignments from below will agree on all new equality atoms. It is easy to see that we can also adapt $R$ to have a higher arity by extending the contained tuples with enough constant dummy values according to $\mathbf{x}$, which must contain the non-constant variable $x$.

For the special case $\tau = \varnothing, V = \{x\}, X = \varnothing, \mathbf{x} = x$ and $R$ monadic, consider

$$\mathbf{M} = (\{a, b\}, R^{\mathbf{M}}, T_{\mathbf{M}}) \qquad \text{and} \qquad \mathbf{N} = (\{0, \ldots, n+1\}, R^{\mathbf{N}}, T_{\mathbf{N}})$$

where $R^{\mathbf{M}} = \{b\}$, $R^{\mathbf{N}} = \{1, \ldots, n+1\}$ and the teams are full, so $T_{\mathbf{M}} = \{a, b\}$ and $T_{\mathbf{N}} = \{0, \ldots, n+1\}$.[2] Now the bisimulation is obvious:

$$Z \coloneqq \{(a, 0)\} \cup \{(b, k) \mid k \in \{1, \ldots, n+1\}\}.$$

It is clear that this is indeed an $\mathsf{LFD}^{=}$-bisimulation; the related assignments agree on $Rx$ and do not agree on any variables with any other assignment in their respective teams. Since in this special case we have $X = \varnothing$, the claim now follows by considering an arbitrary $(s, t) \in Z$ and noting that

$$\begin{array}{lll} \mathbf{M}, s \sim_{\mathsf{LFD}^{=}} \mathbf{N}, t & \text{and} & |\{s' \in T_{\mathbf{M}} \mid s' =_X s,\ \mathbf{M}, s' \models Rx\}| = 1 \\ & \text{but} & |\{t' \in T_{\mathbf{N}} \mid t' =_X t,\ \mathbf{N}, t' \models Rx\}| \quad = n+1. \end{array}$$

**Proposition A.5.** Let $(\tau, V)$ be a finite type with $x, y, z \in V$ and define $\sigma$ for $(\tau, V)$ as in the context of the modal translation, see Definition 4.12. Set

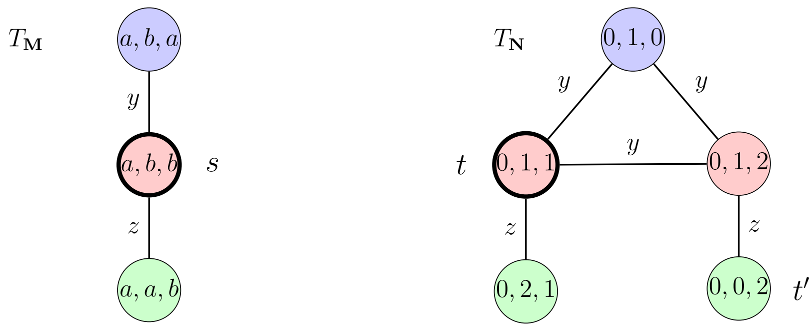
$$\varphi(s) \coloneqq \forall t (t \sim_x s \ \rightarrow\ (t \sim_y s \vee t \sim_z s)).$$

Then there exists no $\psi \in \mathsf{LFD}$ with $\mathsf{tr}_{\mathrm{mod}}(\psi) \equiv_{\mathcal{EQD}} \varphi$.

*Proof.* The notation $\equiv_{\mathcal{EQD}}$ was introduced in Notation 4.30. First consider the special case $\tau = \varnothing$ and $V = \{x, y, z\}$. We construct $\mathsf{LFD}$-bisimilar $(\tau, V)$-dependence models $\mathbf{M}, s$ and $\mathbf{N}, t$ such that their translations via $\mathsf{T}_{\mathrm{dep} \to \mathrm{eqd}}$ disagree on $\varphi$.

[2] Remember that when an order of $V$ is clear from context, we denote assignments by their tuple of values, so here the number 0 listed for $T_{\mathbf{N}}$ represents the assignments $(x \mapsto 0) \in T_{\mathbf{N}}$.

Consider $\mathbf{M}$ with universe $M = \{a, b\}$ and $\mathbf{N}$ with universe $N = \{0, 1, 2\}$. We denote assignments $s$ by their tuple of values $s(xyz)$. Depicted below are $T_{\mathbf{M}}$ and $T_{\mathbf{N}}$. The colors represent a bisimulation $Z \subseteq T_{\mathbf{M}} \times T_{\mathbf{N}}$, (e.g. since $(a, a, b), (0, 2, 1), (0, 0, 2)$ are green, we have $(a, a, b)Z(0, 2, 1)$ and $(a, a, b)Z(0, 0, 2)$). The nodes with a thicker border mark $s = (a, b, b)$ and $t = (0, 1, 1)$. The labels of the edges denote on which variables the conjoined assignments agree, i.e. they represent $=_y$ and $=_z$. We left out $=_x$ for ease of presentation.




This view of the models $\mathbf{M}$ and $\mathbf{N}$ as is essentially already the modal perspective, viewing the assignments as atomic objects. In particular, remember that $T_{\mathbf{M}}$ is the universe of $\mathsf{T}_{\text{dep}\to\text{eqd}}(\mathbf{M})$ and the $=_x, =_y, =_z$ correspond to the equivalences $\sim_x, \sim_y, \sim_z$ on $\mathsf{T}_{\text{dep}\to\text{eqd}}(\mathbf{M})$.

It is straightforward to see that for all $s' \in T_{\mathbf{M}}$ we have $s' =_y s$ or $s' =_z s$, whereas we have $t' = (0, 0, 2) \in T_{\mathbf{N}}$ with $t' =_x t$ but $t' \neq_y t$ and $t' \neq_z t$. Hence, it readily follows that $\mathsf{T}_{\text{dep}\to\text{eqd}}(\mathbf{M}, s) \models \varphi$ but $\mathsf{T}_{\text{dep}\to\text{eqd}}(\mathbf{N}, t) \not\models \varphi$. We verify that $Z$ is indeed an $\mathsf{LFD}$-bisimulation as given in Fact 3.2. For all $(s, t) \in Z$:

1. $s$ and $t$ agree on atoms, i.e. are 0-bisimilar. There are no relations, the role of $x$ can be disregarded and $y, z$ are not constant, so we mention only

   (a) Blue nodes satisfy $D_z y$ and $\neg D_y z$.

   (b) Red nodes satisfy $\neg D_z y$ and $\neg D_y z$.

   (c) Green nodes satisfy $\neg D_z y$ and $D_y z$.

2. Up to $\sim^0$, the $=_y$-classes and $=_z$-classes of each assignment in both teams are determined by its own $\sim^0$-class. Here "up to $\sim^0$" just means "up to the same color", so for example in both teams the $=_y$-class of a red node has red and blue nodes, whereas its $=_z$-class has red and green nodes. With this information, the back and forth conditions are obvious.

Hence $Z$ is a bisimulation, so $\mathbf{M}, s \sim \mathbf{N}, t$ and therefore $\mathbf{M}, s \equiv_{\mathsf{LFD}} \mathbf{N}, t$. For any

$\psi \in \mathsf{LFD}$ we have

$$\begin{aligned}\mathsf{T}_{\text{dep} \to \text{eqd}}(\mathbf{M}, s) \models \mathsf{tr}_{\text{mod}}(\psi) \quad &\text{iff} \quad \mathbf{M}, s \models \psi \\ &\text{iff} \quad \mathbf{N}, t \models \psi \\ &\text{iff} \quad \mathsf{T}_{\text{dep} \to \text{eqd}}(\mathbf{N}, t) \models \mathsf{tr}_{\text{mod}}(\psi).\end{aligned}$$

Since $\mathsf{T}_{\text{dep} \to \text{eqd}}(\mathbf{M}, s)$ and $\mathsf{T}_{\text{dep} \to \text{eqd}}(\mathbf{N}, t)$ disagree on $\varphi$, it follows that there cannot exist a $\psi \in \mathsf{LFD}$ with $\varphi \equiv_{\mathcal{EQD}} \mathsf{tr}_{\text{mod}}(\psi)$.

As for most other examples in this appendix, we can adapt this example to arbitrary finite signatures $(\tau, V)$ where $x, y, z \in V$, by simply letting $R^{\mathbf{M}} = R^{\mathbf{N}} = \varnothing$ for all new relation symbols in $R \in \tau$, and letting all new variables $v \in V \setminus \{x, y, z\}$ be constant in both $T_{\mathbf{M}}$ and $T_{\mathbf{N}}$. This ensures that the adapted versions of the assignments that are bisimilar in the above example will still be bisimilar in the adapted example. □

# Appendix B

# Syntactic Type Models

We cite the definitions of [6, Section 4] that are relevant to the proof of Proposition 5.1, where we prove $\mathrm{Sat}(\mathsf{LFD}) \in$ 2-NEXPTIME by analyzing the proof of decidability in said section. We use everything stated about big-O-notation at the start of Section 5.1.

Let $\varphi \in \mathsf{LFD}$ with $n \coloneqq |\varphi|$ and $V_\varphi$ be the set of all occurring variables. Also consider the set $\Phi$ that contains all subformulae of $\varphi$ (at most $n^2$ many) all dependence atoms $D_X Y$ for $X, Y \subseteq V_\varphi$ (at most $2^{2n}$ many) and is closed under one round of negations, where explicit negations themselves are left as they are.

**Remark B.1.** The number of elements of $\Phi$ is bounded by $2(n^2 + 2^{2n}) \in 2^{\mathcal{O}(n)}$. Formally, the input to some algorithm is encoded over some fixed reasonable (non-unary) alphabet such as $\{0, 1\}$. Therefore a bound on the actual size of $\Phi$, encoded as string over said alphabet, is slightly larger, obtained by multiplying the cardinality of $\Phi$ with $|\varphi| = n$. But asymptotically this makes no difference, since $2^{\mathcal{O}(n)} n = 2^{\mathcal{O}(n)}$. Generally, since we are only interested in complexity classes, it does not matter whether our input is polynomially longer. In particular, it is irrelevant whether we consider $|\Phi|$ to be the cardinality or the length of a reasonable encoding of $\Phi$, because we will always use the bound $|\Phi| \in 2^{\mathcal{O}(n)}$, which holds in either case.

**Definition B.2** (Hintikka Set, [6, Definition 4.1])**.** A subset $\Sigma \subseteq \Phi$ is a Hintikka set for $\Phi$ if it satisfies the following conditions, where all formulas mentioned run over $\Phi$ only:

1. $\neg\psi \in \Sigma$ iff $\psi \notin \Sigma$.
2. $\varphi \wedge \psi \in \Sigma$ iff $\varphi, \psi \in \Sigma$.
3. if $\mathsf{D}_X\, \psi \in \Sigma$, then $\psi \in \Sigma$.
4. $D_X x \in \Sigma$ for all $x \in X \subseteq V_\varphi$.
5. if $D_X Y, D_Y Z \in \Sigma$, then $D_X Z \in \Sigma$.

Note that deciding whether $\Sigma \subseteq \Phi$ is a Hintikka set for $\Phi$ is clearly possible in time that is polynomial in $|\Phi| \in 2^{\mathcal{O}(n)}$, so overall still in time $2^{\mathcal{O}(n)}$.

**Definition B.3** (Dependence closure, [6, Definition 4.2]). Given a Hintikka set $\Sigma \subseteq \Phi$, a set of variables $X \subseteq V_\varphi$ is dependence-closed (with respect to $\Sigma$) if we have for every $y \in V_\varphi$ that $D_X y \in \Sigma$ implies $y \in X$. *The* dependence-closure of a set of variables $X \subseteq V_\varphi$ with respect to $\Sigma$ is the set

$$D_X^\Sigma \coloneqq \{y \in V_\varphi \mid D_X y \in \Sigma\}.$$

**Fact B.4** ([6, Fact 4.3]). For a Hintikka set $\Sigma \subseteq \Phi$ and some $X \subseteq V_\varphi$ we have

1. $X \subseteq D_X^\Sigma$.
2. $D_X^\Sigma$ is dependence-closed with respect to $\Sigma$.

Also note that we can compute $D_X^\Sigma$ in time $|V_\varphi||\Sigma| \in 2^{\mathcal{O}(n)}$, by linearly searching $\Sigma$ for $D_X y$, for each $y \in V_\varphi$.

**Definition B.5** ([6, Definition 4.4]). For Hintikka sets $\Sigma, \Delta \subseteq \Phi$ and $X \subseteq V_\varphi$,

$$\Sigma \sim_X \Delta \qquad \text{iff} \qquad \Sigma \text{ and } \Delta \text{ have the same formulae } \psi \in \Phi \text{ with } \mathrm{Free}(\psi) \subseteq X.$$

Again it is easy to see that we can check $\Sigma \sim_X \Delta$ in time that is polynomial in $|\Sigma| + |\Delta| \leq 2|\Phi| \in 2^{\mathcal{O}(n)}$, hence overall in time $2^{\mathcal{O}(n)}$.

**Definition B.6** (Type model, [6, Definition 4.6]). A type model for $\Phi$ is a set $\mathfrak{M}$ of Hintikka sets for $\Phi$ obeying the following "witness condition" for existential dependence quantifiers:

(W) If $\mathsf{E}_X\, \psi \in \Sigma \in \mathfrak{M}$ and $Y \coloneqq D_X^\Sigma$, then there exists $\Delta \in \mathfrak{M}$, such that $\psi \in \Delta$ and $\Sigma \sim_Y \Delta$.

**Remark B.7.** The same argument about length vs. cardinality we made for $|\Phi|$ in Remark B.1 holds also for any set $\mathfrak{M}$ of subsets of $\Phi$, so in particular for type models. We will simply state that for such sets $\mathfrak{M}$,

$$|\mathfrak{M}| \leq \sum_{\Delta \subseteq \Phi} |\Delta| \leq 2^{|\Phi|}|\Phi| \in 2^{2^{\mathcal{O}(n)}} \cdot 2^{\mathcal{O}(n)} = 2^{2^{\mathcal{O}(n)}}.$$

**Lemma B.8.** For a given set $\mathfrak{M}$ of Hintikka sets for $\Phi$, we can check the witness condition for $\mathfrak{M}$ in time $2^{2^{\mathcal{O}(n)}}$.

*Proof.* Consider the following procedure.

(1) Iterate over all formulae in all Hintikka sets of $\mathfrak{M}$.

(2) Whenever we find $\mathsf{E}_X\,\psi \in \Sigma \in \mathfrak{M}$:

   (a) Compute the dependence closure $Y := D_X^\Sigma$.

   (b) Iterate over all formulae in all Hintikka sets of $\mathfrak{M}$ (nested in (1)).

   (c) If we find that $\psi$ is contained in some $\Delta \in \mathfrak{M}$:

      i. Check whether $\Sigma \sim_Y \Delta$.

      ii. If yes, continue the outer iteration (1), otherwise the inner one (b).

   (d) If we do not find such a $\Delta$ containing $\psi$, Reject.

(3) Accept.

Clearly every step, and hence the whole process, is possible in time that is polynomial in $2^{2^{\mathcal{O}(n)}}$. Overall, we can check the witness condition in time $2^{2^{\mathcal{O}(n)}}$. □

**Corollary B.9.** Given some set $\mathfrak{M}$ of subsets of $\Phi$, we can verify in time $2^{2^{\mathcal{O}(n)}}$ whether $\mathfrak{M}$ is a type model for $\Phi$.

*Proof.* To do this, we only need the following two steps:

1. Verify that each of the sets in $\mathfrak{M}$ is a Hintikka-Set of $\Phi$. A single set can be checked in time $2^{\mathcal{O}(n)}$ (cf. Definition B.2). We have at most $|\mathfrak{M}| \in 2^{2^{\mathcal{O}(n)}}$ such sets. Hence this first step can be done in deterministic time $2^{2^{\mathcal{O}(n)}} 2^{\mathcal{O}(n)} = 2^{2^{\mathcal{O}(n)}}$.

2. Check the witness condition for $\mathfrak{M}$ in time $2^{2^{\mathcal{O}(n)}}$.

Overall this takes time $2^{2^{\mathcal{O}(n)}}$. □

# Appendix C

# Further Details

**Lemma C.1.** Let $\mathcal{L}$ denote $\mathsf{LFD}$ or $\mathsf{LFD}^=$ and $\psi \in \mathcal{L}_\infty$. If the (quantifier) rank of $\psi$ as defined in Definition 3.7 is some limit ordinal $\lambda$, then we can write $\psi$ as a boolean combination of formulae $\varphi_i$ with $\mathsf{qr}(\varphi_i) < \lambda$.

*Proof.* Fix some limit ordinal $\lambda$. We do a simple induction over the way we constructed formulae in $\mathcal{L}_\infty$. Clearly no $\psi \in \mathcal{L}$ can have quantifier rank $\lambda$, so the claimed implication "$\mathsf{qr}(\varphi) = \lambda$ implies $\varphi$ is can be written as a boolean combination of $\varphi_i$ with $\mathsf{qr}(\varphi_i) < \lambda$" holds true for all $\varphi \in \mathcal{L}$.

Now assume that the claimed implication holds for $\varphi \in \mathcal{L}_\infty$, then it obviously also holds for $\neg\varphi$. Moreover, $\mathsf{qr}(\mathsf{D}_X\,\varphi) \neq \lambda$ for all finite $X \subseteq V$, since $\mathsf{qr}(\mathsf{D}_X\,\varphi)$ is always a successor ordinal, so the implication also holds for $\mathsf{D}_X\,\varphi$.

The last induction step is conjunction. So assume that $\Phi \subseteq \mathcal{L}_\infty$ is a set of formulae such that the claimed implication holds for every $\varphi \in \Phi$. If $\mathsf{qr}(\bigwedge \Phi) = \sup_{\varphi\in\Phi} \mathsf{qr}(\varphi) = \lambda$, then $\mathsf{qr}(\varphi) \leq \lambda$ for all $\varphi \in \Phi$. Moreover, we use the induction hypothesis to write every $\varphi \in \Phi$ with $\mathsf{qr}(\varphi) = \lambda$ as a boolean combination of $\varphi_i$ with $\mathsf{qr}(\varphi_i) < \lambda$. But then $\bigwedge \Phi$ itself is written as a boolean combination of formulae $\varphi_i$ with $\mathsf{qr}(\varphi_i) < \lambda$. Hence the claimed implication also holds for $\bigwedge \Phi$. This concludes the induction. □

**Corollary C.2.** Let $\mathbf{M}, s$ and $\mathbf{N}, t$ be dependence models of the same (not necessarily finite) type. Also let $\lambda$ be a limit ordinal. Then

$$\mathbf{M}, s \equiv^\lambda_{\mathcal{L}_\infty} \mathbf{N}, t \qquad \text{iff} \qquad \mathbf{M}, s \equiv^\alpha_{\mathcal{L}_\infty} \mathbf{N}, t, \quad \alpha < \lambda.$$

**Theorem C.3** (Ehrenfeucht-Fraïssé and Karp theorems for $\mathsf{LFD}$ and $\mathsf{LFD}^=$)**.**
Let $\mathcal{L}$ denote $\mathsf{LFD}$ or $\mathsf{LFD}^=$. For $\mathbf{M}, s$ and $\mathbf{N}, t$ of the same *finite* type it holds that

$$\mathbf{M}, s \sim^k_{\mathcal{L}} \mathbf{N}, t \qquad \text{iff} \qquad \mathbf{M}, s \equiv^k_{\mathcal{L}} \mathbf{N}, t, \qquad k \in \mathbb{N}.$$

As a consequence we obtain that under those same conditions

$$\mathbf{M}, s \sim^{\omega}_{\mathcal{L}} \mathbf{N}, t \qquad \text{iff} \qquad \mathbf{M}, s \equiv_{\mathcal{L}} \mathbf{N}, t.$$

For arbitrary (i.e. not necessarily finite) types it holds that

$$\mathbf{M}, s \sim^{\alpha}_{\mathcal{L}} \mathbf{N}, t \qquad \text{iff} \qquad \mathbf{M}, s \equiv^{\alpha}_{\mathcal{L}_\infty} \mathbf{N}, t, \qquad\qquad \alpha \in \mathbf{Ord}$$

and therefore

$$\mathbf{M}, s \sim_{\mathcal{L}} \mathbf{N}, t \qquad \text{iff} \qquad \mathbf{M}, s \equiv^{\infty}_{\mathcal{L}} \mathbf{N}, t.$$

*Proof.* The finite case was already shown in Theorem 3.12. For the sake of completeness, we state the analogous proof for the case of infinite types and $\mathcal{L}_\infty$.

Consider some arbitrary type $(\tau, V)$ and let $\mathbf{M}, s$ and $\mathbf{N}, t$ be $(\tau, V)$ dependence models. It is again clear that $\sim^0_{\mathcal{L}}$ corresponds to $\equiv^0_{\mathcal{L}_\infty}$. We proceed by (transfinite) induction. First consider successor ordinals: So assume that we have shown that $\sim^{\alpha}_{\mathcal{L}}$ coincides with $\equiv^{\alpha}_{\mathcal{L}_\infty}$ for some $\alpha \in \mathbf{Ord}$. We want to show the same for $\alpha + 1$:

1. "$\Longrightarrow$": We assume $\mathbf{M}, s \sim^{\alpha+1}_{\mathcal{L}} \mathbf{N}, t$ and show $\mathbf{M}, s \equiv^{\alpha+1}_{\mathcal{L}_\infty} \mathbf{N}, t$.

   A formula of quantifier rank $\alpha+1$ is just a boolean combination of at least one formula of the form $\mathsf{D}_X\, \varphi$ with $\mathsf{qr}(\varphi) = \alpha$, and other formulae with quantifier rank at most $\alpha$. Since $(\alpha+1)$-$\mathcal{L}$-bisimilarity entails $\alpha$-$\mathcal{L}$-bisimilarity, the induction hypothesis yields $\mathbf{M}, s \equiv^{\alpha}_{\mathcal{L}_\infty} \mathbf{N}, t$. So it suffices to show that $\mathbf{M}, s \models \mathsf{D}_X\, \varphi$ iff $\mathbf{N}, t \models \mathsf{D}_X\, \varphi$, for all finite $X \subseteq V$ and $\varphi \in \mathcal{L}_\infty(\tau, V)$ of quantifier rank $\alpha$.

   To this end, suppose $\mathbf{M}, s \models \mathsf{D}_X\, \varphi$ for such an $X$ and $\varphi$. We want to show $\mathbf{N}, t \models \mathsf{D}_X\, \varphi$. If $t' \in T_{\mathbf{N}}$ is an arbitrary assignment in the $=_X$-class of $t$, then by the $(\alpha + 1)$-back condition there exists some $s' \in T_{\mathbf{M}}$ with

   $$\mathbf{M}, s' \sim^{\alpha}_{\mathcal{L}} \mathbf{N}, t' \qquad \text{and} \qquad s' =_X s.$$

   Since we assumed $\mathbf{M}, s \models \mathsf{D}_X\, \varphi$ this implies $\mathbf{M}, s' \models \varphi$. By the induction hypothesis we infer from $\mathbf{M}, s' \sim^{\alpha}_{\mathcal{L}} \mathbf{N}, t'$ that

   $$\mathbf{M}, s' \equiv^{\alpha}_{\mathcal{L}_\infty} \mathbf{N}, t' \qquad \text{and therefore} \qquad \mathbf{N}, t' \models \varphi.$$

   As $t'$ was an arbitrary assignment in the $=_X$-class of $t$, we conclude that $\mathbf{N}, t \models \mathsf{D}_X\, \varphi$. The converse implication "$\mathbf{N}, t \models \mathsf{D}_X\, \varphi$ implies $\mathbf{M}, s \models \mathsf{D}_X\, \varphi$" follows analogously, using the $(\alpha+1)$-forth condition instead. By our above argument we obtain that $\mathbf{M}, s \equiv^{\alpha+1}_{\mathcal{L}_\infty} \mathbf{N}, t$, and conclude that $(\alpha+1)$-$\mathcal{L}$-bisimilarity implies $\mathcal{L}_\infty$-equivalence up to rank $\alpha + 1$.

2. "$\Longleftarrow$": We assume that $\mathbf{M}, s \equiv^{\alpha+1}_{\mathcal{L}_\infty} \mathbf{N}, t$ and show $\mathbf{M}, s \sim^{\alpha+1}_{\mathcal{L}} \mathbf{N}, t$.

Since $\mathbf{M}, s$ satisfies its own characteristic formula of rank $\alpha + 1$, we obtain

$$\mathbf{N}, t \models \chi^{\alpha+1}_{\mathbf{M},s} \qquad \text{which is equivalent to} \qquad \mathbf{M}, s \sim^{\alpha+1}_{\mathcal{L}} \mathbf{N}, t$$

by Lemma 3.11. Hence $\mathcal{L}_\infty$-equivalence up to rank $\alpha + 1$ implies $(\alpha + 1)$-$\mathcal{L}$-bisimilarity.

This concludes the induction step for successor ordinals. Now let $\lambda$ be a limit ordinal and assume we have shown the claim for all $\alpha < \lambda$. Using Corollary C.2 we obtain

$$\begin{aligned} \mathbf{M}, s \sim^{\lambda}_{\mathcal{L}} \mathbf{N}, t \quad & \text{iff} \quad \mathbf{M}, s \sim^{\alpha}_{\mathcal{L}} \mathbf{N}, t, \quad \alpha < \lambda \\ & \text{iff} \quad \mathbf{M}, s \equiv^{\alpha}_{\mathcal{L}_\infty} \mathbf{N}, t, \quad \alpha < \lambda \\ & \text{iff} \quad \mathbf{M}, s \equiv^{\lambda}_{\mathcal{L}_\infty} \mathbf{N}, t. \end{aligned}$$

This concludes the induction step for limit ordinals. Now we know that $\alpha$-$\mathcal{L}$-bisimilarity coincides with $\mathcal{L}_\infty$-equivalence up to rank $\alpha$, for all ordinals $\alpha$. As a corollary we then obtain

$$\begin{aligned} \mathbf{M}, s \sim_{\mathcal{L}} \mathbf{N}, t \quad & \text{iff} \quad \mathbf{M}, s \sim^{\alpha}_{\mathcal{L}} \mathbf{N}, t, \quad \alpha \in \mathbf{Ord} \\ & \text{iff} \quad \mathbf{M}, s \equiv^{\alpha}_{\mathcal{L}_\infty} \mathbf{N}, t, \quad \alpha \in \mathbf{Ord} \\ & \text{iff} \quad \mathbf{M}, s \equiv^{\infty}_{\mathcal{L}} \mathbf{N}, t. \end{aligned}$$

□

**Proposition C.4.** Let $(\tau, V)$ be a finite type and $(\mathcal{C}[\sigma], \mathsf{F}, \mathsf{G}, \mathsf{tr}, (\vartheta_X)_{X \subseteq V})$ a good translation of $\mathsf{LFD}$ over $\mathcal{DEP}[\tau, V]$ to $\mathsf{FO}$ over $\mathcal{C}[\sigma]$. If $k \in \mathbb{N}$, then

$$\mathfrak{A}, \mathbf{a} \sim^k \mathfrak{B}, \mathbf{b} \qquad \text{iff} \qquad \mathsf{G}(\mathfrak{A}, \mathbf{a}) \sim^k \mathsf{G}(\mathfrak{B}, \mathbf{b})$$

for all $(\mathfrak{A}, \mathbf{a}), (\mathfrak{B}, \mathbf{b}) \in \mathcal{C}[\sigma]$. Likewise $\mathfrak{A}, \mathbf{a} \sim \mathfrak{B}, \mathbf{b}$ iff $\mathsf{G}(\mathfrak{A}, \mathbf{a}) \sim \mathsf{G}(\mathfrak{B}, \mathbf{b})$.

*Proof.* We prove the first claim via an induction on $k$ for all $(\mathfrak{A}, \mathbf{a}), (\mathfrak{B}, \mathbf{b}) \in \mathcal{C}[\sigma]$ simultaneously. The base case of $k = 0$ is clear, since by Definition 4.29

$$\begin{aligned} \mathfrak{A}, \mathbf{a} \sim^0 \mathfrak{B}, \mathbf{b} \quad & \text{iff} \quad \mathfrak{A}, \mathbf{a} \equiv^0_{\mathsf{LFD}} \mathfrak{B}, \mathbf{b} \\ & \text{iff} \quad \mathsf{G}(\mathfrak{A}, \mathbf{a}) \equiv^0_{\mathsf{LFD}} \mathsf{G}(\mathfrak{B}, \mathbf{b}) \\ & \text{iff} \quad \mathsf{G}(\mathfrak{A}, \mathbf{a}) \sim^0 \mathsf{G}(\mathfrak{B}, \mathbf{b}). \end{aligned}$$

For the induction step, assume the correspondence holds for $k \in \mathbb{N}$.

1. "$\Longrightarrow$": We assume $\mathfrak{A}, \mathbf{a} \sim^{k+1} \mathfrak{B}, \mathbf{b}$ and show $\mathsf{G}(\mathfrak{A}, \mathbf{a}) \sim^{k+1} \mathsf{G}(\mathfrak{B}, \mathbf{b})$.

   First, let $s := \mathsf{G}(\mathbf{a}) \in T_{\mathsf{G}(\mathfrak{A})}$ and $t := \mathsf{G}(\mathbf{b}) \in T_{\mathsf{G}(\mathfrak{B})}$. Since $\mathfrak{A}$ and $\mathfrak{B}$ are clear from context, we will simply write $\mathbf{a} \sim^{k+1} \mathbf{b}$ instead of $\mathfrak{A}, \mathbf{a} \sim^{k+1} \mathfrak{B}, \mathbf{b}$

and $s \sim^{k+1} t$ instead of $\mathsf{G}(\mathfrak{A}, \mathbf{a}) \sim^{k+1} \mathsf{G}(\mathfrak{B}, \mathbf{b})$ in the following. Hence, the situation is that we assumed $\mathbf{a} \sim^{k+1} \mathbf{b}$ and want to show $s \sim^{k+1} t$. We verify the $(k+1)$-back condition, see Definition 3.4.

Let $t' \in T_{\mathsf{G}(\mathfrak{B})}$. Per definition of good translations (cf. Definition 4.23), we have the surjection $\mathsf{G}\colon \mathrm{Team}(\mathfrak{B}) \twoheadrightarrow T_{\mathsf{G}(\mathfrak{B})}$, so there exists $\mathbf{b}' \in \mathrm{Team}(\mathfrak{B})$ with $\mathsf{G}(\mathbf{b}') = t'$. Let $X \subseteq t' \overline{\cap} t$, so $\mathfrak{B} \models \vartheta_X(\mathbf{b}', \mathbf{b})$. By the $(k+1)$-back condition for $\mathbf{a} \sim^{k+1} \mathbf{b}$ as defined in Definition 4.32, we know there exists $\mathbf{a}' \in \mathrm{Team}(\mathfrak{A})$ with $\mathbf{a}' \sim^k \mathbf{b}'$ and $\mathfrak{A} \models \vartheta_X(\mathbf{a}', \mathbf{a})$. Then we can set $s' := \mathsf{G}(\mathbf{a}') \in T_{\mathsf{G}(\mathfrak{A})}$ to obtain that $s' =_X s$, and by the induction hypothesis $s' \sim^k t'$. This proves the $(k+1)$-back condition for $s \sim^{k+1} t$, see Definition 3.4.

The $(k+1)$-forth condition is shown analogously, using the $(k+1)$-forth condition for $\mathbf{a} \sim^{k+1} \mathbf{b}$ instead. Hence we obtain that $s \sim^{k+1} t$.

2. "$\Longleftarrow$": Using the same notation as above, $s \sim^{k+1} t$ and show $\mathbf{a} \sim^{k+1} \mathbf{b}$. We verify the $(k+1)$-back condition, as given in Definition 4.32.

   Let $\mathbf{b}' \in \mathrm{Team}(\mathfrak{B})$ and $X \subseteq V$ with $\mathfrak{B} \models \vartheta_X(\mathbf{b}', \mathbf{b})$. For $t' := \mathsf{G}(\mathbf{b}')$ we have $t' =_X t$, so by the $(k+1)$-back condition as given in Definition 3.4, we obtain an $s' \in T_{\mathsf{G}(\mathfrak{A})}$ with $s' =_X s$ and $s' \sim^k, t'$. Since we have the surjection $\mathsf{G}\colon \mathrm{Team}(\mathfrak{A}) \twoheadrightarrow T_{\mathsf{G}(\mathfrak{A})}$, we find an $\mathbf{a}' \in \mathrm{Team}(\mathfrak{A})$ with $\mathsf{G}(\mathbf{a}') = s'$. Then $\mathfrak{A} \models \vartheta_X(\mathbf{a}', \mathbf{a})$, and per induction hypothesis $\mathbf{a}' \sim^k \mathbf{b}'$. This proves the $(k+1)$-back condition for $\mathbf{a} \sim^{k+1} \mathbf{b}$, see Definition 4.32.

   The $(k+1)$-forth condition is shown analogously, using the $(k+1)$-forth condition for $s \sim^{k+1} t$ instead. Hence we obtain that $\mathbf{a} \sim^{k+1} \mathbf{b}$.

This concludes the induction.

The case for full bisimulation is shown analogously. One compares Definitions 3.1 and 4.31, and uses the surjection on teams induced by $\mathsf{G}$ in the same fashion as above. Given a bisimulation $Z \subseteq \mathrm{Team}(\mathfrak{A}) \times \mathrm{Team}(\mathfrak{B})$, the set

$$L := \{(\mathsf{G}(\mathbf{a}), \mathsf{G}(\mathbf{b})) \mid (\mathbf{a}, \mathbf{b}) \in Z\} \subseteq T_{\mathsf{G}(\mathfrak{A})} \times T_{\mathsf{G}(\mathfrak{B})}$$

is then a bisimulation between $\mathsf{G}(\mathfrak{A})$ and $\mathsf{G}(\mathfrak{B})$. For the other direction, if $L \subseteq T_{\mathsf{G}(\mathfrak{A})} \times T_{\mathsf{G}(\mathfrak{B})}$ is a bisimulation between $\mathsf{G}(\mathfrak{A})$ and $\mathsf{G}(\mathfrak{B})$, then

$$Z := \{(\mathbf{a}, \mathbf{b}) \mid (\mathsf{G}(\mathbf{a}), \mathsf{G}(\mathbf{b})) \in L\} \subseteq \mathrm{Team}(\mathfrak{A}) \times \mathrm{Team}(\mathfrak{B})$$

is a bisimulation between $\mathfrak{A}$ and $\mathfrak{B}$. ☐

**Proposition C.5.** Our translation of $\mathsf{LFD}$ into $\mathsf{FO}(\subseteq, |)$ which we defined in Section 4.3 works as intended. Namely, for every $\varphi \in \mathsf{LFD}$ and fitting dependence model $\mathbf{M}, s \in \mathcal{DEP}$ with $\mathbf{M} = (\mathfrak{M}, T_{\mathbf{M}})$ we have $\mathsf{tr}_{\text{team}}(\varphi) \in \mathsf{FO}(\subseteq, |)$ and

$$\mathbf{M}, s \models \varphi \qquad \text{iff} \qquad \mathfrak{M} \models_{T_{\mathbf{M},s}} \mathsf{tr}_{\text{team}}(\varphi).$$

*Proof.* The claim follows from an induction on $\varphi$ in negation normal form. The base case for literals $\varphi \in \mathsf{LFD}$ as well as the induction steps for boolean connectives $\wedge, \vee$ are easily verified. Consider the following technical observation:

$$\mathfrak{M} \models_{T_{\mathbf{M},s}[\mathbf{v}' \mapsto t(\mathbf{v})]} \mathsf{tr}_{\text{team}}(\varphi)[\tilde{\mathbf{v}} \mapsto \mathbf{v}'] \qquad \text{iff} \qquad \mathfrak{M} \models_{T_{\mathbf{M},t}} \mathsf{tr}_{\text{team}}(\varphi). \tag{C.1}$$

Indeed, we have $\text{Free}(\mathsf{tr}_{\text{team}}(\varphi)[\tilde{\mathbf{v}} \mapsto \mathbf{v}']) \subseteq V \uplus V'$ and

$$T_{\mathbf{M},s}[\mathbf{v}' \mapsto t(\mathbf{v})] \upharpoonright (V \uplus V') \;=\; (T_{\mathbf{M},s} \upharpoonright V)[\mathbf{v}' \mapsto t(\mathbf{v})] \;=\; T_{\mathbf{M}}[\mathbf{v}' \mapsto t(\mathbf{v})].$$

Therefore, by locality of $\mathsf{FO}(\subseteq, |)$ (cf. Lemma 4.56), we obtain

$$\begin{aligned} & \mathfrak{M} \models_{T_{\mathbf{M},s}[\mathbf{v}' \mapsto t(\mathbf{v})]} \mathsf{tr}_{\text{team}}(\varphi)[\tilde{\mathbf{v}} \mapsto \mathbf{v}'] \\ \text{iff} \quad & \mathfrak{M} \models_{T_{\mathbf{M},s}[\mathbf{v}' \mapsto t(\mathbf{v})] \upharpoonright (V \uplus V')} \mathsf{tr}_{\text{team}}(\varphi)[\tilde{\mathbf{v}} \mapsto \mathbf{v}'] \\ \text{iff} \quad & \mathfrak{M} \models_{T_{\mathbf{M}}[\mathbf{v}' \mapsto t(\mathbf{v})]} \mathsf{tr}_{\text{team}}(\varphi)[\tilde{\mathbf{v}} \mapsto \mathbf{v}']. \end{aligned}$$

Now notice that no variables of $\widetilde{V}$ occur anymore in the domain of the regarded team or as variables in the regarded formula. Since variables have no intrinsic meaning in team semantics, we can simply replace the variables $V'$ with those in $\widetilde{V}$. Per definition we have $T_{\mathbf{M}}[\tilde{\mathbf{v}} \mapsto t(\mathbf{v})] = T_{\mathbf{M},t}$, so we can continue the above equivalences with

$$\begin{aligned} & \mathfrak{M} \models_{T_{\mathbf{M}}[\mathbf{v}' \mapsto t(\mathbf{v})]} \mathsf{tr}_{\text{team}}(\varphi)[\tilde{\mathbf{v}} \mapsto \mathbf{v}'] \\ \text{iff} \quad & \mathfrak{M} \models_{T_{\mathbf{M}}[\tilde{\mathbf{v}} \mapsto t(\mathbf{v})]} \mathsf{tr}_{\text{team}}(\varphi) \\ \text{iff} \quad & \mathfrak{M} \models_{T_{\mathbf{M},t}} \mathsf{tr}_{\text{team}}(\varphi). \end{aligned}$$

This proves the equivalence (C.1). The induction steps for $\mathsf{E}_{\mathbf{x}}\,\varphi$ and $\mathsf{D}_{\mathbf{x}}\,\varphi$ are now straightforward. With the definition of $\mathsf{tr}_{\text{team}}(\mathsf{E}_{\mathbf{x}}\,\varphi)$ and (C.1) we obtain

$$\begin{aligned} & \mathfrak{M} \models_{T_{\mathbf{M},s}} \mathsf{tr}_{\text{team}}(\mathsf{E}_{\mathbf{x}}\,\varphi) & \\ \text{iff} \quad & \mathfrak{M} \models_{T_{\mathbf{M},s}[\mathbf{v}' \mapsto t(\mathbf{v})]} \mathsf{tr}_{\text{team}}(\varphi)[\tilde{\mathbf{v}} \mapsto \mathbf{v}'] & \text{for some } t \in T_{\mathbf{M}} \text{ with } t =_{\mathbf{x}} s \\ \text{iff} \quad & \mathfrak{M} \models_{T_{\mathbf{M},t}} \mathsf{tr}_{\text{team}}(\varphi) & \text{for some } t \in T_{\mathbf{M}} \text{ with } t =_{\mathbf{x}} s \\ \text{iff} \quad & \mathbf{M}, t \models \varphi & \text{for some } t \in T_{\mathbf{M}} \text{ with } t =_{\mathbf{x}} s \\ \text{iff} \quad & \mathbf{M}, s \models \mathsf{E}_{\mathbf{x}}\,\varphi, & \end{aligned}$$

where we also used the induction hypothesis in the second to last step. Analogously

$$\begin{array}{lll}
& \mathfrak{M} \models_{T_{\mathbf{M},s}} \mathsf{tr}_{\text{team}}(\mathsf{D}_{\mathbf{x}}\,\varphi) & \\
\text{iff} & \mathfrak{M} \models_{T} \mathbf{x}' = \widetilde{\mathbf{x}} \to \Gamma\mathbf{v}'.(\mathsf{tr}_{\text{team}}(\varphi)[\widetilde{\mathbf{v}} \mapsto \mathbf{v}']) & \text{where } T = \bigcup_{t \in T_{\mathbf{M}}} T_{\mathbf{M},s}[\mathbf{v}' \mapsto t(\mathbf{v})] \\
\text{iff} & \mathfrak{M} \models_{T} \Gamma\mathbf{v}'.(\mathsf{tr}_{\text{team}}(\varphi)[\widetilde{\mathbf{v}} \mapsto \mathbf{v}']) & \text{where } T = \bigcup_{\substack{t \in T_{\mathbf{M}} \\ t =_{\mathbf{x}} s}} T_{\mathbf{M},s}[\mathbf{v}' \mapsto t(\mathbf{v})] \\
\text{iff} & \mathfrak{M} \models_{T_{\mathbf{M},s}[\mathbf{v}' \mapsto t(\mathbf{v})]} \mathsf{tr}_{\text{team}}(\varphi)[\widetilde{\mathbf{v}} \mapsto \mathbf{v}'] & \text{for all } t \in T_{\mathbf{M}} \text{ with } t =_{\mathbf{x}} s \\
\text{iff} & \mathfrak{M} \models_{T_{\mathbf{M},t}} \mathsf{tr}_{\text{team}}(\varphi) & \text{for all } t \in T_{\mathbf{M}} \text{ with } t =_{\mathbf{x}} s \\
\text{iff} & \mathbf{M}, t \models \varphi & \text{for all } t \in T_{\mathbf{M}} \text{ with } t =_{\mathbf{x}} s \\
\text{iff} & \mathbf{M}, s \models \mathsf{D}_{\mathbf{x}}\,\varphi. & \qquad \square
\end{array}$$

# Bibliography


[1] S. Abramsky, J. Kontinen, J. Väänänen, and H. Vollmer. *Dependence Logic. Theory and Applications*. 2017.

[2] H. Andréka, J. van Benthem, and I. Németi. Modal Languages and Bounded Fragments of Predicate Logic. In: *Journal of Philosophical Logic* 27 (1998), pp. 217–274.

[3] W. W. Armstrong. Dependency Structures of Data Base Relationships. In: *IFIP Congress*. 1974.

[4] J. Balcázar, J. Díaz, and J. Gabarró. *Structural Complexity I*. Springer, 1988.

[5] J. Balcázar, J. Díaz, and J. Gabarró. *Structural Complexity II*. Springer, 1990.

[6] A. Baltag and J. van Benthem. A Simple Logic of Functional Dependence, 2020. [Pre-print] `https://eprints.illc.uva.nl/1733`. An improved version was submitted to JPL.

[7] J. van Benthem. Modal Correspondence Theory. PhD thesis. University of Amsterdam, 1976.

[8] J. van Benthem. *Modal Logic and Classical Logic*. Bibliopolis, 1983.

[9] J. van Benthem. Dynamic Bits and Pieces. In: *ILLC research report*. University of Amsterdam, 1997.

[10] D. Berwanger and E. Grädel. Games and Model Checking for Guarded Logics. In: *Logic for Programming, Artificial Intelligence, and Reasoning*. Springer, 2001, pp. 70–84.

[11] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge University Press, 2001.

[12] E. Börger, E. Grädel, and Y. Gurevich. *The Classical Decision Problem*. Springer, 1997.

[13] C. C. Chang and H. J. Keisler. *Model Theory*. Elsevier Science, 1990.

[14] A. Dawar and M. Otto. Modal characterisation theorems over special classes of frames. In: *Ann. Pure Appl. Logic* 161 (Oct. 2009), pp. 1–42.

[15] M. Fürer. The computational complexity of the unconstrained limited domino problem (with implications for logical decision problems). In: *Logic and Machines: Decision Problems and Complexity*. Springer, 1984, pp. 312–319.

[16] P. Galliani. Inclusion and exclusion dependencies in team semantics: On some logics of imperfect information. In: *Ann. Pure Appl. Log.* 163 (2012), pp. 68–84.

[17] E. Grädel. On the Restraining Power of Guards. In: *The Journal of Symbolic Logic* 64.4 (1999), pp. 1719–1742.

[18] E. Grädel. Why Are Modal Logics So Robustly Decidable? In: *BULLETIN EATCS* 68 (1999), pp. 90–103.

[19] E. Grädel and M. Otto. On logics with two variables. In: *Theoretical Computer Science* 224.1 (1999), pp. 73–113.

[20] E. Grädel and M. Otto. The Freedoms of (Guarded) Bisimulation. In: *Johan van Benthem on Logic and Information Dynamics*. Springer, 2014, pp. 3–31.

[21] E. Grädel and J. Väänänen. Dependence and Independence. In: *Studia Logica* 283 (Aug. 2012).

[22] E. Grädel and I. Walukiewicz. Guarded fixed point logic. In: *Proceedings. 14th Symposium on Logic in Computer Science (Cat. No. PR00158)*. 1999, pp. 45–54.

[23] Erich Grädel. Decision Procedures for Guarded Logics. In: *Automated Deduction — CADE-16*. Springer, 1999, pp. 31–51.

[24] Jaakko Hintikka and Gabriel Sandu. Informational Independence as a Semantical Phenomenon. In: *Logic, Methodology and Philosophy of Science VIII*. Ed. by J. E. Fenstad, I. T. Frolov, and R. Hilpinen. Vol. 126. Studies in Logic and the Foundations of Mathematics. Elsevier, 1989, pp. 571–589.

[25] W. Hodges. *A Shorter Model Theory*. Cambridge University Press, 1997.

[26] W. Hodges. Compositional Semantics for a Language of Imperfect Information. In: *Log. J. IGPL* 5 (1997), pp. 539–563.

[27] N. D. Jones and W. T. Laaser. Complete Problems for Deterministic Polynomial Time. In: *Proceedings of the Sixth Annual ACM Symposium on Theory of Computing*. STOC ’74. Association for Computing Machinery, 1974, pp. 40–46.

[28] E. Kieronski, J. Michaliszyn, I. Pratt-Hartmann, and L. Tendera. Two-Variable First-Order Logic with Equivalence Closure. In: *2012 27th Annual IEEE Symposium on Logic in Computer Science*. 2012, pp. 431–440.

[29] E. Kieroński and M. Otto. Small substructures and decidability issues for first-order logic with two variables. In: *Journal of Symbolic Logic* 77.3 (Sept. 2012), pp. 729–765.

[30] H. R. Lewis. Complexity results for classes of quantificational formulas. In: *Journal of Computer and System Sciences* 21.3 (1980), pp. 317–353.

[31] A. Mann, G. Sandu, and M. Sevenster. *Independence-friendly Logic: A game-theoretic approach*. English. London Mathematical Society Lecture Note Series. Cambridge University Press, 2011.

[32] I. Németi. Cylindric-relativised set algebras have strong amalgamation. In: *Journal of Symbolic Logic* 50.3 (1985), pp. 689–700. DOI: 10.2307/2274323.

[33] C. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[34] J. Väänänen. *Dependence Logic*. London Mathematical Society Student Texts. Cambridge University Press, 2007.