A Simple Logic of Functional Dependence

Alexandru Baltag   Johan van Benthem

Abstract

This paper presents a simple decidable logic of functional dependence LFD, based on an extension of classical propositional logic with dependence atoms plus dependence quantifiers treated as modalities, within the setting of generalized assignment semantics for first order logic. The expressive strength, complete proof calculus and meta-properties of LFD are explored. Various language extensions are presented as well, up to undecidable modal-style logics for independence and dynamic logics of changing dependence models. Finally, more concrete settings for dependence are discussed: continuous dependence in topological models, linear dependence in vector spaces, and temporal dependence in dynamical systems and games.

1   Introduction: Toward a logic of local dependence

Dependence is a ubiquitous notion, pervading areas from probability to reasoning with quantifiers, and from informational correlation in databases to causal connections or interactive social behavior. How the Moon moves depends on how the Earth moves, and vice versa. What you will do in our current Chess game depends on how I play. And dependence, or independence, matters. Whether variables are dependent or not is crucial to probabilistic calculation. And as for qualitative reasoning, dependence is at the heart of quantifier combinations in logic.

Now ubiquity does not mean unity: there need not be one coherent notion behind all talk of dependence in science or daily life.[1] Still, over the last century, various proposals have been made for a basic logic of reasoning about dependence and independence, witness publications such as [5], [55], [3], [47], [62], [71]. While some of these logics are weak calculi of pure dependence statements, others are very strong and second-order. And most of them are non-classical: the propositional connectives break classical laws such as Tertium Non Datur, while the semantics differs radically from that of First Order Logic (FOL); being either a game semantics, or some higher-order version of first-order semantics, evaluating formulas on sets of assignments.

In this paper we explore one more perspective, minimalistic in its simplicity: a Logic of Functional Dependence between variables (LFD), based on classical logic. LFD is obtained by adding

---

[1]There may not even be one uniform conception of dependence in logic running from, say, dependent choices in quantifier combinations as mentioned above to independence of sets of axioms. In particular, the latter widespread sense, studied for instance in [52], [50], is not what is at issue in this paper.

local dependence atoms to a known generalization of the usual semantics of First Order Logic, namely, the logic CRS of generalized assignment models, [3]. CRS treats quantifiers as modalities over accessibility relations between assignments, and LFD adopts this view also for further modalities for reasoning about dependence.

This new calculus, which offers a simple base level for analyzing reasoning about functional dependence, is decidable and yet reasonably expressive. Unlike the other approaches mentioned, it focuses on a 'local' sense of dependence, which may be more fundamental than the usual global version. This locality brings LFD close to modal logic, but still allows it to cover many senses of dependence, both 'ontic' in terms of linked behavior in the world, and 'epistemic' in terms of information: learning something about one thing implies learning about another. Taking this minimal modal perspective, one can then take a fresh look at the surplus of many richer notions of dependence and correlation, and design further logics for reasoning about these.

1.1 Global and local dependence in a complete database

As an example, consider relational databases with tuples of values assigned to attributes, cf. [1].

Example 1.1. Here is a simple information structure. The table below is meant as a *complete database*, i.e., a full description of the restaurant situation in a small town, sorted by four variables (so-called 'attributes'): Restaurant name, Food type, Price range, and Location.

Restaurant	Food	Price	Location
Roma	Italian	Moderate	Center
Hasta La Pasta	Italian	Cheap	Center
Mama Makan	Indonesian	Moderate	South
Bunga Mawar	Indonesian	Cheap	West
Wilde Zwider	Dutch	Expensive	East
Greetje	Dutch	Expensive	West

A variable v *depends* on a variable u (or on a *set U* of variables) if, whenever the value of u is the same in two rows (or the values of all variables in U are the same in the two rows), so is the value of v. This is in line with the intuitive sense of dependence: the value of u *determines* the value of v. From the table, we see that each of the attributes *Food*, *Price* and *Location* depends on *Restaurant* (each restaurant offers a unique type of food, has a unique price range and a unique location). But neither *Restaurant* nor *Price* nor *Location* depend on *Food* (e.g., there are two Indonesian restaurants, with different price ranges and locations), and *Food* does not depend on *Price* either (there are cheap Italian and Indonesian places). These facts imply others. E.g., *Restaurant* does not depend on *Price*: if it did, then by transitivity and the fact that *Food* depends on *Restaurant*, *Food* would depend on *Price*, quod non. In fact, *Restaurant* does not even depend on *Price* and *Food* taken together (i.e. on the set $\{Price, Food\}$): both Wilde Zwider and Greetje are expensive Dutch restaurants. But *Restaurant* does depend on *Price* and *Location* taken together: for every possible price range and location, there is at most one restaurant offering that food type in that location.

What was listed so far are *global* dependencies. However, underlying these are many *local* dependencies in the table: given a current row, a variable v locally depends on u if, every row

matching the current value of u, also matches the current value of v. For example, in the fifth row of the table, *Price* depends on *Food* (since all Dutch restaurants are expensive), *Food* locally depends on *Price* (all expensive restaurants are Dutch), and *Restaurant* locally depends on *Location* (the only restaurant located in the East is Wilde Zwider). Again, some of these things follow from others. E.g., still at the fifth row, the fact that *Food* locally depends on *Location* follows by transitivity from the local dependence of *Restaurant* on *Location* plus the global (and hence also local) dependence of *Food* on *Restaurant*.[2]

A set of variable assignments as above, with a designated current assignment, connects in an obvious way with the semantics of first-order logic. At the same time, the distinction local vs. global is crucial to the essentially *modal* approach taken in this paper.

1.2   Dependence in first-order semantics: CRS logic

What does dependence have to do with first-order logic? In classical FOL, distinct variables have no intrinsic meaning and are fully inter-changeable. No correlations between their values are allowed, beyond the fact that they all range over the given domain. Accordingly, first-order models are 'full': all possible variable assignments are available, and the FOL quantifiers can arbitrary reset the values of any variables, while keeping the values of all the other variables fixed. This effectively amounts to a strong *independence* assumption, reflected for instance in the commutation law $\exists x\, \exists y\, \varphi \leftrightarrow \exists y\, \exists x\, \varphi$. Often seen as a triviality, this validity is in fact a very strong symmetry principle, which is entailed by the above strong independence assumption.

To allow for dependencies between variables, a simple solution is to just drop the 'fullness' assumption, while keeping the good features of FOL such as its perspicuous syntax and compositional truth definition. This approach, known as *generalized assignment semantics*, was developed in the 1990s. A 'generalized assignment model' (M, A) consists of a first order model M and a family A of 'admissible' variable assignments $s : V \to O$ (with V the variables and O the objects in M), circumscribing the global states that can occur when evaluating the first-order quantifiers. These models widen the scope of applicability of FOL to settings with significant correlations, or even functional dependencies, between variables. As stated in [3], p. 46, generalized assignment semantics *"models the natural phenomenon of* dependencies *between variables: which occurs when changes in value for one variable x may induce, or be correlated with, changes in value for another variable y. (...) Dependence cannot be modeled in standard Tarskian semantics, which modifies values for variables completely arbitrarily."* In this perspective, classical FOL describes the special case of 'full models' in which all possible assignments are available, i.e. $A = O^V$.

What is the meaning of quantifiers in these generalized models? The original generalized assignment semantics, known as the logic CRS[3], simply restricts the usual Tarskian definition to the family A of admissible assignments. Unlike in FOL, polyadic quantifiers such as $\forall xy\, \varphi$ can no longer be reduced to iterated monadic ones $\forall x \forall y\, \varphi$. Hence, CRS takes polyadic quantifiers $\forall X\, \varphi$ as a primitive notion, for every finite *set* of variables $X \subseteq V$ (while defining monadic quantifiers $\forall x\, \varphi$ as just an abbreviation for $\forall \{x\} \varphi$): for any assignment $s \in A$, we put

$$s \models \forall X \varphi \quad \text{iff} \quad t \models \varphi \text{ for every } t \in A \text{ satisfying } s(y) = t(y) \text{ for all } y \in V - X.$$

---

[2]Databases may also satisfy *non-functional* dependencies. E.g., the above table satisfies the rule 'if not Dutch, then not Expensive'. Such constraints, too, can be expressed in the dependence logic of this paper.

[3]The technical name CRS stands for 'cylindric relativized set algebra', referring to algebraic origins in [63].

As we have seen, dependencies between variables are present in non-full models. In fact, the language can spot these dependencies in an *implicit* way: via the *failure* of some classical FOL validities in the weaker logic CRS. For instance, if a dependence model invalidates the above law $\exists x \exists y \phi \to \exists y \exists x \phi$, then there exist some non-trivial correlations between variables.

A key goal of generalized assignment semantics was analyzing the causes of the undecidability of validity for FOL. The intent was to decouple the desideratum of a compositional semantics for the first-order language from additional mathematical assumptions (about existence of all possible functional assignments) that increase complexity. Indeed, while CRS semantics is clearly compositional, the set of validities is decidable, forming roughly a core calculus of monotonicity and persistence reasoning inside full predicate logic.[4] Thus, CRS makes a distinction between general simple inferences inside FOL and more complex reasoning relying on special mathematical existence assumptions.[5]

This lower complexity may be understood as a result of 'modalization', [13]. The above analysis also works on abstract state models for the first-order language without underlying objects, where first-order logic becomes a modal logic. This modal perspective will be significant in what follows, as it explains how a logic of dependence can be decidable.

Still, from a dependence perspective, the CRS quantifiers have some peculiar features. Notably, the *Locality* property of FOL fails: the truth value of a CRS-formula φ need not depend only on the values of its free variables, it may well depend on values of variables that do not even occur in φ. This 'dependence on irrelevant variables' is an artifact of the specific way in which CRS generalizes FOL semantics by letting only the values of X vary, *keeping the values of all other variables fixed*, including the ones not occurring at all in the given formula.

This problem was noticed early on in the CRS literature, leading to an alternative proposal for generalizing FOL quantifiers.[6] Since these alternative operators do satisfy Locality, we will call them *local quantifiers*, denoted here by $\forall_X \varphi$:

$$s \models \forall_X \varphi \quad \text{iff} \quad t \models \varphi \text{ for every } t \in A \text{ with } s(y) = t(y) \text{ for all } y \in Free(\forall_X \varphi) = Free(\varphi) - X,$$

where $Free(\varphi)$ is the set of free variables in φ. This fixes only the values of the *actually occurring* free variables that do *not* belong to X, allowing all the others to vary.

Note that in full models (with $A = D^V$), both $\forall_X \varphi$ and $\forall X \varphi$ collapse to classical FOL quantifiers; so they are both entitled to play the role of generalized FOL quantifiers. Even so, both versions of CRS still have a major drawback: there is no explicit way to say that a variable x functionally depends on other variables. Moreover, no new validities are added that capture interesting laws of dependence. For this, further steps are needed, to be previewed now.

Remark 1.2. The language of CRS also supports modalities for *substitutions*. A formula $[y/x]\varphi$ (φ with all free occurrences of x replaced by y, where no substituted y becomes bound) is true at an assignment s if there is an available assignment t in the model equal to s except that $t(x) = s(y)$ with φ true at t. There is also a natural extension for simultaneous substitutions $[\mathbf{y}/\mathbf{x}]\varphi$, which do not reduce to iterated single ones. The usual recursive definition of syntactic substitution in

---

[4]Further axioms such as the above commutation law $\exists x \exists y \, \varphi \leftrightarrow \exists y \exists x \, \varphi$ then impose a confluent Church-Rosser structure on the set of assignments, leading to undecidability arguments via encoding tiling problems, [59].

[5]For much more information on CRS and related modal logics, cf. [74], [13], [61].

[6]See e.g. Marx [59], who attributes the proposal to Venema.

4

FOL now expresses various substantial properties of the (in general, partial) semantic substitution function on assignments and its interactions with CRS quantifiers, cf. [13]. For the proof theory of this modal view of substitution, cf. [61].

1.3 Explicit logic of local dependence

As we saw, CRS is an 'implicit' logic of dependence. In this paper, we add the explicit syntactic *atomic dependence formulas $D_X y$* of [71], now read locally as: X *locally determines (the value of)* y, or y *locally depends on X*. These atomic formulas are interpreted at assignments $s \in A$ using the local dependence relation $D_X^s y$, saying that all admissible assignments that keep the values of X fixed to the *current* ones also fix the value of y:

$$s \models D_X y \quad \text{iff} \quad s(y) = t(y) \text{ holds for every } t \in A \text{ satisfying } s(x) = t(x) \text{ for all } x \in X.$$

Next, we reconsider the quantifiers. From a dependence perspective, it is natural to introduce *dependence modalities* or *dual quantifiers* $\mathbb{D}_X \varphi$, which 'fix' the values of X to the current ones. More precisely, like the dependence atoms, these talk about all the assignments that keep X equal to its current value(s), saying that they also fix the truth value of φ to 'true':

$$s \models \mathbb{D}_X \varphi \quad \text{iff} \quad t \models \varphi \text{ holds for every } t \in A \text{ satisfying } s(x) = t(x) \text{ for all } x \in X.^7$$

We read $\mathbb{D}_X \varphi$ as X *locally determines the truth of φ*. Recall that in standard FOL, 'free' variables are the ones whose current values are kept fixed (while the values of 'bound' variables are ignored as irrelevant). This fixing the values of X explains why we sometimes call dependence modalities $\mathbb{D}_X \varphi$ 'dual quantifiers': they 'free' the variables in X (rather than binding them), while *binding all the other variables* (in $V - X$, regardless of whether they occur in the formula).

Like the local universal quantifiers $\forall_X \varphi$, dependence modalities do satisfy Locality. But they appear to be more fundamental: indeed, $\forall_X \varphi$ is simply definable via the equivalence $\forall_X \varphi \leftrightarrow \mathbb{D}_{Free(\varphi)-X} \varphi$, whereas the converse is not as straightforward.[8] Note also here that, like both FOL and CRS quantifiers $\forall X$ (but in contrast to local quantifiers $\forall_X$), dependence modalities validate the standard Distribution axiom $\mathbb{D}_X(\varphi \to \psi) \to (\mathbb{D}_X \varphi \to \mathbb{D}_X \psi)$.[9] Dependence modalities can also quantify over *all* assignments in A: taking X to be the empty set yields the *universal modality* $\mathbb{A}\varphi := \mathbb{D}_\emptyset \varphi$, saying that *all* admissible assignments satisfy φ. As a consequence, *global dependence* of y on X can be expressed as $\mathbb{A} D_X y$.

The resulting logic of functional dependence LFD is more expressive than may meet the eye, as will become clear in what follows. Also, while capturing the main properties of functional

---

[7]The analogy between $D_X y$ and $\mathbb{D}_X \varphi$ is made more precise in [6]: introducing natural Boolean variables $?\varphi$, with value 1 at any assignment $s \in A$ satisfying φ (and 0 at all others), validates the equivalence $\mathbb{D}_X \varphi \leftrightarrow (\varphi \wedge D_X ?\varphi)$. This can be turned into a definition, thus unifying the two types of LFD dependence statement.

[8]One can indeed go the other way around, but via a more complicated formula. Let $\top_X$ be an abbreviation for any tautology whose free variables are exactly the ones in X. Then $\mathbb{D}_X \varphi$ is equivalent to $\forall_{Free(\varphi)-X}(\varphi \wedge \top_X)$. As for non-local CRS quantifiers, they are equally expressive to the dependence modalities when V is *finite*, via the equivalences $\forall X \varphi \leftrightarrow \mathbb{D}_{V-X} \varphi$ and $\mathbb{D}_X \varphi \leftrightarrow \forall(V - X)\varphi$. When V is infinite, the two notions seem to be independent of each other (at least with our syntax, allowing for quantifiers only over *finite* sets of variables).

[9]See the footnote to Example 3.8 for a counterexample to Distribution for $\forall_X$. The deeper reason for this difference is that, as we will see, the FOL and CRS quantifiers, as well as the dependence modalities, are in fact normal *relational modalities*, quantifying over assignments that are accessible via some accessibility relation ($=_{V-X}$ or $=_X$), while the local quantifiers are not modalities of this kind.

dependence, it retains all classical Boolean operators with their standard laws; thus demonstrating that dependence is not an intrinsically non-classical phenomenon. Neither is basic reasoning about dependence necessarily complex, LFD is simple and well-behaved, with transparent axiomatizations and good meta-properties: decidability, forms of the finite model property, compactness, strong interpolation, and a form of cut elimination. Of course, this does not come for free. As always in logic, system design involves a balance between expressive power and other nice system properties. The more expressive the language, the more complex the validities – or stated conversely, the more well-behaved the logic, the less expressive the language. On the minimal basis language of LFD, however, one can analyze just which additional features in modeling dependence (and independence) force greater complexity for a logical system. Moreover, the *modal* flavor of LFD brings interesting connections with epistemic logics [33, 32, 6], interrogative and inquisitive logics [20, 26, 25], and situation-theoretic logics of informational correlations, [19]. Finally, as we shall show, LFD offers a platform for studying concrete notions of dependence in many fields in a way that imports only a minimum of logical complexity.

1.4   Structure of this article

Section 2 defines our models, giving a structural characterization of dependence. Section 3 introduces the logic LFD, together with a translation into FOL, a discussion of the differences between LFD quantifiers and the classical ones, and an equivalent modal relational semantics. The tandem of first-order and modal views will recur throughout the paper. Section 4 proves the decidability of LFD using object-free 'type models', while Appendix A has proofs of decidability and completeness using standard modal techniques. Section 5 presents a Hilbert-style axiomatization and a sequent calculus admitting a form of cut elimination, as well as interpolation and Beth definability results (with proofs in Appendix B). Section 6 explores extensions of LFD, including function terms, identity, independence, informational correlation, and dynamic modalities over changing dependence models. Section 7 looks at richer settings for dependence: including vector spaces, topological models, and dynamical systems. Section 8 draws comparisons with other approaches, including some discussion of their expressive surplus over LFD and questions raised by this. Conclusions and further prospects are found in Section 9.

2   State spaces, dependence graphs, functions

The starting point of this paper are the basic properties of semantic dependence relations, which will be determined here. Also a natural duality will emerge with explicit functional definitions for dependence, as well as appealing connections with consequence relations.

2.1   Dependence models

Throughout this paper, we assume given a set of *variables* V and a relational vocabulary $(Pred, ar)$, where $Pred$ is a set of *predicate symbols* and $ar : Pred \to N$ is an *arity* map, associating to each predicate $P \in Pred$ a natural number $ar(P)$.

Definition 2.1 (Dependence models, agreement, local dependences). *A dependence model $\mathbf{M}$ is a pair $\mathbf{M} = (M, A)$ of a (relational) FOL model $M = (O, I)$ with a domain O of* objects *and interpretation map I (sending each predicate symbol $P \in Pred$ of arity n into a set $I(P) \subseteq O^n$ of n-tuples of objects), together with a set $A \subseteq O^V$ of* admissible assignments *of objects to variables.*

6

A dependence model is full *if all possible assignments are admissible, i.e., if $A = O^V$. For assignments $s \in A$ and sets $X \subseteq V$, we put $s{\upharpoonright}X$ for the* restriction *of s to domain X.*

Definition 2.2 (**Agreement, local dependence, atoms**). *In dependence models, we define three basic relations: (a) for each set $X \subseteq V$ of variables, an* agreement relation *$s =_X t$ on assignments $s, t \in A$, (b) for each $s \in A$, a* local dependence *relation $D_X^s y$ between sets $X \subseteq V$ and variables $y \in V$, and (c) for each n-ary predicate P and each assignment $s \in A$, an n-ary relation $P^s \subseteq V^n$ on variables (where we use the notation $=_y$ for $=_{\{y\}}$):*

$$s =_X t \quad iff \quad s{\upharpoonright}X = t{\upharpoonright}X,$$

$$D_X^s y \quad iff \quad s =_X t \text{ implies } s =_y t \text{ for all } t \in A,$$

$$P^s x_1 \ldots x_n \quad iff \quad I(P)(s(x_1), \ldots, s(x_n)) \text{ holds,}$$

$\cdot$ *If $s =_X t$, s and t are said to* agree on X, *and if $D_X^s y$ we say that y* locally depends on X *at s. For any $Y \subseteq V$, we write $D_X^s Y$ if $D_X^s y$ holds for* all *$y \in Y$. Finally, we skip the set brackets for singletons, writing $D_x^s Y$ for $D_{\{x\}}^s Y$, and $D_x^s y$ for $D_{\{x\}}^s \{y\}$.*

Definition 2.3 (**Global dependence**). *The* global dependence *relation $D^{\mathbf{M}} \subseteq \mathcal{P}(V) \times V$ quantifies over all assignments in A: y depends on X in $\mathbf{M}$, written $D_X^{\mathbf{M}} y$, if $D_X^s y$ holds locally at all assignments $s \in A$. As for local dependence, this notation is extended to sets $Y \subseteq V$, by writing $D_X^{\mathbf{M}} Y$ if $D_X^{\mathbf{M}} y$ holds for* all *$y \in Y$; and again, set brackets are skipped for singletons. When the context is clear, superscripts $\mathbf{M}$ for current models will be dropped.*

Note that our global dependence statement $D_X^{\mathbf{M}} y$ matches the semantic clause for the so-called *dependence atom $= (X; y)$* introduced in Väänänen's Dependence Logic [71], when interpreted on the 'team' A of all admissible assignments.[10]

2.2 Dependence graphs

The basic structural properties of dependence relations are as follows.

Definition 2.4. *Let $R \subseteq \mathcal{P}(V) \times V$ be a relation between sets of variables and variables. Using the same conventions as for the dependence relation D above (writing $R_X y$ instead of $(X, y) \in R$, $R_X Y$ as an abbreviation for $\bigwedge_{y \in Y} R_X y$, and skipping set brackets for singletons), we say that:*

- *R satisfies* Reflexivity *if $R_x x$ holds for all $x \in V$*
- *R satisfies* Transitivity *if $R_X Y$ and $R_Y Z$ imply $R_X Z$*
- *R satisfies* Monotonicity *if $R_X y$ and $X \subseteq Z$ imply $R_Z y$*
- *R satisfies the* Projection *property if $R_X x$ holds for all $x \in X$*
- *R satisfies the* Inclusion *property if $R_X Y$ holds for all $Y \subseteq X$*
- *A variable $y \in V$ is an* R-constant *iff $R_\emptyset y$ holds.*

The following is easy to see:

---

[10]More generally, structures resembling our dependence models occur in areas such as epistemic logic, [6], temporal logic, [65], and situation theory, [11], [15].

Fact 2.5. *If $R \subseteq \mathcal{P}(V) \times V$ satisfies Transitivity, then the following are equivalent:*

1. *R satisfies Reflexivity and Monotonicity;*

2. *R satisfies the Projection property;*

3. *R satisfies the Inclusion property.*

It is well-known that the combination of Reflexivity, Transitivity and Monotonicity provides a characterization of classical logical consequence, cf. [69]. The following two results show that the same three properties characterize the relation of (local and global) dependence:[11]

Fact 2.6. *For every dependence model $\mathbf{M} = (M, A)$ and assignment $s \in A$, both global dependence $D_X^{\mathbf{M}} y$ and local dependence $D_X^s y$ satisfy Reflexivity, Transitivity and Monotonicity. For both relations $R \in \{D^{\mathbf{M}}, D^s\}$, the R-constants are exactly the variables $y \in V$ whose value is the same for every assignment in A.*

Fact 2.5 follows immediately from Definitions 2.3 and 2.2. The converse takes more work:

Proposition 2.7.   1. *For every relation $R \subseteq \mathcal{P}(V) \times V$ satisfying Reflexivity, Transitivity and Monotonicity, there is a dependence model $\mathbf{M}$ whose global dependence relation $D^{\mathbf{M}}$ coincides with R. Moreover, if V is finite, then $\mathbf{M}$ can be taken to be finite as well, of size bounded by $2^{|V|}$.*

2. *For every relation $R \subseteq \mathcal{P}(V) \times V$ satisfying Reflexivity, Transitivity and Monotonicity, there is a dependence model $\mathbf{M}_R$ where R coincides with all the local dependence relations D^s (at all assignments $s \in A$), and hence it also coincides with the global dependence $D^{\mathbf{M}}$. Moreover, if V is finite, then $\mathbf{M}_R$ can be taken to be finite as well, of size bounded by $2^{2^{|V|}}$.*

3. *Let $\mathcal{R} \subseteq \mathcal{P}(V) \times V$ be a family of relations satisfying Reflexivity, Transitivity and Monotonicity, s.t. all relations "agree on constants" (i.e., $R_\emptyset y$ iff $R'_\emptyset y$, for all $y \in V$ and $R, R' \in \mathcal{R}$). Then $\mathcal{R}$ coincides with the family $\{D^s : s \in A\}$ of all local dependence relations of some dependence model $\mathbf{M}_\mathcal{R}$. Moreover, if V is finite then $\mathbf{M}_\mathcal{R}$ can be taken to be finite.*

Proof. For a start, we need some preliminary notations and results. Let $R \subseteq \mathcal{P}(V) \times V$ be a relation satisfying Reflexivity, Transitivity and Monotonicity. A subset $X \subseteq V$ is *R-closed* if we have $y \in X$ for all $y \in V$ satisfying $R_X y$. Let Γ be the family of all R-closed subsets of V. Note that Γ is closed under arbitrary intersections[12]. [Note: for greater readability in what follows, we have put the simple proof of this and some later auxiliary statements in footnotes.] Also, it is immediate that the family Γ contains the set V of all variables. We put $\tilde{X} := \{y \in V : R_X y\}$ for *the R-closure of X*, which is the least R-closed set s.t. $X \subseteq \tilde{X}$.[13]

---

[11]However, in our formal axiomatizations in Section 5, we will use the equivalent combination of Projection and Transitivity, cf. Fact 2.5.

[12]Let $\{X^i : i \in I\} \subseteq \Gamma$ be a family of R-closed sets, with $X := \bigcap_{i \in I} X^i$. To show that X is R-closed, let $R_X y$ for some $y \in V$. By Monotonicity, $R_{X^i} y$ for all $i \in I$, and so by R-closure, $y \in X^i$ for all $i \in I$, i.e. $y \in X$.

[13]$X \subseteq \tilde{X}$ follows from the fact that $R_X x$ holds for all $x \in X$, by Reflexivity and Monotonicity. To see that $\tilde{X}$ is R-closed, let $z \in V$ be s.t. $R_{\tilde{X}} z$. This, together with the fact that $R_X \tilde{X}$ (by the definition of $\tilde{X}$) yields $R_X z$ (by Transitivity), i.e., $z \in \tilde{X}$. Finally, if Y is any R-closed set with $X \subseteq Y$, we show that $\tilde{X} \subseteq Y$. Let $y \in \tilde{X}$, i.e., $R_X y$. Then $R_Y y$ (by Monotonicity and $X \subseteq Y$), and therefore $y \in Y$ (by the R-closure of Y).

Proof of Part 1. Let $R \subseteq \mathcal{P}(V) \times V$ satisfy Reflexivity, Transitivity and Monotonicity. Consider the model $\mathbf{M} = (O, I, A)$ with (a) $O = V \cup \mathcal{P}(V)$, (b) the interpretation map I makes all predicates *false*, and (c) the family $A = \{s_X : X \in \Gamma\}$ consists of assignments s_X, one for each R-closed set X, with

$$s_X(x) = x \text{ if } x \in X, \text{ and } s_X(x) = X \text{ if } x \notin X.$$

Note that $|A| = |\Gamma| \leq |\mathcal{P}(V)| = 2^{|V|}$. The model $\mathbf{M}$ validates the following two claims, for all $Y, Z \in \Gamma$ and $U \subseteq V$:

(a) $s_Y =_U s_Z$ iff either $Y = Z$ or $U \subseteq Y \cap Z$.

(b) $D_X^{\mathbf{M}} y$ holds iff $R_X y$ holds.

Claim (a): This follows from the definition of the assignments s_X via the following sequence of equivalences: $s_Y =_U s_Z$ iff $s_Y(x) = s_Z(x)$ for all $x \in U$ iff either $Y = Z$ or $s_Y(x) = s_Z(x) = x$ for all $x \in U$ iff either $Y = Z$ or $x \in Y \cap Z$ for all $x \in U$.

Claim (b): From left to right, let $D_X^{\mathbf{M}} y$, and consider the assignments $s_{\tilde{X}}$ and s_V, with V the set of all variables. Note that $s_V(y) = y$ for all variables y. By (a), we have $s_V =_X s_{\tilde{X}}$ (since $X \subseteq \tilde{X} = V \cap \tilde{X}$). Therefore, since $D_X^{\mathbf{M}} y$, $s_V =_y s_{\tilde{X}}$, and this means by the definition of the two assignments that $s_V(y) = s_{\tilde{X}}(y) = y$. In particular, then, $y \in \tilde{X}$, i.e., $R_X y$.

From right to left, assume that $R_X y$. To show that $D_X^{\mathbf{M}} y$ holds, let $s_Z, s_U \in A$ (with $Z, U \in \Gamma$) be any two assignments with $s_Z =_X s_U$. By (a), $s_Z =_X s_U$ implies that either $Z = U$ or $X \subseteq Z \cap U$. In the first case, $Z = U$ immediately gives $s_Z =_y s_U$, as desired. In the second case, $R_X y$ and $X \subseteq Z \cap U$ imply $R_Z y$, $R_U y$ by Monotonicity, which means by the R-closure of Z, U that $y \in Z$, $y \in U$, By definition then $s_Z(y) = s_U(y) = y$.[14]

Proof of Part 2. Let $R \subseteq \mathcal{P}(V) \times V$ satisfy Reflexivity, Transitivity and Monotonicity. For each $x \in V$, we define a binary relation $\sim_x$ on *families* of R-closed sets $\mathcal{A}, \mathcal{B} \in \mathcal{P}(\Gamma)$, by putting:

$$\mathcal{A} \sim_x \mathcal{B} \text{ iff } x \in \bigcap(\mathcal{A} \triangle \mathcal{B}),$$

where $\mathcal{A} \triangle \mathcal{B} := (\mathcal{A} - \mathcal{B}) \cup (\mathcal{B} - \mathcal{A})$ is the symmetric difference of the two families.

It is easy to check that each $\sim_x$ is an equivalence relation.[15] For any family $\mathcal{A} \subseteq \Gamma$ and variable $x \in V$, we denote by $[\mathcal{A}]_x$ the equivalence class of $\mathcal{A}$ modulo $\sim_x$.

We construct now a model $\mathbf{M}_R = (O_R, I_R, A_R)$ with: $O_R = \{[\mathcal{A}]_x : \mathcal{A} \subseteq \Gamma, x \in V\}$, i.e., all the equivalence classes modulo all the relations $\sim_x$, the interpretation I_R makes all predicates *false*; and $A_R = \{s^{\mathcal{A}} : \mathcal{A} \subseteq \Gamma\}$ consists of assignments $s^{\mathcal{A}}$ with $s^{\mathcal{A}}(x) := [\mathcal{A}]_x$ for all $x \in V$. Note that, if V is finite, then $\mathbf{M}_R$ is finite as well, and in fact $|A_R| \leq |\mathcal{P}(\Gamma)| \leq |\mathcal{P}(\mathcal{P}(V))| = 2^{2^{|V|}}$.

This model validates the following claims, for all $s^{\mathcal{A}}, s^{\mathcal{B}} \in A_R$ and $U \subseteq V$:

(a) $s^{\mathcal{A}} =_U s^{\mathcal{B}}$ iff $U \subseteq \bigcap(\mathcal{A} \triangle \mathcal{B})$.

(b) $D_X^{s^{\mathcal{A}}} y$ holds in $\mathbf{M}_R$ iff $R_X y$.

---

[14]Closer inspection of this argument shows that the local dependence relation at the special assignment s_V, with V the set of all variables, actually equals the given relation R.

[15]Reflexivity follows since $\bigcap(\mathcal{A} \triangle \mathcal{A}) = \bigcap \emptyset = V$. Symmetry follows from the commutativity of symmetric difference. Transitivity follows from the fact that $\mathcal{A} \triangle \mathcal{C} \subseteq (\mathcal{A} \triangle \mathcal{B}) \cup (\mathcal{B} \triangle \mathcal{C})$, which implies that $\bigcap(\mathcal{A} \triangle \mathcal{C}) \supseteq \bigcap(\mathcal{A} \triangle \mathcal{B}) \cap \bigcap(\mathcal{B} \triangle \mathcal{C})$. If $\mathcal{A} \sim_x \mathcal{B} \sim_x \mathcal{C}$, then $x \in \bigcap(\mathcal{A} \triangle \mathcal{B}) \cap \bigcap(\mathcal{B} \triangle \mathcal{C})$, hence $x \in \bigcap(\mathcal{A} \triangle \mathcal{C})$, i.e. $\mathcal{A} \sim_x \mathcal{C}$.

Claim (a): This follows directly from the definition of the assignments $s^{\mathcal{A}}$, via the following sequence of equivalences: $s^{\mathcal{A}} =_U s^{\mathcal{B}}$ iff $[\mathcal{A}]_x = [\mathcal{B}]_x$ for all $x \in U$ iff $\mathcal{A} \sim_x \mathcal{B}$ for all $x \in U$ iff $x \in \bigcap(\mathcal{A} \triangle \mathcal{B})$ for all $x \in U$.

Claim (b): From left to right, suppose that $D_X^{s^{\mathcal{A}}} y$ holds in $\mathbf{M}_R$. Take the family $\mathcal{B} := \mathcal{A} \cup \{\tilde{X}\}$. Case (i): $\tilde{X} \notin \mathcal{A}$. Then we have $\mathcal{A} \triangle \mathcal{B} = \{\tilde{X}\}$, hence $\bigcap(\mathcal{A} \triangle \mathcal{B}) = \tilde{X}$. Thus, by (a) we have $s^{\mathcal{A}} =_X s^{\mathcal{B}}$ (since $X \subseteq \tilde{X}$). It follows by the truth of $D_X^s y$ at $s^{\mathcal{A}}$ that $s^{\mathcal{A}} =_y s^{\mathcal{B}}$. But this means by the already proved equivalence (a) that $y \in \tilde{X}$, i.e., $R_X y$. Case (ii): $\tilde{X} \in \mathcal{A}$. Repeat the preceding argument, but now w.r.t. the families $\mathcal{A}$ and $\mathcal{A} - \{\tilde{X}\}$.

From right to left, assume that $R_X y$, and consider any assignment $s^{\mathcal{A}} \in A_R$. Let $s^{\mathcal{B}} \in A_R$ be any admissible assignment s.t. $s^{\mathcal{A}} =_X s^{\mathcal{B}}$. By claim (a), $X \subseteq \bigcap(\mathcal{A} \triangle \mathcal{B})$. Putting this together with $R_X y$, we obtain by Monotonicity that $R_{\bigcap(\mathcal{A} \triangle \mathcal{B})} y$. But $\bigcap(\mathcal{A} \triangle \mathcal{B})$ is R-closed (being the intersection of a family of R-closed sets), and therefore, $y \in \bigcap(\mathcal{A} \triangle \mathcal{B})$. Applying claim (a) again, we conclude that $s^{\mathcal{A}} =_y s^{\mathcal{B}}$. Thus $s^{\mathcal{A}}$ satisfies $D_X^s y$, as desired.

The desired conclusion follows immediately from the second claim.[16]

Proof of Part 3. Let $\mathcal{R}$ be a family of binary relations on V satisfying Reflexivity, Transitivity and Monotonicity, and agreeing on constants. For each $R \in \mathcal{R}$, put $C = \{y \in R : R_{\emptyset} y\} \subseteq V$ for the common set of R-constants. Construct all the models $\mathbf{M}_R$ as in Part 2, for every $R \in \mathcal{R}$. Then each R is both the local and the global dependence in the corresponding $\mathbf{M}_R$.

The only remaining step for our main proof involves the following general disjoint union construction on dependence models. Define a new model $\mathbf{M}_{\mathcal{R}} = (O_{\mathcal{R}}, I_{\mathcal{R}}, A_{\mathcal{R}})$, where (a) $O_{\mathcal{R}} := C + \sum_{R \in \mathcal{R}} O_R = C \cup \bigcup_{R \in \mathcal{R}} \{R\} \times O_R$ is the *disjoint union* of the common set of constants and all sets of objects of the models $\mathbf{M}_R$, (b) the interpretation $I_{\mathcal{R}}$ makes all predicates *false*, and (c) $A_{\mathcal{R}} := \{s_R : R \in \mathcal{R}, s \in A_R\}$ consists of new assignments s_R, each associated to an old assignment $s \in A_R$ with $R \in \mathcal{R}$, with

$$s_R(x) := (R, s(x)) \in \{R\} \times O_R \text{ for } x \in V - C, \text{ and } s_R(x) := x \text{ for } x \in C.$$

Note that in this model, $s_R =_X s'_{R'}$ for all $X \subseteq C$ and all $s_R, s'_{R'} \in A_{\mathcal{R}}$. Also, for $X \not\subseteq C$, $s_R =_X s'_{R'}$ holds in $\mathbf{M}_{\mathcal{R}}$ iff $R = R'$ and $s =_X s'$ holds in $\mathbf{M}_R$. Using these facts, it is easy to see that the local dependence statement $D_X^w y$ holds in $\mathbf{M}_{\mathcal{R}}$ at a state $w = s_R \in A_{\mathcal{R}}$ iff it holds at s in the corresponding component $\mathbf{M}_R$, and so the global dependence statement $D_X^{\mathbf{M}} y$ holds in $\mathbf{M}_{\mathcal{R}}$ iff it holds in all components. It follows that $\mathcal{R}$ coincides with the family of all local dependence relations within $\mathbf{M}_{\mathcal{R}}$, and that $\bigcap \mathcal{R}$ is the global dependence relation on $\mathbf{M}_{\mathcal{R}}$.[17] □

The preceding representation method uses a large number of objects in general. What happens when one restricts the available objects that can be assigned to variables?

Example 2.8. Consider a dependence model given by the table below:

x	y	z
0	*1*	*0*
1	*1*	*0*
2	*0*	*0*

---

[16]There may be a way of proving Part 2 using some general product construction on the simpler models produced by in the proof for Part 1, but we have not yet been able to find one.

[17]This representation argument can be turned into a proof of completeness and finite model property for a simple logic of dependence atoms plus the universal modality over available assignments, a precursor to the completeness proof for the richer language of LFD in Section 5.

This table uses three values to represent a *strict* linear dependence order of three variables x, y, z: we have global dependencies $D_x y$ and $D_y z$, but not the other way around. But as is easy to see, this cannot be done with only two objects.[18] To state the underlying observation positively, the following is *valid on two-valued models*:

if $D_x y$ and $D_y z$, then $D_y x$ or $D_z y$.

More generally, the following can be shown:

Arbitrarily high finite numbers of values are needed to represent arbitrary finite linear orders.

What are minimal sets of objects for representing given dependence graphs? How can one axiomatize the structural dependence properties for each fixed finite set of objects?

Remark 2.9 (Dependence and consequence). As already mentioned, the three stated structural properties (Reflexivity, Transitivity and Monotonicity) are known to be characteristic for the relation of classical *logical consequence*, [69]. But the preceding observations show one essential difference. To represent a three-element linear sequence of variables ordered by dependence, three objects were needed in Example 2.8. But to represent an analogous sequence of strict consequences, only two truth values are needed, e.g.:

p	q	r
0	1	1
0	0	1

In fact, any finite acyclic graph can be represented in terms of classical logical consequence.

All this suggests a move to non-classical consequence relations without a simple truth value semantics. In fact, the format $D_X y$, with multiple 'premises' in X and a single 'conclusion' y, resembles Gentzen-style sequents for intuitionistic logic, and dependence has been related to intuitionistic implication, [2]. Also, given the analogies between dependence and implication between questions to be discussed in Section 3.5, dependence has been related to notions of implication in interrogative and inquisitive logics, [6], [25].

The analogy between dependence and consequence can also be extended in other ways. For instance, adopting a classical sequent format, one can study dependencies $D_X Y$ read disjunctively in the set Y. Or, softening the strict universal quantification over assignments in our semantic notion, one obtains new non-monotonic varieties of dependence where the dependence only holds 'under normal circumstances', by analogy with non-monotonic logics, [23].

2.3 Explicit function definitions

Our semantic definition makes dependence $D_X y$ a form of implicit definability, as fixing the values of the dependent variable y by fixing the values of the variables in X. But there is also a broad alternative intuition of dependence, viz. as y being *definable* in terms of X using some repertoire of available operations.[19] The two views are connected. In mathematics, implicit semantic definability justifies the explicit introduction of a corresponding function.

This discussion suggests the following general line.

---

[18]One can fill in the table for y and z to get the right failures of dependence, but then, by functionality, a third value must be assigned somewhere to x.

[19]Explicit functional dependence relies crucially on the available operations. For instance, in the set $\{1, 2\}$, 2 depends on 1 if addition is present, but not if the only operation is multiplication. Another typical example are dependent vectors that are special *linear combinations* of other vectors.

Definition 2.10. *Given a dependence model* $\mathbf{M}$, *a set* $X \subseteq V$ *of variables and a variable* $y \in V$, *let* F_X^y *be the partial function from* X-*indexed tuples in* O^X *to objects in* O, *satisfying for all tuples* $\mathbf{u} \in O^X$: $F_X^y(\mathbf{u}) = o$ *iff* o *is the unique object in* O *s.t.* $s(y) = o$ *holds for some* $s \in A$ *with* $s{\restriction}X = \mathbf{u}$; *if no such object exists,* $F_X^y(\mathbf{u})$ *is undefined. In other words:* $F_X^y(\mathbf{u}) = o$ *holds iff we have both (1)* $\mathbf{u} = s{\restriction}X$ *for some assignment* $s \in A$, *and (2) for all assignments* $t, t' \in A$, $t{\restriction}X = t'{\restriction}X$ *implies* $t(y) = t'(y) = o$. *We denote by* $dom(F_X^y)$ *the domain of this function. The expansion of* $\mathbf{M}$ *with all these partial functions* F_X^y *is called the* induced function model $F(\mathbf{M})$.

The partial functions introduced in this Skolemization-like manner make explicit the functions that underlie local and global dependencies:

Fact 2.11. *Induced function models satisfy the following two equivalences:*

$$D_X^s y \quad iff \quad s{\restriction}X \in dom(F_X^y) \quad iff \quad s(y) = F_X^y(s{\restriction}X)$$
$$D_X^{\mathbf{M}} y \quad iff \quad dom(F_X^y) = \{s{\restriction}X : s \in A\} \quad iff \quad s(y) = F_X^y(s{\restriction}X) \text{ for all } s \in A.$$

Explicit definability is a natural companion to our semantic view as implicit determination.[20] Developing an abstract purely operational approach matching our semantic view of dependence may be worthwhile, and some concrete instances of how such an approach might work can be found in Section 7.2 on the notion of linear dependence in vector spaces.

3 The logic of functional dependence

We now introduce the language of our logic LFD of functional dependence.

3.1 Syntax and semantics of LFD

Definition 3.1. *Given a vocabulary* $(V, Pred, ar)$, *the* language LFD *is recursively given by:*

$$\varphi ::= \quad P\mathbf{x} \quad | \quad \neg\varphi \quad | \quad \varphi \wedge \varphi \quad | \quad \mathbb{D}_X\varphi \quad | \quad D_X y$$

where $y \in V$ *is any variable,* P *is any predicate symbol,* $\mathbf{x} = (x_1 \ldots x_n)$ *is a finite string of variables of length* $n = ar(P)$ *and* $X \subseteq V$ *is a finite set of variables.*

Definition 3.2. (*Semantics*) *Truth of a formula* φ *in a dependence model* $\mathbf{M} = (M, A)$ *at an assignment* $s \in A$ (*written* $\mathbf{M}, s \models \varphi$, *with the index* $\mathbf{M}$ *dropped when the model is understood*) *is defined by the following clauses:*

$$
\begin{aligned}
s &\models P\mathbf{x} & iff \quad & I^{\mathbf{M}}(P) \text{ holds for the tuple } s(\mathbf{x}) \\
s &\models \neg\varphi & iff \quad & s \not\models \varphi \\
s &\models \varphi \wedge \psi & iff \quad & s \models \varphi \text{ and } s \models \psi \\
s &\models \mathbb{D}_X\varphi & iff \quad & t \models \varphi \text{ holds for all } t \in A \text{ with } s =_X t \\
s &\models D_X y & iff \quad & D_X^s y \text{ holds in } \mathbf{M}.
\end{aligned}
$$

---

[20]An operational view may also underlie dependence notions in logic. Dependence in quantifier combinations $\forall x \exists y$ means that a value for y can be produced given one for x, something that can be made concrete by a Skolem function. And dependence in a set of formulas like $\{p, p \vee q\}$ may mean that some proof 'produces' $p \vee q$ from p.

As announced in the Introduction, the *dependence modality* $\mathbb{D}_X\varphi$ is read as "X locally determines φ": *the current values of X determine the truth of φ*. Similarly, $D_X y$ is read as "X locally determines y": it says that *the current values of X determine the value of y*.[21]

One important notion in LFD is that of *free variables*. Here we have to be careful. As in FOL, we want the free variables to be those whose current values *determine the truth value of a formula*; indeed, binding a variable is a way of "forgetting" its value as irrelevant, while the specific value currently assigned to a free variable *is relevant* for the meaning of the formula. But the definition is subtly different from FOL, since the dual quantifiers $\mathbb{D}_X\varphi$ explicitly list the variables that are left *free* (rather than listing the bound ones, as do the usual quantifiers).

Definition 3.3. Free(φ) *is defined by the following recursion:*

(a) Free$(Px_1 \ldots x_n) = \{x_1, \ldots, x_n\}$,

(b) Free$(\neg\varphi)$ = Free(φ), (c) Free$(\varphi \wedge \psi)$ = Free$(\varphi) \cup$ Free(ψ),

(d) Free$(\mathbb{D}_X\varphi) = X$, (e) Free$(D_X y) = X$.

Now we can check that indeed the values of the free variables occurring in a formula uniquely determine its truth value. In other words, like FOL (and unlike CRS), LFD is 'local':

Fact 3.4. (*Locality.*) *If* Free$(\varphi) \subseteq X$, *and* $s =_X t$, *then* $s \models \varphi$ *iff* $t \models \varphi$.

Proof. The proof is by induction on φ. The atomic and Boolean cases are entirely straightforward. *Dependence modalities*: Assume that $s =_X t$ and $s \models \mathbb{D}_X\varphi$. To show that $t \models \mathbb{D}_X\varphi$, consider any $w \in A$ with $t =_X w$. Here $s =_X t$ and $t =_X w$ imply $s =_X w$. This together with $\mathbb{D}_X\varphi$ then yields $w \models \varphi$ as desired, by the semantics of $\mathbb{D}_X$. *Dependence atoms*: Assume that $s =_X t$ and $s \models D_X y$. Then $s =_y t$ by the semantics of D_X. To show that $t \models D_X y$, consider any $w \in A$ with $t =_X w$. Here $s =_X t$ and $t =_X w$ imply $s =_X w$, which together with $s \models D_X y$ gives that $s =_y w$. From this and $s =_y t$, it follows that $t =_y w$ as desired. $\square$

Here is a useful immediate consequence in the presence of additional local dependencies.

Corollary 3.5. *If two assignments s, t agree on all formulas with free variables in X, then they also agree on all formulas with free variables in the extended set $\{y \mid s \models D_X y\}$.*[22]

Abbreviations. Boolean connectives $\top, \bot, \varphi \vee \psi, \varphi \rightarrow \psi, \varphi \leftrightarrow \psi$ are defined as usual. We use $D_x y$ for $D_X y$ when $X = \{x\}$ is a singleton, and same for $\mathbb{D}_x\varphi$. Other abbreviations are:

(a) $\mathsf{A}\varphi := \mathbb{D}_\emptyset\varphi$ (universal modality)   (b) $\mathbb{E}_X\varphi := \neg\mathbb{D}_X\neg\varphi$ (c) $\mathsf{E}\varphi := \neg\mathsf{A}\neg\varphi$

(d) $Cy := D_\emptyset y$ (constant value)   (e) $= (X; y) := \mathsf{A}D_X y$ (global dependence)

(f) $D_X Y := \bigwedge_{y \in Y} D_X y$ (multi-variable dependence)

(h) $\forall_X\varphi := \mathbb{D}_{Free(\varphi)-X}\varphi$ (universal quantifier)

(i) $\exists_X\varphi := \neg\forall_X\neg\varphi$ (existential quantifier)

---

[21] As already mentioned in the Introduction, the obvious analogy between $\mathbb{D}_X\varphi$ and $D_X y$ can be made precise by introducing Boolean variables $?\varphi$ as in [6], that record the truth-values of formulas φ, and then *defining* the dependence quantifiers as $\mathbb{D}_X\varphi := \varphi \wedge D_X ?\varphi$. Here, we choose to take these quantifiers as primitive, as they have an independent logical motivation and in our view they play an equally important role as the dependence atoms in the study of (partial) dependencies and correlations between variables.

[22] The set $\{y \mid s \models D_X y\}$ includes X, as a consequence of Reflexivity and Monotonicity.

These defined connectives behave as expected. E.g., syntactically, we have that: $Free(\forall_X \varphi) = Free(\exists_X \varphi) = Free(\varphi) - X$. Semantically, e.g., $\mathsf{A}\varphi$ means that *all* assignments satisfy φ, etc.:

$$
\begin{aligned}
s &\models \mathsf{A}\varphi &&\text{iff} &&t \models \varphi \text{ for all } t \in A \\
s &\models \mathbb{E}_X \varphi &&\text{iff} &&t \models \varphi \text{ for some } t \in A \text{ with } s =_X t \\
s &\models Cy &&\text{iff} &&t =_y s \text{ for all } t \in A \\
s &\models = (X;y) &&\text{iff} &&D_X^{\mathbf{M}} y \text{ holds} \\
s &\models \forall_X \varphi &&\text{iff} &&t \models \varphi \text{ for all } t \in A \text{ with } s =_{Free(\varphi)-X} t.
\end{aligned}
$$

Note that our defined formula $\forall_X \varphi$ matches the semantics of the 'local' version of the universal quantifier (in the sense of satisfying the Locality principle from Fact 3.4), as given in the Introduction. Recall that these amount to the standard FOL quantifiers on full models, and are their closest analogue on arbitrary dependence models. Thus, LFD *contains the first-order quantifiers*, generalized from their standard models to the larger realm of dependence models. For further discussion of the meaning of LFD quantifiers, cf. Section 3.2.

Remark 3.6 (Informational interpretation). The set A of admissible assignments in a dependence model is a 'complete database', as in Example 1.1, and can be interpreted as an *information structure*, encoding the 'knowledge base' of an (anonymous) agent: a full list of all the tuples that are consistent with the agent's background information. The underlying assumption is that only one tuple (the 'current assignment') represents the actual state of the world, but that tuple is typically unknown: the agent can only narrow down the possibilities to the set A. The universal modality $\mathsf{A}\varphi := \mathbb{D}_\emptyset \varphi$ then captures the agent's *information*: $\mathsf{A}\varphi$ means that φ *is 'known'*. The constant-value formula $Cx := D_\emptyset x$ says that the value of x is 'known'. Dependence quantifiers capture a form of *conditional knowledge*: $\mathbb{D}_X \varphi$ means that the agent can know that φ if she is given the current values of X. Analogously, dependence atoms $D_X y$ express a form of *conditional knowledge of a value*: the agent can know the value of y if given the values of X. Finally, global dependence $\mathsf{A}D_X y$ captures *known correlations*: the agent knows how to determine the value of y from the values of X.

Remark 3.7 (Further notions of dependence). We can also formalize a common alternative intuition of dependence, [13], as 'changing x involves changing y': this is just $D_y x$. Moreover, weaker notions of dependence can be defined, such as 'restricting the value of x to property P restricts the value of y to have property Q' (cf. Remark 1.2). This is expressed by $\mathsf{A}(Px \rightarrow Qy)$. Yet another definable notion of dependence is that the current values of X restrict the value of y to have property Q, captured by the formula $\mathbb{D}_X Qy$.

Example 3.8. Here are some illustrations of valid and invalid consequences:

1. *(a) $\varphi \rightarrow \mathbb{D}_X \varphi$ is valid if $Free(\varphi) \subseteq X$. (b) $Px \rightarrow \mathbb{D}_y Px$ is not valid.*

2. *(a) $\mathbb{D}_{X \cap Y} \varphi \rightarrow \mathbb{D}_X \mathbb{D}_Y \varphi$ is valid. (b) $\mathbb{D}_X \mathbb{D}_Y \varphi \rightarrow \mathbb{D}_{X \cap Y} \varphi$ is not valid.*

3. *Let $X \cap Y = \emptyset$. (a) $(\mathbb{E}_X \varphi \wedge \mathbb{E}_Y \psi) \rightarrow \mathbb{E}_{X \cup Y}(\varphi \wedge \psi)$ is not valid. However, (b) if $Free(\varphi) \subseteq X$ and $Free(\psi) \subseteq Y$, the preceding implication is valid, and in fact we have the stronger validity $(\mathbb{E}_X \varphi \wedge \mathbb{E}_Y \psi) \rightarrow (\varphi \wedge \psi)$.*[23]

---

[23]To check this: assume that $s \models \mathbb{E}_X \varphi \wedge \mathbb{E}_Y \psi$. Then there are $s' =_X s$ and $s'' =_Y s$ with $s' \models \varphi$ and $s'' \models \psi$. Using these facts and the assumptions that $Free(\varphi) \subseteq X$ and $Free(\psi) \subseteq Y$, we obtain $s \models \varphi$ and $s \models \psi$ by Locality, Fact 3.4. It follows that $s \models \varphi \wedge \psi$.

4. *The Distribution axiom is sound for dual quantifiers:* $\mathbb{D}_X(\varphi \to \psi) \to (\mathbb{D}_X\varphi \to \mathbb{D}_X\psi)$ *is valid. However, (b) Distribution is not sound for local quantifiers:* $\forall_X(\varphi \to \psi) \to (\forall_X\varphi \to \forall_X\psi)$ *is* not *valid.*[24]

5. *(a)* $\mathbb{D}_X\varphi \to \varphi$ *and* $\forall_X\varphi \to \varphi$ *are valid. However, (b) the classical elimination rule for the universal quantifier is* not *sound:* $\forall_x\varphi \to [y/x]\varphi$ *is* not *valid.*[25]

The last non-validity is explained by the fact that in LFD variables are no longer arbitrary place-holders, but have an individual meaning, denoting specific quantities (as commonly done in the empirical sciences, where e.g. t stands for time, etc). This means that, unlike in FOL, bound alphabetic variants may have *different* truth values: $\forall_x Px$ can be true in a model while $\forall_y Py$ is false. On the other hand, LFD still allows for a formulation of the intuition behind bound variants that the choice of variables is arbitrary: though no longer holding inside one given model, the invariance under renaming still holds *across models.*

Fact 3.9. *(*Renaming Lemma*.) Consider any dependence model* $\mathbf{M} = (M, A)$ *and* LFD*-formula* φ. *Let* σ *be a permutation of all variables, and let* $\sigma(\varphi)$ *be the result of replacing in* φ *every occurrence of any variable* $x \in V$ *by* $\sigma(x)$. *Moreover, for every* $s \in A$, *let* s^σ *be the assignment given by putting* $s^\sigma(x) := s(\sigma^{-1}(x))$ *for all variables* x, *and let* $\mathbf{M}^\sigma = (M, A^\sigma)$ *be the dependence model obtained by taking* $A^\sigma = \{s^\sigma : s \in A\}$*). Then the following equivalence holds:*

$$\mathbf{M}, s \models \varphi \ \ iff \ \ \mathbf{M}^\sigma, s^\sigma \models \sigma(\varphi).$$

As a consequence, validity is invariant under variable renaming*:* φ *is valid iff* $\sigma(\varphi)$ *is valid.*

Proof. The proof of the first claim is by induction on φ.

Atomic formulas. Note that, for all $s \in A, \mathbf{x} \in V$ we have $s(\mathbf{x}) = s(\sigma^{-1}(\sigma(\mathbf{x})) = s^\sigma(\sigma(\mathbf{x}))$, and also $\sigma(P\mathbf{x}) = P\sigma(\mathbf{x})$. Using these and the truth definition, we obtain the equivalences: $\mathbf{M}, s \models P\mathbf{x}$ iff $s(\mathbf{x}) \in I(P)$ iff $s^\sigma(\sigma(\mathbf{x})) \in I(P)$ iff $\mathbf{M}, s^\sigma \models P\sigma(\mathbf{x})$ iff $\mathbf{M}, s^\sigma \models \sigma(P\mathbf{x})$.

Boolean combinations. This inductive step follows immediately by the truth clauses for Boolean operations, the induction hypothesis and the fact that permutations satisfy $\sigma(\neg\varphi) = \neg\sigma(\varphi)$ and $\sigma(\varphi \wedge \psi) = \sigma(\varphi) \wedge \sigma(\psi)$.

Dependence modalities. First note the following equivalence: $s =_X t$ iff $s^\sigma =_{\sigma(X)} t^\sigma$. Using this, the truth clause for the universal dependence modality and the induction hypothesis, we obtain the following sequence of equivalences: $\mathbf{M}, s \models \mathbb{D}_X\varphi$ iff $\forall t \in A\,(s =_X t \Rightarrow \mathbf{M}, t \models \varphi)$ iff $\forall t \in A\,(s^\sigma =_{\sigma(X)} t^\sigma \Rightarrow \mathbf{M}^\sigma, t^\sigma \models \sigma(\varphi))$ iff $\forall w \in A^\sigma\,(s^\sigma =_{\sigma(X)} w \Rightarrow \mathbf{M}^\sigma, w \models \sigma(\varphi))$ iff $\mathbf{M}^\sigma, s^\sigma \models \mathbb{D}_{\sigma(X)}\sigma(\varphi)$ iff $\mathbf{M}^\sigma, s^\sigma \models \sigma(\mathbb{D}_X\varphi)$.

Dependence atoms. Using the same observation as in the previous case, together with the fact that $\sigma(D_X y) = D_{\sigma(X)}\sigma(y)$, as well as truth clause for dependence atoms and the induction hypothesis, we obtain the sequence of equivalences: $\mathbf{M}, s \models D_X y$ iff $\forall t \in A\,(s =_X t \Rightarrow s =_y t)$ iff $\forall t^\sigma \in A^\sigma\,(s^\sigma =_{\sigma(X)} t^\sigma \Rightarrow s^\sigma =_{\sigma(y)} t^\sigma)$ iff $\forall w \in A^\sigma\,(s^\sigma =_{\sigma(X)} w \Rightarrow s^\sigma =_{\sigma(y)} w)$ iff $\mathbf{M}^\sigma, s^\sigma \models D_{\sigma(X)}\sigma(y)$ iff $\mathbf{M}^\sigma, s^\sigma \models \sigma(D_X y)$.

Finally, the second claim follows immediately from the first, by quantifying over both admissible assignments and dependence models. $\square$

---

[24]A counterexample, for $V = \{x, y, z\}$, is given by taking $O = \{0, 1\}$, $I(P) = \{(1, 1)\}$ and $A = \{(1, 1, 1), (1, 1, 0), (0, 0, 1)\}$, where a triplet of values denotes the corresponding assignment on (x, y, z). The assignment $(1, 1, 1)$ satisfies $\forall_x(Pxy \to Pxz)$ and $\forall_x Pxy$, but fails to satisfy $\forall_x Pxz$.

[25]Here, $[y/x]\varphi$ is the result of substituting y for x in the formula φ.

3.2 Discussion: Quantification over objects in LFD

Having entered the world of LFD with its special behavior of variables, one might ask whether the above quantifier companions $\forall_X \varphi$ of the dependence modalities are 'true' quantifiers. This question calls for some distinctions. First, as we saw earlier, both the CRS-style quantifiers $\forall X \varphi$ and their local versions $\forall_X \varphi$ are simply generalizations of the FOL quantifiers to a broader class of models, and the dependence modalities of LFD are their close duals. However, one might require a true quantifier to be a semantic operator, acting on *objects*, so that the precise variable used in its syntax does not matter.

Now in one sense, this is true in LFD: variable names do not matter when we look *across models*. What can be said in one model using x can be said in another model using another variable y, by the Renaming Principle 3.9, which underpins, for instance, the use of alphabetic variants for proofs in our axiomatic systems of Sections 5.1, 6.2. But *locally* within one model, the existence of non-trivial dependencies gives rise to *asymmetries* of behavior between variables: as already observed, variables in a fixed dependence model acquire 'individuality'. As a result, in a given model, quantifiers in LFD quantify over admissible assignments, *not over tuples of objects* like the first-order quantifiers. Thus, existential quantifiers in LFD do not seem at first sight to be obviously related to the usual Skolem functions in the semantics of FOL.

But more can be said. In fact, a string of LFD quantifiers *does induce a semantic operator over tuples of objects*, albeit one that, in contrast to its classical counterpart: (a) quantifies over a restricted range of tuples (the ones that are in the range of admissible joint values for the given variables), and (b) imposes further constraints on the corresponding Skolem function, requiring it to behave well with respect to the admissible assignments. These additional features are both natural and informative in generalized assignment semantics. Indeed, quantifier combinations in LFD play a twofold role, giving information both about *objects* and about *variable ranges and dependencies.*

More precisely, let us compare the meaning of some quantifiers and quantifier combinations in LFD with their classical meanings in FOL. To do this, we need some notation. Given a tuple of variables $\mathbf{x} = (x_1, \ldots, x_n) \in V^*$, let $X := \{x_1, \ldots, x_n\}$ be the set of its variables. Also, for any given dependence model $\mathbf{M} = (O, I, A)$, we denote by

$$O^{\mathbf{x}} := \{s(\mathbf{x}) : s \in A\} = \{(s(x_1), \ldots, s(x_n)) : s \in A\}$$

the *range of admissible* $\mathbf{x}$-*values*, as a subset of O^n. As a special case, we have

$$O^{(x)} = \{s(x) : s \in A\} \subseteq O.$$

For a start, with the given notational convention, consider the LFD formula

$$\forall_X P\mathbf{x}$$

This holds in a full model (with $A = O^V$) iff we have

$$O^n \subseteq I(P)$$

So on *full models*, $\forall_X$ captures universal quantification, exactly as $\forall X$ does in FOL. But over an *arbitrary* dependence model $\mathbf{M} = (O, I, A)$, the same formula holds iff we have

$$O^{\mathbf{x}} \subseteq I(P)$$

This is clearly universal quantification, but only over the restricted range $O^{\mathbf{x}}$ of admissible simultaneous $\mathbf{x}$-values. Though weaker than the FOL formula $\forall\mathbf{x}P\mathbf{x}$, this is precisely the natural meaning in a dependence model, where each variable x or tuple of variables $\mathbf{x}$ has its own range of (tuples of) values. The universal LFD quantifier simply quantifies over that range.

Our second, perhaps more telling, example is a quantifier combination expressing a functional dependence.[26] It is easy to see that the LFD formula

$$\forall_x \exists_y Pxy$$

holds in a full model iff there is a function witnessing this fact:

$$\exists F : O \to O \,\forall o \in O : (o, F(o)) \in I(P)$$

This matches the usual Skolem-type meaning of the FOL formula $\forall x\exists y Pxy$. But spelling out the LFD semantics for the defined dependence quantifiers, over an arbitrary dependence model $\mathbf{M} = (O, I, A)$, the same formula $\forall_x \exists_y Pxy$ holds iff we have

$$\exists F : O^{(x)} \to O \,\forall o \in O^{(x)} \,(o, F(o)) \in I(P) \cap O^{(x,y)} \ \ ^{27}$$

This statement is *neither weaker, nor stronger* than the one expressed by the FOL formula. On the one hand, the domain of F is restricted to the admissible x-values, which is a weakening. But on this restricted range, we have a stronger statement: not only do all resulting pairs $(o, F(o))$ satisfy P, but they are all realized by admissible simultaneous assignments of values to (x, y). Once again, this is a natural statement: in the earlier terms, the combination $\forall_x \exists_y$ gives information about objects *and* on how these objects can be accessed by variables.

This twofold nature is shared by all quantifier combinations in LFD, making them meaningful in a broader realm than the classical quantifier combinations to which they reduce on full models. Alternatively, they can also be viewed as just being restricted versions of the classical quantifier combinations, but with the added value that they are now forced to also give information beyond their traditional comfort zone (about constraints and correlations on variable ranges).

Summing up the presentation so far, LFD has both dependence modalities and quantifiers in one setting. But one can also view the system at a higher level. Modalities can be seen as quantifiers, as is well-known in modal logic, [22], and conversely, a system like CRS shows how first-order quantifiers can be seen as modalities. Thus, two perspectives are possible on LFD: it is both a first-order logic and a modal logic. This interplay will continue throughout this paper, as it allows for borrowing notions and techniques from both sides. In the remainder of this section, the two intertwined perspectives are taken a bit further, starting with a connection of LFD to standard FOL in terms of translation between languages and semantics.

3.3   First-order translation

As is the case for modal logic, the preceding language can be translated faithfully into a first-order language. But before doing so, it is important to be clear in which sense this is meant.

---

[26] The analysis here is also reminiscent of the discussion in Section 2.3, but we forego details.

[27] In more detail, starting from any assignment, the first quantifier $\forall x$ ranges over all admissible assignments s and any value for x there, the second $\exists y$ then ranges over all admissible assignments t with $s =_x t$.

LFD can be seen as a weak, decidable first-order logic over generalized models. But sometimes, a language interpreted over generalized models can be translated into a fragment of that same language interpreted over the original standard models.[28]

By the Locality of LFD, it is enough to consider a finite set $V = \{v_1, \ldots, v_n\}$ of n variables, with a given enumeration. First take fresh copies of these variables $V' = \{v'_1, \ldots, v'_n\}$. Also introduce a new n-ary predicate A where intuitively, $A(v_1, ..., v_n)$ encodes the fact that the tuple of values assigned to $v_1, \ldots, v_n$ belongs to the admissible assignments A of the current dependence model. Now consider FOL with variables in $V \cup V'$ and predicates in $Pred \cup \{A\}$.

For each dependence model $\mathbf{M}$, there is an *associated FOL model* $T(\mathbf{M})$ for this extended language, having the same domain and the same interpretation of the old predicate symbols, and with the new predicate A interpreted as above. Conversely, *every FOL model for the extended language is the translation $T(\mathbf{M})$ of some dependence model.*

Definition 3.10. *The* first-order translation $tr(\varphi)$ *from* LFD-*formulas* φ *to first-order formulas in the above finite vocabulary is defined as follows.*

(a) $tr(P\mathbf{x}) = P\mathbf{x}$, (b) $tr(\neg\varphi) = \neg tr(\varphi)$, (c) $tr(\varphi \wedge \psi) = tr(\varphi) \wedge tr(\psi)$,

(d) $tr(\mathbb{D}_X\varphi) = \forall\mathbf{z}(A\mathbf{v} \to tr(\varphi))$, where $\mathbf{v}$ is the enumeration of all the variables in V and $\mathbf{z}$ is the enumeration of all the variables in $V - X$,

(e) $tr(D_X y) = \forall\mathbf{z}\forall\mathbf{z}'((A\mathbf{v} \wedge A\mathbf{v}[\mathbf{z}'/\mathbf{z}]) \to y = y')$, where $\mathbf{v}$ and $\mathbf{z}$ are as in part (d), $\mathbf{z}'$ and y' are the corresponding fresh V'-copies of $\mathbf{z}$ and respectively y, and $A\mathbf{v}[\mathbf{z}'/\mathbf{z}]$ is the result of replacing the variables $\mathbf{z}$ by $\mathbf{z}'$ in the formula $A\mathbf{v}$.

Free and *bound occurrences* of a variable in a FOL formula are defined as usual. Variables are allowed to occur both free and bound in different parts of the same formula, so that freely occurring variables can be reused in quantification. The *free variables of a FOL formula* are also defined as usual: as those variables that occur free at least once in the formula.

It is easy to see from the above translation that, for every formula φ of LFD over V, *the set of free variables of its FOL translation $tr(\varphi)$ is exactly $Free(\varphi)$.*

Fact 3.11. *For all* LFD *models* $\mathbf{M}$ *and LFD formulas* φ, *we have:*

$$\mathbf{M}, s \models \varphi \quad \text{iff} \quad T(\mathbf{M}), s \models tr(\varphi),$$

where $tr(\varphi)$ is the above FOL translation of φ, and $T(\mathbf{M})$ is the FOL model associated to $\mathbf{M}$.

The proof is a simple induction following the idea of the stated translation.

Corollary 3.12. *The validities of* LFD *are recursively enumerable.*[29]

Further benefits of the above translation include immediate transfer of the fundamental Compactness and Löwenheim-Skolem properties of FOL to LFD.

---

[28]Cf. the analysis of two-way connections between CRS and the Guarded Fragment of FOL in [13], [14].

[29]By Fact 3.11 (and the above observation that every FOL model for the extended language is the translation of a dependence model), a LFD formula is satisfiable iff its first-order translation is. The statement then follows from the completeness theorem for FOL and the effectiveness of the above translation. Since the above translation is easily extended to all other dependence logics considered in this article, the corollary holds for all of these.

3.4 Modalization of LFD: standard relational semantics

Next, we elaborate the *modal* perspective on LFD. An equivalent semantics is obtained by abstracting away the assignments from their concrete set-theoretical interpretation as functions and treating them as abstract possible worlds. This eliminates all references to values assigned to variables, and replaces identity of values $s =_x t$ by abstract equivalence relations $\sim_x$.

Definition 3.13. *A* standard relational model *is a triple $\mathcal{M} = (W, \sim, \|\bullet\|)$, consisting of: (a) a set W of worlds or 'states'; (b) a map $\sim: V \to \mathcal{P}(W \times W)$ associating to each variable $x \in V$ an equivalence relation $\sim_x$ on W; and (c) a valuation $\|\bullet\|$ associating to each formula of the form $P\mathbf{x}$ a set of worlds $\|P\mathbf{x}\| \subseteq W$. It is useful to introduce auxiliary relations $\sim_X$ on W, for sets of variables $X \subseteq V$, defined by taking intersections $\sim_X := \bigcap_{x \in X} \sim_x$. With this notation, the valuation is required to satisfy the following additional condition:*

    if $w \sim_X v$ and $w \in \|Px_1 \ldots x_n\|$ for some $x_1, \ldots, x_n \in X$, then $v \in \|Px_1 \ldots x_n\|$.

We interpret dual quantifiers $\mathbb{D}_X \varphi$ as universal modalities for the relation $\sim_X$, while dependence atoms $D_X y$ capture a local inclusion (every $\sim_X$-successor is also a y-successor):

Definition 3.14. *In a standard relational model $\mathcal{M} = (W, \sim, \|\bullet\|)$, the notion of truth $\mathcal{M}, s \models \varphi$ (with the index $\mathcal{M}$ dropped when the model is fixed) is given by the valuation for atomic formulas $P\mathbf{x}$, by the usual recursive clauses for the Boolean operators, and by:*

$w \models \mathbb{D}_X \varphi$ *iff* $\forall v \in W \ (\ w \sim_X v \ implies \ v \models \varphi \ )$

$w \models D_X y$ *iff* $\forall v \in W \ (\ w \sim_X v \ implies \ w \sim_y v \ )$

The two kinds of models introduced so far are closely related: we can show that *the standard relational semantics is equivalent to the dependence-model semantics.*
    One direction is given by the following observation:

Fact 3.15. *Every dependence model $\mathbf{M} = (O, I, A)$ induces a standard relational model $rel(\mathbf{M}) = (W, \sim, \|\bullet\|)$, whose possible worlds are the admissible assignments A (so $W := A$), the accessibility relations $s \sim_x t$ are given by pointwise equality of x-values $s =_x t$ (as already defined in dependence models, for both individual variables x and sets of variables X), and the valuation is given by $\|P\mathbf{x}\| = \{s \in W : s(\mathbf{x}) \in I(P)\}$. Moreover, the dependence-model semantics agrees with the relational semantics on the induced model: for all $s \in A = W$ and formulas φ of LFD,*

$$\mathbf{M}, s \models \varphi \ iff \ rel(\mathbf{M}), s \models \varphi.$$

This construction is so tight, that its adequacy should be clear without further proof.
    A slightly less routine construction yields the opposite direction:

Definition 3.16. *Every standard relational model $\mathcal{M} = (W, \sim, \|\bullet\|)$ induces a dependence model $dep(\mathcal{M}) = (O^\sim, I^\sim, A^\sim)$, obtained by taking:*

(a) $O^\sim = \{(x, [w]_x) : w \in W, x \in V\}$, *where objects are pairs $(x, [w]_x)$ of a variable and an equivalence class $[w]_x = \{v \in W : w \sim_x v\}$*

(b) $A^\sim = \{w^\sim : w \in W\}$, *with the admissible assignments $w^\sim(x) = (x, [w]_x)$ for all $x \in V$*

(c) *the interpretation* $I^\sim$ *maps each n-ary predicate* P *to the set*

$$I^\sim(P) := \{(x_1, [w]_{x_1}), \ldots, (x_n, [w]_{x_n})) : w \in W, x_1, \ldots, x_n \in V \ with \ \mathcal{M}, w \models Px_1 \ldots x_n\}.$$

Note that $w^\sim(x) = v^\sim(y)$ implies that $x = y$ and $w \sim_x v$. Using this, one easily checks that $I^\sim(P)$ is well-defined on objects, i.e., independent of the choice of representatives for the equivalence classes. Moreover, the construction preserves truth of LFD formulas:

Fact 3.17. *Given a standard relational model* $\mathcal{M} = (W, \sim, \| \bullet \|)$, *the relational semantics on* $\mathcal{M}$ *agrees with the dependence-model semantics on* $dep(\mathcal{M})$: *that is, for all worlds* $w \in W$ *and all formulas* φ *of* LFD, *we have*

$$\mathcal{M}, w \models \varphi \ \ iff \ \ dep(\mathcal{M}), w^\sim \models \varphi.$$

Proof. The proof is by induction on φ. The atomic case holds by the definition of $I^\sim$ in $dep(\mathcal{M})$. Boolean cases are routine. The inductive cases for $\mathbb{D}_X\varphi$ and D_Xy follow easily from the semantic definitions, together with the following simple fact: $w \sim_x v$ in $\mathcal{M}$ iff $w^\sim =_x v^\sim$ in $dep(\mathcal{M})$. $\square$

The two constructions can also be intertwined, with outcomes such as the following.

Fact 3.18. *For every standard relational model* $\mathcal{M} = (W, \sim, \| \bullet \|)$, *the function* $w \mapsto w^\sim$ *is a surjective homomorphism from* $\mathcal{M}$ *to* $rel(dep(\mathcal{M}))$.

Remark 3.19. An obvious next desideratum is a natural notion of modal bisimulation for LFD, capturing its precise range within the first-order language over standard models. One lead here might be the connection with generalized assignment semantics for FOL. A natural analogue to modal bisimulation for FOL is *potential isomorphism*, using partial assignments from finite sets of variables to objects. The crucial back-and-forth clauses of a potential isomorphism F are easily adapted to generalized assignment semantics.

Open problem   *Find a bisimulation invariance theorem characterizing LFD.*[30]

3.5   Other interpretations: information, knowledge, questions

The relational semantics, and its equivalence with the dependence-models semantics, shows that the actual values of variables do not play an essential role in LFD: what is important are the *relations* of 'agreement on values' of X, and 'dependence of y on X'. This suggests other, non-variable-based interpretations of our logic. Three such interpretations will be outlined here (epistemic, interrogative, and mixed), all *information-based*, like the informational interpretation in Remark 3.6. The informational perspective is ubiquitous: one often talks informally about even ontic dependence in the real world as knowing the value for one variable implying knowing that of the other, or as answers to some questions implying answers to other questions.

A straightforward *epistemic* reading of LFD re-interprets the variables $x \in V$ as *agents*, while the equivalence relations $\sim_x$ represent the agents' *uncertainty relations*. Then the modal statement

---

[30]In response to a preprint version of this paper, Koudijs [54] defined notions of *dependence bisimulation* for our dependence models as well as their modal relational versions, and proved a Characterization Theorem for LFD as a fragment of FOL that is invariant under dependence bisimulations. A similar characterization of LFD was found independently in Pútzstück [68].

$\mathbb{D}_x\varphi$ captures agent x's *individual knowledge*, while $\mathbb{D}_X\varphi$ expresses *distributed knowledge* among the group of agents X, [33]. Dependence atoms $D_x y$ express *knowledge subsumption*: 'agent x knows at least as much as agent y', [29], while atoms $D_X Y$ stand for the analogue notion of group subsumption.[31]

Next, since an equivalence relation is also a partition as used in the traditional semantics of *questions*, [42], dependence models also have an *interrogative interpretation*. Variables x represent *basic questions*, and sets of variables X are *joint questions* asking for the answers to all the given questions). The dependence modality $\mathbb{D}_x\varphi$ is the 'interrogative modality' $Q\varphi$ of [20], while $\mathbb{D}_X\varphi$ extends this to joint questions. Dependence atoms $D_X y$ are local versions of 'inquisitive implication' between questions, see [26, 25] for modern versions.

Finally, in *mixed readings*, some variables stand for agents, others denote objects, while yet others represent questions. Such mixtures greatly enhance the range of LFD. For instance, the logic for mixed readings in [6] captures a group's distributed knowledge of the value of a variable, as well as individual or group knowledge of a dependence between variables.

4 Decidability via type models

In this section, we show that LFD is decidable, using *type models*. These 'models' are just syntactic constructs, with no explicit objects, resembling the 'quasi-models' used in [3], [13] to investigate the Guarded Fragment. Sparse models like this have an independent interest, and they yield a bare-bones proof of decidability. However, the price of this directness is a certain amount of ad-hoc syntactic construction. In Appendix A, we use general semantic methods from modal logic to give a more elegant (though less direct) proof of decidability for LFD.

4.1 Syntactic type models

Consider any finite set F of LFD formulas, and let V_F be the finite set of all variables occurring in F. Add to F all formulas $D_X Y$ for all sets of variables $X, Y \subseteq V_F$. Close the resulting set under subformulas, as well as one round of negations, where explicit negations themselves are left as they are. Call the resulting finite set $\Phi = \Phi_F$. This set will be fixed henceforth, and models and arguments about them will only involve these formulas.

Definition 4.1. *A subset $\Sigma \subseteq \Phi$ is a* Hintikka set *for Φ (also occasionally called a syntactic 'type') if it satisfies the following conditions, where all formulas mentioned run over Φ only:*

  (a) $\neg\psi \in \Sigma$ iff $\psi \notin \Sigma$,   (b) $(\varphi \wedge \psi) \in \Sigma$ iff $\varphi \in \Sigma$ and $\psi \in \Sigma$

  (c) if $\mathbb{D}_X\psi \in \Sigma$, then $\psi \in \Sigma$,   (d) $D_X x \in \Sigma$ for all $x \in X$

  (e) if $D_X Y, D_Y Z \in \Sigma$, then $D_X Z \in \Sigma$.

Note that there are only finitely many Hintikka sets for a given finite set Φ. Moreover, the property of being a Hintikka set for a set F of bounded size $\leq N$ is clearly *decidable*.

Definition 4.2. *For every Hintikka set $\Sigma \subseteq \Phi$ and every set of variables $X \subseteq V_F$, the* dependence closure *of X wrt Σ is the set of variables $D_X^\Sigma := \{y \in V_F : D_X y \in \Sigma\}$.*

---

[31]Even so, some natural epistemic notions lack an obvious match in LFD. What is a dependence counterpart to *common knowledge $C_G\varphi$*, or other epistemic fixed-point notions?

The terminology 'closure' is justified by the following observations. First, clause (d) on Hintikka sets implies that the dependence closure D_X^Σ contains X; second, clauses (b), (e) together imply that D_X^Σ is closed under adding variables z with $D_Y z \in \Sigma$ for any $Y \subseteq D_X^\Sigma$; third, D_X^Σ is the smallest set (in the sense of set inclusion) of variables satisfying the first two properties. If Z is any other set satisfying the two properties, then $D_X^\Sigma \subseteq Z$. For, let $z \in D_X^\Sigma$, so $D_X z \in \Sigma$. We have $X \subseteq Z$ by the first property, and so $z \in Z$ by the second property.

Definition 4.3. *For Hintikka sets $\Sigma, \Delta \subseteq \Phi$ and $X \subseteq V_F$,*

$$\Sigma \sim_X \Delta \quad \text{iff} \quad \Sigma \text{ and } \Delta \text{ have the same formulas } \psi \in \Phi \text{ with } Free(\psi) \subseteq D_X^\Sigma.$$

Fact 4.4. *The following hold for all Hintikka sets Σ, Δ, and sets of variables $X, Y \subseteq V$:*

1. *$\Sigma \sim_X \Delta$ implies $D_X^\Sigma = D_X^\Delta$,*

2. *$\sim_X$ is an equivalence relation,*

3. *$\Sigma \sim_X \Delta$ and $D_X Y \in \Sigma$ imply $\Sigma \sim_Y \Delta$.*

Proof. For the first item: if $\Sigma \sim_X \Delta$, then the sets Σ, Δ contain the same dependence atoms $D_X y$ – since the latter have only free variables X, and $X \subseteq D_X^\Sigma$. It follows that $D_X^\Sigma = D_X^\Delta$.

For the second item: $\sim_X$ is evidently reflexive, by its definition. Symmetry and transitivity also follow immediately from the definition of $\sim_X$ together with the first item (the invariance of $D_X Y$ under $\sim_X$).

The third item follows from the fact that $D_X Y \in \Sigma$ implies that $D_Y^\Sigma \subseteq D_X^\Sigma$. Indeed, if $z \in D_Y^\Sigma$ and $D_X Y \in \Sigma$, then we have $D_Y z, D_X Y \in \Sigma$, thus $D_X z \in \Sigma$ by property (e) of Hintikka sets, and hence $z \in D_X^\Sigma$. $\square$

Next we define a syntactic notion capturing key aspects of the families of Hintikka sets that can occur together in one dependence model. Here Clause (f) reflects the witnessing for existential dependence modalities in the model, and Clause (g) the fact that constants (i.e., variables x for which $D_\emptyset x$ holds) behave uniformly in the model.

Definition 4.5. *A type model for Φ is a family $\mathfrak{M}$ of Hintikka sets for Φ obeying the following two conditions. The first is an additional 'witness condition' for existential modalities:*

(f) if $\mathbb{E}_X \psi \in \Sigma \in \mathfrak{M}$, then there exists a set $\Delta \in \mathfrak{M}$, such that (i) $\psi \in \Delta$, (ii) $\Sigma \sim_X \Delta$.

The second condition expresses uniformity for constants:

(g) $\Sigma \sim_\emptyset \Delta$ (as given in Definition 4.3) holds for all $\Sigma, \Delta \in \mathfrak{M}$.[32]

Once again, for a given finite set F, there are only finitely many type models for Φ_F, and moreover, the property of being a type model for a set F of bounded size N is decidable.

---

[32]I.e., $\sim_\emptyset$ is the universal relation on $\mathfrak{M}$. This means that all Hintikka sets in $\mathfrak{M}$ contain the same atomic statements $D_\emptyset x$ (if any) and the same formulas whose free variables are all in the set of these constant x.

4.2 Representation of type models as dependence models

First, it is easy to see that every dependence model induces a type model.

Definition 4.6. *Given a dependence model* $\mathbf{M} = (M, A)$ *and a set* Φ *as in the previous section, the* Φ-*type of an assignment* $s \in A$ *is defined as*

$$type(s) \;=\; \{\psi \in \Phi \mid \mathbf{M}, s \models \psi\}.$$

Fact 4.7. *For every assignment* $s \in A$ *in a dependence model* $\mathbf{M} = (M, A)$, *its* Φ-*type* $type(s)$ *is a Hintikka set. Moreover, the set* $type(A) := \{type(s) : s \in A\}$ *of all* Φ-*types occurring in* $\mathbf{M}$ *is a type model.*

Proof. Checking conditions (a)–(e) on Hintikka sets is straightforward. For the witness condition (f) in the type model, let $\Sigma = type(s)$ for $s \in A$, and let $\mathbb{E}_X \psi \in \Sigma$, i.e. $s \models \mathbb{E}_X \psi$. By the semantics of LFD, there exists $t \in A$ with $s =_X t$ and $t \models \psi$, i.e. $\psi \in type(t)$. By the Locality Lemma 3.4, s, t make the same formulas true whose free variables are among the X, which includes all dependence atoms $D_X y$. Therefore, s, t agree on all variables in the set D_X^{Σ}, and so, once more by Locality, we have that $type(s) \sim_X type(t)$ in the sense of Definition 4.3. Finally, condition (g) reflecting the uniform behavior of constants again follows from Locality in dependence models.

$\square$

The more challenging direction is now the converse: that every type model can be represented as the set of types of some dependence model.

Theorem 4.8. *Given a type model* $\mathfrak{M}$, *there exists a dependence model* $\mathbf{M} = (M, A)$ *with*

$$\mathfrak{M} = \{type(s) : s \in A\}.$$

Proof. First fix any Hintikka set $\Sigma_0 \in \mathfrak{M}$. Define a *good path* to be a finite sequence $\pi = \langle \Sigma_0, X^1, \Sigma_1, \ldots, X^n, \Sigma_n \rangle$ of any length $n + 1 \geq 1$ such that (i) $\Sigma_k \in \mathfrak{M}$ for each k (hence each Σ_k is a Hintikka set in $\mathfrak{M}$), and (ii) each $X^k \subseteq V_F$ satisfies $\Sigma_{k-1} \sim_{X^k} \Sigma_k$. Write $last(\pi) = \Sigma_n$ for the *last element* of path π.[33]

In what follows, it is convenient to view good paths as consisting of successive *good transitions* of the form (Σ, X, Δ). Here we think of the variables in X, and those depending on them according to Σ, as keeping their value in the transition. More precisely, we say that

the variables kept fixed in a transition (Σ, X, Δ)

are all those in the extended set of variables D_X^{Σ} introduced in Definition 4.2.[34] Sets of variables kept fixed in good transitions underlie many of the definitions and proofs that follow.

The good paths are finite sequences that form a rooted branching tree in a standard manner, with the 1-length path $\langle \Sigma_0 \rangle$ as its root. It may help the reader to keep a tree picture in mind in what follows, cf. Figure 1 below for a visual aid.

Next, *objects* will be special pairs of good paths and variables. Instead of defining these objects separately, we introduce them simultaneously with the following inductive definition of *path assignments* v_π for good paths π, that send variables to objects:

---

[33]This definition creates infinitely many good paths, and as we shall see in a moment, infinitely many objects. Whether this can be restricted to a finite set of paths and values is at present an open problem.

[34]D_X^{Σ} equals D_X^{Δ} by Fact 4.4, so the reverse transition is also good.

23

$v_\pi(x) = (\pi, x)$ if π has length 1, i.e. $\pi = \langle \Sigma_0 \rangle$ is the root of our tree.

$v_\pi(x) = v_{\pi'}(x)$ if $\pi = (\pi', X, \Sigma)$ with $x \in D_X^{last(\pi')}$.

$v_\pi(x) = (\pi, x)$ if $\pi = (\pi', X, \Sigma)$ with $x \notin D_X^{last(\pi')}$

The second clause leaves the same values for variables if the last transition keeps them 'fixed'. The third clause creates fresh objects as soon as this fixing is not prescribed. In particular, note that *constants* x, i.e. special variables with $D_\emptyset x$ present in *all* Hintikka sets in $\mathfrak{M}$, will get the *same value* $(\langle \Sigma_0 \rangle, x)$ under all path assignments. By condition (g) on type models, that value never changes for longer paths.

Now, we define a first-order model $M = (O, I)$ by letting

$$O = \{ v_\pi(x) : \pi \text{ good path and } x \in V_F \}$$

be the set of all objects (π, x) assigned by the assignments v_π in the above manner. Next, an interpretation $I(P)$ is given to each predicate by means of the following 'coherence condition':

$I(P)$ holds for a finite sequence of objects $\overline{(\pi, x)}$ in O if all paths π occurring in the sequence are linearly ordered by the relation of initial segment, and the formula $P\mathbf{x}$ occurs in $last(\pi^*)$ on the longest path π^* among these.

Finally, a dependence model $\mathbf{M} = (M, A)$ is obtained over the first-order model M by setting

$$A = \{ v_\pi : \pi \text{ is a good path} \}$$

The crucial semantic notion of equality of values among assignments v_π, $v_{\pi'}$ in the dependence model $\mathbf{M}$ wrt a given set X of variables may be described concretely as follows. In general, the paths π, π' fork beyond a shared initial segment π'', that includes at least $\langle \Sigma_0 \rangle$. The semantic equality $v_\pi =_X v_{\pi'}$ means that the values assigned by v_π and $v_{\pi'}$ to all variables in X have been set already by the final stage of π'' (cf. Figure 1):

Fact 4.9. *For any two $v_\pi, v_{\pi'}$ in $\mathbf{M}$ and any set of variables X, the following are equivalent:*

(a) $v_\pi =_X v_{\pi'}$

(b) π and π' have the form $\pi = \pi'', X_1, \ldots, X_n, last(\pi)$, $\pi' = \pi'', X_1', \ldots, X_m', last(\pi')$ with a shared path π'', where all variables in X are kept fixed in the transitions involving the displayed sets $X_1, \ldots, X_n$ and $X_1', \ldots, X_m'$.

Proof. This follows by inspection of the above definitions for values of assignments, noting that the identical objects assigned by v_π and $v_{\pi'}$ to any variables $x \in X$ must be of the form $(\pi^\bullet, x)$ for some initial segment $\pi^\bullet$ of the shared path π'', while no further changes have taken place.[35] $\square$

To complete the proof of the main theorem, we must show that our initially given type model coincides with the set of Φ-types of all assignments in $\mathbf{M}$, i.e. that we have: $\mathfrak{M} = \{ type(v_\pi) : v_\pi \in A \}$. And in order to establish this identity, it suffices to prove that

$$type(v_\pi) = last(\pi) \quad \text{for all good paths } \pi.$$

---

[35] Note that this description also covers the case when X is empty: the 'fork' can then be right after $\langle \Sigma_0 \rangle$.
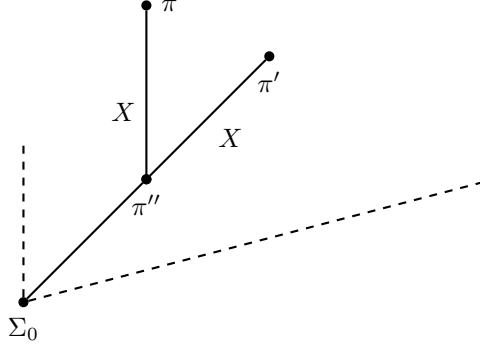
Figure 1: Forking paths π and π' with $v_\pi =_X v_{\pi'}$: according to Fact 4.9, all variables in X are kept fixed in all transitions (on these paths) beyond the shared path π''.

Once we proved this claim, the desired identity $\mathfrak{M} = \{type(v_\pi) : v_\pi \in A\}$ is immediate.[36] Unfolding now the claim $type(v_\pi) = last(\pi)$, we can see that our remaining task is to prove the following result.

Fact 4.10 (Truth Lemma). *For all formulas $\varphi \in \Phi$ and good paths π, the following holds:*

$$\mathbf{M}, v_\pi \models \varphi \text{ iff } \varphi \in last(\pi).$$

Proof. The proof is by induction on the formula φ.

Case 1: *Atomic formulas.* By the truth definition for LFD, $\mathbf{M}, v_\pi \models P\mathbf{x}$ iff $I(P)(v_\pi(\mathbf{x}))$. By the above definition of the atomic predicates in the first-order model M, the objects $v_\pi(\mathbf{x})$ are pairs (π', x) whose paths π' are all initial subpaths of the longest path π^* among them. Moreover, the formula $P\mathbf{x}$ belongs to $last(\pi^*)$. Now, given the above inductive definition of assignments, all objects assigned by v_π to variables have a path component which is an initial segment of π. In particular, π^* is an initial segment of π, and also, again by the inductive definition of the assignments, no values of variables $x \in \mathbf{x}$ have changed along the remaining path from π^* to π. This means, by Definition 4.3 for the relations $\sim_X$ that the formula $P\mathbf{x}$ itself occurs in every Hintikka set in π after π^*, and in particular, that $P\mathbf{x}$ occurs in the set $last(\pi)$.

Case 2: *Boolean combinations.* The proof is a straightforward appeal to the truth definition, the inductive hypothesis, and the definition of Hintikka sets.

Case 3: *Dependence modalities.* For ease of presentation, we consider the existential LFD dependence modality instead of the universal one.

From right to left. Let $\mathbb{E}_X\varphi \in last(\pi)$. By the witness condition (f) on type models, there exists a set $\Delta \in \mathfrak{M}$ with $\varphi \in \Delta$ and $last(\pi) \sim_X \Delta$. Let $\pi^+ = (\pi, X, \Delta)$ be the good path consisting of π with a $\sim_X$-transition to Δ added. By the inductive hypothesis, $\mathbf{M}, v_{\pi^+} \models \varphi$, and hence also $\mathbf{M}, v_{\pi^+} \models \mathbb{E}_X\varphi$. Now consider the objects that v_{π^+} assigns to the variables in X. By the above definition for v_{π^+}, none of the variables $x \in X$ changed their value in the last step $\sim_X$ – and so,

---

[36]To see this, in one direction, each set $last(\pi)$ for a path π is by definition a Hintikka set in $\mathfrak{M}$, In the opposite direction, each set $\Gamma \in \mathfrak{M}$ immediately gives a good path $\pi = (\Sigma_0, \emptyset, \Gamma)$ of length 2, since $\sim_\emptyset$ was the universal relation on $\mathfrak{M}$, and hence there is a matching assignment v_π with $type(v_\pi) = last(\pi) = \Gamma$.

these objects are the same as those assigned by v_π. Thus, the assignments v_{π^+}, v_π agree on the values of the free variables for $\mathbb{E}_X\varphi$, and so, by the Locality Lemma 3.4, the latter formula is also true at $\mathbf{M}, v_\pi$.

From left to right. Let $\mathbf{M}, v_\pi \models \mathbb{E}_X\varphi$. By the truth definition, there is an assignment $v_{\pi'} =_X v_\pi$ with $\mathbf{M}, v_{\pi'} \models \varphi$, so, by the inductive hypothesis $\varphi \in last(\pi')$. By condition (c) on Hintikka sets (dualized to the existential dependence modality), we then have $\mathbb{E}_X\varphi \in last(\pi')$. Now compare the two good paths π, π', keeping Fact 4.9 in mind concerning their shape wrt some shared initial path π'', and the fact that X is contained in the set of variables kept fixed in each transition made on the paths extending beyond π'' toward $last(\pi)$ and toward $last(\pi')$.

Given that $\mathbb{E}_X\varphi \in last(\pi')$, with free variables X, it follows by Definition 4.3 that this formula is present in each Hintikka set on the path toward $last(\pi'')$ and then in each Hintikka set on the path from there toward $last(\pi)$.[37] So, finally, $\mathbb{E}_X\varphi \in last(\pi)$.

Case 4: *Dependence atoms.* The case of dependence atoms is proved in a similar manner, but interestingly, it makes no appeal to a witness clause for non-dependence in type models.

From right to left. Let $D_X y \in last(\pi)$. Local semantic dependence of y on X at the assignment v_π is shown as follows. Consider any assignment $v_{\pi'} \in A$ assigning the same objects to the variables in X, i.e., $v_\pi(X) = v_{\pi'}(X)$. Just as in the preceding Case 3, X-values have not changed after the largest common initial segment π'' of π and π'. But then, since $Free(D_X y) = X$, the formula $D_X y$ is shared by the Hintikka sets in each of these later transitions. Now the above recursive definition of values $v_\pi(u)$ for variables u under assignments v_π worked with extended sets of variables $D_X^{last(\pi')}$ for immediately preceding subpaths π', and these sets all include y in the present case. It follows that $v_\pi(y) = v_{\pi'}(y)$, as desired,

From left to right. Let $\mathbf{M}, v_\pi \models D_X y$. Consider the good path $\pi^+ = (\pi, X, last(\pi))$ extending π with one good $\sim_X$ transition to the Hintikka set $last(\pi)$. By the earlier definitions for the values given by our assignments, v_π, v_{π^+} assign the same objects to all the variables $x \in X$, kept fixed in the final transition. Therefore, by the given local semantic dependence at v_π, we also have that $v_{\pi^+}(y) = v_\pi(y)$. But this can only happen if the variable y, too, was kept fixed in the last transition of π^+, which means by definition that $y \in Y = D_X^{last(\pi)}$: i.e., $D_X y \in last(\pi)$. $\qquad\square$

This concludes the proof of Theorem 4.8. $\qquad\square$

4.3 Decidability

The decidability of LFD can now be established.

Theorem 4.11. *Validity for formulas of* LFD *on dependence models is decidable.*

Proof. By Theorem 4.8 and Fact 4.7, satisfiability for a formula φ in dependence models is equivalent to φ's occurring in some Hintikka set of some type model for the set Φ generated by $F = \{\varphi\}$ and all its subformulas and dependence formulas in the manner described earlier. As there are only finitely many type models of this sort, the latter test is decidable. $\qquad\square$

Open problems *Does LFD have the Finite Model Property? What is the computational complexity of satisfiability for LFD?*

---

[37]What we use here is the earlier observation that good transitions are good in both directions.

Remark 4.12. As noted earlier, the proof of decidability for LFD presented here is an extension of that for the Guarded Fragment of first-order logic, [3]. An open problem is whether we can reduce the decidability problem for LFD to that for the Guarded Fragment with identity, [39], though this seems unlikely given the syntax of dependence atoms. Another issue in this connection is whether known decidable extensions of the Guarded Fragment such as the 'loosely guarded fragment', [13], [14], have counterparts in natural extensions of LFD.

Finally, it may be worth noting that the preceding style of decidability argument can also be applied to first-order logic itself. Hintikka sets and type models can be defined just like above, and the representation result for type models as dependence models also goes through. Moreover, it is decidable whether a given first-order formula has a type model. Given the undecidability of FOL, it must then be an undecidable problem whether a given type model can be represented as a standard first-order model, i.e., a *full* assignment model.

5 Axiomatizations

It was shown in Section 3.2 that the set of LFD validities is recursive. In this section the structure of this set will be explored in more depth, in the form of two complete deductive systems.

5.1 A Hilbert-style axiomatization

A Hilbert-style *proof system* **LFD** is given in Table 1, consisting of: (I) the classical axioms and rules of propositional logic; (II) axioms and rules for dependence modalities, that can be seen as restricted duals of the classical Hilbert axioms for quantifiers; (III) axioms governing the behavior of dependence atoms (namely, Projection and Transitivity, already known to be equivalent to the conjunction of Reflexivity, Monotonicity and Transitivity); (IV) the key Transfer axiom, describing the interaction between dependence modalities and dependence atoms. The notions of formal derivation and provability are defined as usual.

(I)	**Axioms and rules of classical propositional logic**
(II)	**Axioms and rules for dependence modalities** $\mathbb{D}$
($\mathbb{D}$-Necessitation)	From φ, infer $\mathbb{D}_X\varphi$
($\mathbb{D}$-Distribution)	$\mathbb{D}_X(\varphi \rightarrow \psi) \rightarrow (\mathbb{D}_X\varphi \rightarrow \mathbb{D}_X\psi)$
($\mathbb{D}$-Introduction)	$\varphi \rightarrow \mathbb{D}_X\varphi$, provided that $Free(\varphi) \subseteq X$
($\mathbb{D}$-Elimination)	$\mathbb{D}_X\varphi \rightarrow \varphi$
(III)	**Axioms for dependence atoms** D
(Projection)	$D_X x$, provided that $x \in X$
(Transitivity)	$(D_X Y \wedge D_Y Z) \rightarrow D_X Z$
(IV)	**Axiom for** $\mathbb{D}$-D **interaction**
(Transfer)	$(D_X Y \wedge \mathbb{D}_Y\varphi) \rightarrow \mathbb{D}_X\varphi$

Table 1: The proof system **LFD**.

Fact 5.1. *In the context of the other axioms and rules presented in Table 1, the axiom schema* *($\mathbb{D}$-Introduction) can be replaced by its instances listed in Table 2.*

($\mathbb{D}$-Intro$_1$)	$Px_1 \ldots x_n \to \mathbb{D}_{\{x_1,\ldots,x_n\}} Px_1 \ldots x_n$
($\mathbb{D}$-Intro$_2$)	$\mathbb{D}_X \varphi \to \mathbb{D}_X \mathbb{D}_X \varphi$
($\mathbb{D}$-Intro$_3$)	$\neg \mathbb{D}_X \varphi \to \mathbb{D}_X \neg \mathbb{D}_X \varphi$
($\mathbb{D}$-Intro$_4$)	$D_X \mathbf{y} \to \mathbb{D}_X D_X \mathbf{y}$

Table 2: The relevant instances of dependence modality introduction.

Note that, unlike with CRS, the provable principles for **LFD** are closed under substitution for predicate letters. Note also the analogy between $\mathbb{D}$-Necessitation, $\mathbb{D}$-Distribution, $\mathbb{D}$-Elimination, $\mathbb{D}$-Intro$_2$ and $\mathbb{D}$-Intro$_3$ with the usual axioms and rules of the modal system $S5$. This is unsurprising and it is more than an analogy: as seen in Section 3.4, our dependence modalities $\mathbb{D}_X$ are in fact relational modalities for equivalence relations $\sim_X$, and so they automatically validate all the $S5$ laws (by known results in classical modal correspondence theory, [22]).

Example 5.2. *The following formulas are derivable as theorems in* **LFD**:

    (a) $D_X Y$, *for $Y \subseteq X$*   *(Inclusion)*
    (b) $(D_X Y \wedge D_Z U) \to D_{X \cup Z}(Y \cup U)$ *(Additivity of Dependence)*
    (c) $D_X Y \to D_Z Y$, *for $X \subseteq Z$*   *(Monotonicity of Dependence)*
    (d) $\mathbb{D}_X \varphi \to \mathbb{D}_Y \varphi$, *for $X \subseteq Y$*   *(Monotonicity of Dependence Quantifiers)*
    (e) $\forall_X \varphi \to \varphi$; $\varphi \to \forall_X \varphi$ *if $Free(\varphi) \cap X = \emptyset$*   *(Universal Quantifier Laws).*

Note that the more general quantifier elimination rule via substitution $\forall_x \varphi \to [y/x]\varphi$ (as in classical FOL) is *not* a theorem or axiom of **LFD**: indeed, as we saw in Example 3.8, this rule is not sound in our semantics.

Theorem 5.3. *(Completeness) The system* **LFD** *is sound and complete for dependence models.*

Proof. Given a consistent formula φ, consider the set $\Phi = \Phi_F$ generated by $F = \{\varphi\}$ as in Section 4.1. Fix some maximally consistent subset Σ^* of Φ that contains φ, and let $\mathfrak{M}$ be the family of all maximally consistent subsets of Φ that are connected to Σ^* via a finite sequence of relations $\sim_X$ as introduced in Definition 4.3.[38]

Fact 5.4. *The family* $\mathfrak{M}$ *is a type model.*

Proof. Maximally consistent subsets are Hintikka sets: they obviously satisfy the Boolean clauses, and the other closure conditions follow from their closure under deduction. To prove that $\mathfrak{M}$ satisfies the witness condition (f) on type models, let $\mathbb{E}_X \psi \in \Sigma \in \mathfrak{M}$. Take $Y := D_X^\Sigma$ (the dependence closure of X with respect to Σ), and consider the set

$$\Delta_0 := \{\psi\} \cup \{\theta \in \Sigma : Free(\theta) \subseteq Y\}.$$

This set of formulas is consistent by a standard modal argument using the $S5$ axioms[39] for $\mathbb{D}$, the presence of the formulas $D_X y$ in Σ, and the Transfer Axiom of **LFD**. The required Hintikka set Δ

---

[38] This corresponds to taking a 'generated submodel', a standard technique in modal logic.

[39] These are $\mathbb{D}$-Necessitation, $\mathbb{D}$-Distribution, $\mathbb{D}$-Elimination, $\mathbb{D}$-Intro$_2$ and $\mathbb{D}$-Intro$_3$, all available in **LFD**.

can be taken to be any maximally consistent set in $\mathfrak{M}$ that includes Δ_0. Finally, condition (g) on type models is satisfied because all sets in $\mathfrak{M}$ are connected by $\sim_X$ transitions, which are also $\sim_\emptyset$ transitions by the Monotonicity property provable in **LFD**.

This concludes the proof of Fact 5.4, and of the completeness theorem. $\qquad\square$

Theorem 5.3 states 'weak completeness' only. 'Strong completeness' says that provability also matches semantic consequence from possibly infinite sets of formulas.

Theorem 5.5. *The proof calculus* **LFD** *is strongly complete.*

Proof. First, the Compactness Theorem holds for LFD. This follows from the first-order translation in Fact 3.8, plus compactness for first-order logic. Given this, given any valid semantic consequence $\Psi \models \varphi$, we also have a valid consequence $\Psi_0 \models \varphi$ from some *finite subset* $\Psi_0 \subseteq \Psi$ of the premises – and this amounts to the validity of a single formula $\bigwedge \Psi_0 \to \varphi$. By the weak completeness theorem, there is a formal proof of this formula, hence φ is provable from Ψ. $\qquad\square$

In Appendix A, we give another proof of strong completeness, that proceeds along more standard lines using modal logic techniques.

5.2   Sequent calculus, cut elimination and strong interpolation

An alternative formulation of the proof system is as a sequent calculus. To avoid the use of the rules of Contraction and Permutation, we take a Gentzen calculus using *sets* of formulas rather than sequences. In the following, Γ, Δ denote sets of formulas, Γ, φ denotes $\Gamma \cup \{\varphi\}$, etc. $Var(\Gamma)$ is the set of all variables occurring in Γ, and $Free(\Gamma)$ is the set of free variables in Γ.

Definition 5.6. *The* sequent calculus *for* LFD *has the standard Gentzen axioms and rules for classical propositional logic (including structural rules of Identity, Weakening and Cut), together with the following additional axioms and rules:*

$$(Projection) \ \frac{}{\vdash D_X x} \quad where \ x \in X$$

$$(Transitivity) \ \frac{\Gamma \vdash \Delta, D_X Y \qquad \Gamma \vdash \Delta, D_Y Z}{\Gamma \vdash \Delta, D_X Z}$$

$$(\mathbb{D}_L) \ \frac{\varphi, \Gamma \ \vdash \ \Delta}{\mathbb{D}_X \varphi, \Gamma \ \vdash \ \Delta}$$

$$(\mathbb{D}_R) \ \frac{\Gamma \ \vdash \ \Delta, \varphi}{D_X Y, \Gamma \ \vdash \ \Delta, \mathbb{D}_X \varphi} \quad where \ Free(\Gamma \cup \Delta) \subseteq Y$$

Note that, compared with the classical sequent calculus for FOL, there are now extra structural rules for D-Projection and D-Transitivity. Next, the left-introduction rule $(\mathbb{D}_L)$ is weaker than (the dual version of) the classical left-introduction rule for the universal first-order quantifier $\forall$, as it does not allow for variable or term substitutions. Also, the right-introduction rule $(\mathbb{D}_R)$ is different from, and in fact stronger then, the (dual version of the) classical rule for $\forall$: note it involves a dependence-atom premise (incorporating the Hilbert-style Transfer axiom). But also note that $(\mathbb{D}_R)$ implies the weaker rule

$$\frac{\Gamma \;\vdash\; \Delta, \varphi}{\Gamma \;\vdash\; \Delta, \mathbb{D}_X\varphi,} \quad \text{where } Free(\Gamma \cup \Delta) \subseteq X,$$

which can indeed be seen as a dualization of the classical right-introduction rule for the universal quantifier of FOL.

It is easy to show that *the two proof calculi are equivalent in terms of their output*:

Fact 5.7. *The provable sequents $\Gamma \vdash \Delta$ in the above calculus match exactly the provable implications $\bigwedge \Gamma \to \bigvee \Delta$ in the axiomatic system* **LFD**.

Although our sequent calculus lacks standard cut elimination in its full generality, it does have it in a restricted form. Namely, Cut is eliminable in favor of 'DA Cut': this version of the Cut Rule allows cutting only *dependence atoms that involve variables actually occurring in the conclusion*. To ensure the subformula/subterm property, it is also convenient to absorb Weakening into the logical rules (cf. [70], or the explanation in Appendix B), while simultaneously restricting Projection and Transitivity to the variables that actually occur in the sequent to be proven. A *restricted-cut proof* uses only these modified rules and the DA Cut rule. We obtain a limited, but very useful, form of the Cut Elimination Theorem:

Theorem 5.8. *(Restricted Cut Elimination) Every provable sequent $\Gamma \vdash \Delta$ has a restricted-cut proof. Such a proof involves only subformulas of the sequent formulas, or dependence atoms for variables occurring in the final sequent proved.*

The details, as well as a sketch of the proof, are in Appendix B.

Remark 5.9 (Decidability revisited)**.** These results yield a purely proof-theoretic proof of *decidability* for *LFD*. For a given sequent $\Gamma \vdash \Delta$, proof search in the above system with no other structural rule than DA Cut is finite. The search produces a tree whose nodes are sequents $\Gamma' \vdash \Delta'$ consisting only of subformulas of the original sequent or formulas $\mathbb{D}_{\mathbf{x}}\mathbf{y}$ with all $x_i, y_j \in Var(\Gamma \cup \Delta)$. There are only finitely many such formulas, and thus only finitely many such sequents $\Gamma' \vdash \Delta'$ (since Γ', Δ' are sets, there are no repetitions). The pruned tree will be finite, and it contains a proof of the original sequent iff such a proof exists.

Another spin-off is a strong version of Craig Interpolation for LFD. A formula θ is a *strong interpolant* for a sequent $\Gamma \vdash \Delta$ if we have: (1) $\Gamma \vdash \theta$ and $\theta \vdash \Delta$ are valid, (2) all predicate symbols in θ occur both in Γ and in Δ, and (3) all *variables* in θ occur in both Γ and in Δ, i.e., we have $Var(\theta) \subseteq Var(\Gamma) \cap Var(\Delta)$.

Theorem 5.10. *(Strong Interpolation) If $\Gamma \vdash \Delta$ is valid, then there exists a strong interpolant for this sequent.*

Proof. By Completeness and Restricted Cut Elimination, $\Gamma \vdash \Delta$ has a restricted-cut proof. So, it is enough to find strong interpolants for all sequents that are restricted-cut-provable. For this, it suffices to provide strong interpolants for the axioms, and then show how to turn strong interpolants for the premises of each of the above modified rules (including DA Cut) into a strong interpolant for the conclusion. This can be done in the usual way. The strong version of the above interpolation result arises thanks to the tighter variable management provided by DA Cut and restricted Projection and Transitivity. □

As usual, interpolation implies a version of the Beth Definability Theorem. Given a sequent Γ, an n-ary relation symbol P and a tuple of n fresh variables $\mathbf{x} = (x_1, \ldots, x_n)$ with $x_i \notin Var(\Gamma)$, say that Γ *implicitly defines P in variables* $\mathbf{x}$ if the sequent

$$\Gamma, \Gamma' \vdash P\mathbf{x} \leftrightarrow P'\mathbf{x}$$

is valid, where P' is any fresh relation symbol of the same arity as P, and Γ' is the sequent obtained from Γ by replacing every occurrence of P with P'.

Theorem 5.11. *(Strong Beth Definability) If Γ implicitly defines P, then there is a formula θ with $Var(\theta) \subseteq Var(\Gamma) \cup \{x_1, \ldots, x_n\}$, such that the sequent $\Gamma \vdash P\mathbf{x} \leftrightarrow \theta$ is provable.*

5.3 Adding special axioms

Further axioms beyond the logic **LFD** may hold on special classes of dependence models. We give just a few illustrations here, relying heavily on known notions and results from modal logic. For convenience, we will mostly use the existential version of the dependence modality.

Example 5.12. *Consider the following operator interchange principle:*

$$\mathbb{E}_X \mathbb{E}_Y \varphi \to \mathbb{E}_Y \mathbb{E}_X \varphi \qquad \text{(Commutation)}$$

The following dependence model $\mathbf{M}$ is a counterexample. Take two variables x, y and three assignments s, t, u with $s(x) = s(y) = 0, t(x) = 0, t(y) = 1, u(x) = u(y) = 1$. Let R be a binary predicate holding only of the tuple of objects $(1, 1)$. Then $\mathbf{M}, s \models \mathbb{E}_x \mathbb{E}_y Rxy$, as one can reach u by first keeping the value of x fixed, and then that of y. But $\mathbf{M}, s \models \mathbb{E}_y \mathbb{E}_x Rxy$ is false: there is no way of getting from s to u by first keeping the value of y fixed, and then that of x.

On the other hand, it is easy to see that $\mathbb{E}_X \mathbb{E}_Y \varphi \to \mathbb{E}_Y \mathbb{E}_X \varphi$ holds on *full* dependence models (with all functions from V to O as assignments). The crucial property here is the following:

Fact 5.13. *The Commutation axiom $\mathbb{E}_X \mathbb{E}_Y \varphi \to \mathbb{E}_Y \mathbb{E}_X \varphi$ is valid on the class of all dependence models $\mathbf{M} = (M, A)$ satisfying the following closure property for available assignments:*

for every three assignments $s, t, u \in A$, if $s =_X t =_Y u$, then
there also exists an assignment $v \in A$ in $\mathbf{M}$ with $s =_Y v =_X u$.

This technical condition is a Church-Rosser principle requiring the set of available assignments to be rich in alternative pathways. It is in fact the exact semantic content of Commutation, but formulating this precisely requires the modal notion of *frame correspondence*, [22], that we will demonstrate with a different example below. The result of the above Church-Rosser restriction on dependence models is striking:

Fact 5.14. *The logic **LFD** plus the Commutation axiom is undecidable.*

Proof. It is known that the modal CRS-type logic of generalized assignment models plus the commutation axiom $\exists x \exists y \varphi \to \exists y \exists x \varphi$ is undecidable, [63]. Given that this logic can be translated effectively into **LFD** plus the Commutation axiom, the latter logic is undecidable too.[40] □

---

[40] The undecidability can be understood as follows. Commutation is a modal 'Sahlqvist'-type axiom supporting a completeness theorem for its frame-corresponding property, cf. [22] for details. In particular, the cited CRS-type logic is complete for dependence models satisfying the Church-Rosser constraint. Given the grid-like structure of such models, one can then express standard undecidable tiling problems on geometrical grids as satisfiability problems for the logic. Cf. [60] for details of this widely used reduction technique for proving undecidability.

To show a bit more detail of how frame correspondence analysis works, we give an illustration for a related special dependence axiom. Recall the invalid principle $\mathbb{D}_X \mathbb{D}_Y \varphi \to \mathbb{D}_{X \cap Y} \varphi$ mentioned in Example 3.8, perhaps better understood in its existential form

$$\mathbb{E}_{X \cap Y} \varphi \to \mathbb{E}_X \mathbb{E}_Y \varphi \qquad \text{('Stepwise')}$$

Like Commutation, Stepwise expresses an existence constraint on available assignments that holds in full dependence models, but not in all of them. We now give a semantic correspondence analysis, for convenience, in terms of only *three variables* x, y, z. Call an LFD formula φ *true* in a *dependence frame* (a dependence model without an added interpretation for atomic predicates) if, for every interpretation of the predicate letters on the frame (where dependence atoms always keep their fixed interpretation), φ is true at every assignment.

Fact 5.15. *The* Stepwise *axiom is true in a dependence frame iff that frame is a full Cartesian product with all possible combinations of values for the values.*

Proof. We show that, with three variables, frame truth of Stepwise expesses that the admissible assignments A include all functions from x, y, z to the Cartesian product $Val(x) \times Val(y) \times Val(z)$, where $Val(x) := \{s(x) : s \in A\}$ and similarly for $Val(y), Val(z)$.

In one direction, this is straightforward. If the frame has the stated Cartesian structure, then it is easily verified that Stepwise will hold everywhere under every interpretation of the atomic predicates. In the opposite direction, starting from the frame truth of Stepwise, the quantification over all interpretations of atomic predicates allows us to assume that for each assignment s, there exists some predicate $Pxyz$ that holds uniquely for the values $s(x), s(y), s(z)$.[41]

Now, suppose some value d occurs for x at some available assignment s. Suppose also that value e occurs for y at some assignment t, uniquely defined by an atomic formula $Pxyz$. One can reach t from s via the universal relation $=_\emptyset$, so s satisfies $\mathbb{E}_\emptyset Pxyz$. Now write $\emptyset = \{x\} \cap \{y, z\}$. Then by Stepwise, we also have $\mathbb{E}_{\{x\}} \mathbb{E}_{\{y,z\}} Pxyz$ true at s. But that means one can go from s to some assignment u keeping the value of x fixed, and then from u to t keeping the values of y, z fixed. It follows that $u(x) = d, u(y) = e$. Next assume that z takes on value f at some assignment v. Repeating the preceding argument for u and v, now making the split $\emptyset = \{x, y\} \cap \{z\}$, we find an assignment w with $w(x) = d, w(y) = e, w(z) = f$.

$\square$

Again, there is a consequence in terms of logics extending **LFD**.

Fact 5.16. *The logic* **LFD** *plus the* Stepwise *axiom is undecidable.*

Proof. The Stepwise axiom has the modal Sahlqvist form mentioned in Footnote 40, and hence, by general results, [22], this logic is complete for dependence frames satisfying the corresponding condition identified above. Now, the Cartesian product structure obtained here is not a full dependence model in our sense, since each variable can have its own range of objects. But this is no obstacle to the following analysis combining two known facts.

Dependence models with the preceding structure are standard models for the *three-variable fragment of many-sorted first-order logic*, whose satisfiability problem is known to be undecidable, [45]. Moreover, CRS quantifiers are definable by LFD dependence modalities (cf. Section 3.1), while CRS quantifiers just are the first-order quantifiers on standard models.

---

[41]See again [22] for details of this standard move in a modal frame correspondence argument.

It follows that satisfiability of first-order formulas in the many-sorted three-variable fragment reduces to satisfiability of LFD formulas in the preceding Cartesian models. In particular, one just replaces first-order quantifiers $\exists u$ by their obvious LFD-counterparts $\mathbb{E}_{\{x,y,z\}-\{u\}}$. $\qquad\qquad\square$

While the above examples concern semantic restrictions in the spirit of modal logic, the dependence setting also suggests new questions of axiomatization. Recall the three representation results for abstract dependence relations listed in Proposition 2.7. The pivotal second result there concerned *uniform dependence models* where all local dependence relations between variables are the same, and hence also equal the global dependence relation. Uniform dependence models validate the following principles, where $\mathbb{D}_\emptyset$ is the universal modality:

$$D_X y \to \mathbb{D}_\emptyset D_X y, \quad \neg D_X y \to \mathbb{D}_\emptyset \neg D_X y, \quad \text{for arbitrary variables } X, y$$

It is easy to find counter-examples to these implications in arbitrary LFD models.

Open problem   *Axiomatize LFD over uniform dependence models.*[42]

This concludes the analysis of properties of the system LFD. The remaining part of this article explores what lies beyond the base system LFD: extensions of the language, enrichments of the framework, and concrete dependence notions in a number of areas.

6   Richer dependence languages

The modal language of LFD can be extended to describe other natural features of dependence. This section contains a few examples, all with first-order truth conditions, thus making it possible to extend the translation of Section 3.3 making all logics effectively axiomatizable. Some of these extensions are straightforward, and do not affect the decidability of the logic, others do.

6.1   Function symbols and constants

Recall the functional perspective of Section 2.3. It makes sense to add to LFD function terms, built from variables x using a given family of operation symbols f with arities marked. 0-ary function symbols are individual constants c denoting objects. Terms are constructed by the rule

$$t ::= x \mid f\mathbf{t}, \qquad \text{with } \mathbf{t} \text{ a tuple of terms of the arity of } f.$$

In the syntax of formulas, the earlier sets of variables X now become sets T of terms, and one can correspondingly extend the LFD syntax with operators $D_T t$ and $\mathbb{D}_T \varphi$ for such sets of terms T and single terms t. This allows for new sorts of dependence statements, such as

$D_{fxy} gyz$ the value of gyz depends on that of fxy

$\mathbb{D}_{fx} \varphi$ the current value of fx fixes the truth of φ.

Models $\mathbf{M}$ for this extended language come with an interpretation map I for operation symbols, where the semantic clauses for term values read

  (a) $val_s(x) = s(x)$

  (b) $val_s(f\mathbf{t}) = I(f)(val_s(\mathbf{t}))$.

---

[42]The disjoint unions of uniform dependence models in the proof for Proposition 2.7, Clause 3, do not validate the above implications. Still, since the components used disjoint sets of values, except for the common constants, these models validate modified uniformity principles. Again a question of axiomatization arises.

In this setting, it is straightforward to define agreement $s =_T s'$ on the values of all terms in a set T, and use it to give the corresponding semantic clauses for $D_T y$ and $\mathbb{D}_T \varphi$.

This logic is still decidable, but to show this the following notion is needed.

Definition 6.1. *A dependence model is* distinguished *if distinct variables can only take distinct values. For every two distinct variables $x \neq y$ and every assignment $s \in A$: $s(x) \neq s(y)$.*

Fact 6.2. *Every dependence model $\mathbf{M} = (O, I, A)$ induces a distinguished model $\mathbf{M}^d$ of the form (O^d, I^d, A^d) with: $O^d = V \times O$; $I^d(P)((x_1, o_1), \ldots (x_n, o_n))$ iff $I(P)(o_1, \ldots, o_n)$ holds; and $A^d = \{s^d : s \in A\}$, where each assignment $s \in A$ has an associated assignment $s^d : V \to O^d$, given by $s^d(x) = (x, s(x))$. Moreover, the two models are LFD-equivalent: for all assignments $s \in A$ and formulas φ of LFD:*

$$\mathbf{M}, s \models \varphi \quad iff \quad \mathbf{M}^d, s^d \models \varphi.$$

Fact 6.3. *The logic LFD extended with function terms is decidable.*

Proof. One can translate formulas φ in the extended language to formulas $\tau(\varphi)$ in the original LFD language so that φ is satisfiable iff $\tau(\varphi)$ is satisfiable. First, associate to each complex term t occurring in φ some distinct new variable v_t, while keeping the old variables the same. Let V' be the total extended set of variables, and let $\tau_0(\varphi)$ be the LFD formula obtained by replacing all terms t in φ by the matching variables v_t. The required functional dependencies between the variables are expressed as global dependence formulas, e.g., $\mathbb{A}D_{\{v_t, v_{t'}, v_{t''}\}} v_{ftt't''}$. Let φ_0 be the conjunction of all these global dependence formulas, for all terms in φ. Then the translation $\tau(\varphi)$ is simply given by the conjunction $\varphi_0 \wedge \tau_0(\varphi)$.[43]

To check that our translation preserves satisfiability, first assume that a formula φ in the extended language holds for some assignment s_0 in a dependence model $\mathbf{M} = (O, I, A)$. Now construct a model $\mathbf{M}'$ for the extended set of variables V', with the same objects $O' = O$ and interpretation $I' = I$. For this, we take $A' = \{s' : s \in A\}$ as our new set of assignments, where we associated to each old assignment $s \in A$ a new *extended assignment* $s' : V' \to O$, defined by recursively putting: $s'(x) = s(x)$ for $x \in V$, and $s'(v_{f\mathbf{t}}) = I(f)(s'(\mathbf{t}))$. It is easy to see that $s'(v_{\mathbf{t}}) = val_s(\mathbf{t})$ holds for all tuples $\mathbf{t}$ of terms in φ, and moreover that $\tau(\varphi)$ is satisfied by the assignment s_0 in the model $\mathbf{M}'$.

For the converse, let the LFD formula $\tau(\varphi)$ hold for some assignment s_0 in a dependence model $\mathbf{M} = (O, I, A)$. By Fact 6.2, $\mathbf{M}$ may be taken to be distinguished. Now construct a model $\mathbf{M}'$ for the language extended with function terms, by enriching $\mathbf{M}$ with an interpretation $I(f)$ for each function symbol, putting $I(f)(s(v_{\mathbf{t}})) = s(v_{f\mathbf{t}})$. Here, if any of the objects $o_1, \ldots, o_n \in O$ is not the value of any term for an assignment in A, just put $I(f)(o_1, \ldots, o_n) = o_0$ for some arbitrarily chosen object $o_0 \in O$. These functions are well-defined because $\mathbf{M}$ is distinguished, so there is no clash. It is easy to check that φ is satisfied in $\mathbf{M}'$ by the same assignment s_0. $\square$

Fact 6.4. LFD *with function terms is axiomatized by the system* **LFD** *plus*

- *The Functionality Axiom $D_{\mathbf{x}} f\mathbf{x}$ for all function symbols f.*

- *The Substitution Rule "from φ, infer $[\mathbf{t}/\mathbf{x}]\varphi$".*

---

[43]For example, the translation of the formula $Pxf(x, g(y))$ is $\mathbb{A}D_y w \wedge \mathbb{A}D_{x,w} v \wedge Pxv$, where w and v are the fresh variables associated to terms $g(y)$, $f(x, g(y))$, respectively.

Proof. The proof is similar to the previous one, except that we now need a theorem-preserving translation $\tau'(\varphi)$ between the two systems. For any given formula φ in the extended language, we associate new variables v_t as in the proof of Fact 6.3 to each of its terms t, and we construct the formulas $\tau_0(\varphi)$ and φ_0 as in that proof. Then our translation $\tau'(\varphi)$ is simply given by the implication $\varphi_0 \to \tau_0(\varphi)$. It is now easy to check that φ is a theorem in the above extended proof system iff $\tau'(\varphi)$ is a theorem in the basic system **LFD**. The Substitution Rule, as well as the theorem $\mathsf{A}D_{\mathbf{x}}f\mathbf{x}$ (provable in the extended system by applying the Necessitation Rule to the Functionality Axiom) plays a key role in this verification. $\qquad\square$

Note that the additional axiom and rule can be used to establish facts about complex terms. For instance, by the Functionality axiom we have $D_x g(x)$, and then by applying the Substitution rule we get $D_{f(x)}g(f(x))$. Combining this with $D_x f(x)$ (itself another instance of the Functionality axiom) and applying the Transitivity of dependence, we obtain that $D_x g(f(x))$. Applying the Necessitation Rule, we see that in fact this holds globally: $\mathsf{A}D_x g(f(x))$, i.e. we have $= (x; g(f(x)))$.

This extended logic can Skolemize implicit dependencies, in the spirit of Section 2.3 on operational views of dependence, using function symbols as witnesses:

Proposition 6.5. *Let* $\varphi(\mathbf{x}, \mathbf{y}, \mathbf{z})$ *be an LFD formula with free variables* $\mathbf{x}, \mathbf{y}, \mathbf{z}$. *Let* $X = \{x_1, \ldots, x_n\}$, $Y = \{y_1, \ldots, y_m\}$, *and* $f_1, \ldots, f_m$ *fresh n-ary relation symbols not in* φ. *Then*

$$\vdash \varphi(\mathbf{x}, f_1\mathbf{x}, \ldots, f_m\mathbf{x}, \mathbf{z}) \quad \textit{iff} \quad \vdash \mathsf{A}D_X Y \to \varphi(\mathbf{x}, \mathbf{y}, \mathbf{z}).$$

Proof. Apply the same construction as in the proof of Fact 6.4 to the formula on the left, associating fresh variables $y'_1, \ldots, y'_m$ to each of the terms $f_1\mathbf{x}, \ldots, f_m\mathbf{x}$. The same argument as in the preceding proof shows that: $\vdash \varphi(\mathbf{x}, f_1\mathbf{x}, \ldots, f_m\mathbf{x}, \mathbf{z})$ (a) iff $\vdash \mathsf{A}D_X Y' \to \varphi(\mathbf{x}, \mathbf{y}', \mathbf{z})$ (b) (where $Y' = \{y'_1, \ldots, y'_m\}$). To show that (b) implies $\vdash \mathsf{A}D_X Y \to \varphi(\mathbf{x}, \mathbf{y}, \mathbf{z})$, take a proof of $\vdash \mathsf{A}D_X Y' \to \varphi(\mathbf{x}, \mathbf{y}', \mathbf{z})$ and replace any occurrence of variables y'_i by the corresponding variables y_i, obtaining a proof of $\vdash \mathsf{A}D_X Y \to \varphi(\mathbf{x}, \mathbf{y}, \mathbf{z})$. The converse is proven by the inverse substitution (replacing every occurrence y_i in the proof by the corresponding y'_i).[44] $\qquad\square$

However, this functional language still cannot talk about identity of term values, making it impossible to witness implicit dependencies by means of explicit statements $\mathsf{A}(y = f(x))$.

6.2 Explicit equality

We can easily extend our set of predicate symbols with an identity relation $=$ on objects, with the obvious semantics. It is convenient to work with a countably infinite set C of constants, and allow complex terms (built from variables and constants using function symbols) as in the previous section. We denote by c, d, etc. arbitrary constants, and by t, t' arbitrary terms. A *ground term* is one that does not contain any variables (i.e., it is constructed only from constants using function symbols). As before, it is useful to extend our dependence quantifiers and dependence atoms to terms, writing e.g. $\mathbb{D}_T \varphi$ and $D_T t$, where t is any arbitrary term and T is any finite set of terms. As before, we use $\mathbf{x}$ and $\mathbf{t}$ for finite tuples of variables and terms.

Let us call this new logic $LFD^=$. Our translation to FOL can be easily extended to $LFD^=$, so the logic is compact and its set of validities is recursively enumerable. But the new syntax has several advantages, such as supporting a more perspicuous axiomatization of the logic.

---

[44]This syntactic variable replacement property for proofs in **LFD** matches the semantic Renaming Lemma 3.9.

A Hilbert-style *proof system* **LFD$^=$** is given in Table 3, where the letters P shown range over all predicate symbols, including equality.

(I)	**Axioms and rules of classical propositional logic**
(II)	**Special rules**
(Variable Substitution)	From φ, infer $[\mathbf{t}/\mathbf{x}]\varphi$.
(Value Existence Rule)	From $t = c \to \varphi$, infer φ, provided that c does not occur in φ.
(III)	**Axioms and rules for the universal modality**
(A-Necessitation)	From φ, infer $\mathsf{A}\varphi$.
(A-Distribution)	$\mathsf{A}(\varphi \to \psi) \to (\mathsf{A}\varphi \to \mathsf{A}\psi)$
(A-Introduction$_1$)	$P\mathbf{t} \to \mathsf{A}P\mathbf{t}$, provided that $\mathbf{t}$ consists only of ground terms.
(A-Introduction$_2$)	$\mathsf{A}\varphi \to \mathsf{A}\mathsf{A}\varphi$
(A-Introduction$_3$)	$\neg\mathsf{A}\varphi \to \mathsf{A}\neg\mathsf{A}\varphi$
(A-Elimination)	$\mathsf{A}\varphi \to \varphi$
(IV)	**Equality axioms**
(Reflexivity)	$x = x$
(Symmetry)	$x = y \to y = x$
(Transitivity)	$(x = y \wedge y = z) \to x = z$
(Functional Substitution)	$\mathbf{x} = \mathbf{y} \to f\mathbf{x} = f\mathbf{y}$
(Substitution of Equals)	$\mathbf{x} = \mathbf{y} \to (P\mathbf{x} \leftrightarrow P\mathbf{y})$
(V)	**Axioms for dependence atoms and modalities**
(Dependence Atom)	$(\mathbf{x} = \mathbf{c} \wedge y = d) \to (D_X y \leftrightarrow \mathsf{A}(\mathbf{x} = \mathbf{c} \to y = d))$
(Dependence Modality)	$\mathbf{x} = \mathbf{c} \to (\mathbb{D}_X \varphi \leftrightarrow \mathsf{A}(\mathbf{x} = \mathbf{c} \to \varphi))$,
	where in both cases, X is the set of variables occurring in $\mathbf{x}$.

Table 3: The proof system **LFD$^=$**.

Substitution of Equals is a special case of Leibniz' Law of 'indiscernability of identicals', allowing substitution of equal variables in atomic formulas.[45] Essentially, the Value-Existence Rule asserts that each term always has a current value. The Dependence-Atom Axiom 'reduces' local dependence to a universal implication when the current values of the variables are explicitly given, and the Dependence-Modality Axiom does the same for dependence modalities. As a consequence of these context-dependent 'reductions', all the **LFD** principles characterizing the dependence modalities and dependence atoms (i.e. the axioms and rules II, III, IV in Table 1 for $\mathbb{D}$ and D) are missing from from the definition of the new system: indeed, they now become provable theorems in **LFD$^=$**.

---

[45]The general Leibniz Law, that allows substitution of equal variables in *arbitrary* formulas, is not valid, due to the modal character of our semantics: the equality may hold locally at a given world/assignment, while the truth of the relevant formula may depend on the values of the variable at other worlds/assignments.

Theorem 6.6. *(Completeness.) The calculus* **LFD**$^=$ *is sound and complete for validity in the dependence language with equality w.r.t. dependence models.*

The completeness proof follows exactly the lines of that for the logic LED in [6]. The proof uses a Henkin-style canonical model, with the additional twist that the maximally consistent theories are also required to be 'witnessed': for every term t there exists some constant c such that $t = c$ is in the theory. Beyond completeness, however, there is a complication. The type model proof used in Section 4 to show the decidability of LFD does not seem to work in the presence of the Value-Existence Rule.

Open problem *Is LFD with equality decidable?*[46]

6.3 Independence

A major natural extension for LFD concerns the notion of *independence.* Intuitively, saying that y is independent from x is by no means the same as the statement $\neg D_x y$ which just expresses that x does not fix the value of u at the current assignment, and even the universal quantification $\mathsf{A}\neg D_x y$ is too weak. What independence of y from x should mean in the present setting is that *the current value of x does not constrain the values that y can take.*[47]

In epistemic terms, this amounts to saying that knowing the current values of x tells us nothing about the current value of y. To make this precise, we need the following notion.

Definition 6.7. *In any dependence model* $\mathbf{M} = (M, A)$, *the information carried by X about Y at an assignment $s \in A$ is given by the range of Y-values that are compatible with the current values of X (at s):*
$$Inf_s(X, Y) \;\; := \;\; \{t \restriction Y : t \in A, t =_X s\},$$
where $t \restriction Y$ is the restriction to Y of the assignment $t : V \to O$.

We now introduce *independence atoms* $I_X Y$, saying that Y is independent from X at s.

Definition 6.8. *For any model* $\mathbf{M}$, *sets of variables X, Y, and assignment $s \in A$, we put:*
$$\mathbf{M}, s \models I_X Y \quad \textit{iff} \quad Inf_s(X, Y) = Inf_s(\emptyset, Y).$$

One can also define more general conditional independence *atoms $I_X Y | Z$, saying that given Z, X gives no further information about Y:*
$$\mathbf{M}, s \models I_X Y | Z \quad \textit{iff} \quad Inf_s(X \cup Z, Y) = Inf_s(Z, Y).$$

Global independence *(both conditional and unconditional)*[48] *can then be defined from the local versions in the obvious manner, using the universal modality available in LFD:*
$$X \perp\!\!\!\perp Y \;\; := \;\; \mathsf{A} I_X Y.$$
$$(X \perp\!\!\!\perp Y) | Z \;\; := \;\; \mathsf{A}(I_X Y | Z).$$

As usual, when either X or Y are singletons, notation simplifies to $I_X y$, $I_x y$, etc.

---

[46] In response to a preprint version of this paper, [68] has announced a negative answer, proved by reducing satisfiability for the undecidable Kahr-Class of first-order formulas to satisfiability for $LFD^=$-formulas.

[47] Varieties of independence have been studied extensively in IF-logic, [47], and in Dependence Logic, [41]. Our central concern here is how dependence and independence behave on our simple modal basis.

[48] Global conditional independence as defined here may be viewed as a qualitative counterpart to the notion of conditional independence found in Probability Theory.

Reasoning with independence atoms has some interesting features. For instance, it is easy to see that $I_X Y$ does not imply $I_Y X$ locally (the current value of X might carry no information about Y, but that of Y might carry information about X). In contrast with this, however, global independence $X \perp\!\!\!\perp Y$ is *commutative*, and the conditional version $(X \perp\!\!\!\perp Y)|Z$ is commutative in X, Y. If in a model, every value of y can be taken at every value of x, then the set of joint values (x, y) must be a full Cartesian product of the ranges of x and y.

There are also interesting valid principles connecting the modalities I and D. As an illustration, $(D_x z \wedge I_x y) \rightarrow I_z y$ is valid. If the current value of x gives full information about z but no information about y, then the current value of z does not yield any information about y either.

Fact 6.9. *The* dependence atoms can be defined in terms of conditional independence, *via the equivalence* $D_X Y \leftrightarrow I_Y Y | X$, *and the same goes for the global versions.*

These principles are part of a new logic LFDI extending the purely structural rules for independence in [36]. It consists of LFD extended with the basic independence modalities $I_X y$.

Open problem   *Axiomatize the logic LFDI.*

Interestingly, the core logic of independence differs essentially from that of dependence: LFDI is more complex than LFD. The reason is explained in the proof to follow.

Theorem 6.10. *The modal logic* LFDI *is undecidable.*

Proof. The proof is reminiscent of that for Fact 5.16, and uses the undecidability of the three-variable fragment of many-sorted first-order logic. Formulas φ of the latter language with variables x, y, z can be translated into formulas $\tau(\varphi)$ of LFD as indicated at several places earlier on, replacing first-order quantifiers $\exists u \psi$ by existential LFD modalities $\mathbb{E}_{\{x,y,z\}-\{u\}} \tau(\psi)$. Now, crucially, the independence modalities can be used as follows to force the values of x, y, z to be a full Cartesian product, via the formula:

$\mathsf{A}(I_x y \wedge I_{\{x,y\}} z)$,   where A is the universal modality, available in LFD.

Let us show that every model of this formula is a full Cartesian product. Suppose that x takes value d at some assignment in $\mathbf{M}$, y can take e, and z can take f. Consider an assignment s with $s(x) = d$. Since $I_x y$ holds everywhere, the value e for y occurs with this value for x, at some assignment t with $t(x) = d, t(y) = e$. But since also $I_{\{x,y\}} z$ holds everywhere, there must also be an assignment u with $u(x) = d, u(y) = e, u(z) = f$.

Now it is immediate that any three-variable first-order formula φ is satisfiable iff the matching LFDI formula $\tau(\varphi) \wedge \mathsf{A}(I_x y \wedge I_{\{x,y\}} z)$ is satisfiable.   □

Remark 6.11. Moving beyond the contrast between independence and functional dependence, the more general perspective for this section is the notion of *correlated behavior*. Local independence $I_x y$ in our sense says that the current value of x does not place any constraint on the values that y can take at the present location. This is at the opposite extreme from functional dependence $D_x y$ which restricts the range of y to just one value. Clearly, there are other natural notions here. For instance, the mere negation of independence, $\neg I_x y$, says that the current value of x excludes at least one value for y, which can be seen as a weak form of *correlation*. This gives a very minimalistic notion of 'dependence', much weaker than our functional dependence, but one that is of interest on its own. The examples of failure of classical laws for CRS quantifiers presented in the Introduction do not necessitate functional dependence, but only weak correlations: any breach

of full independence between any two variables can lead to such a failure of a classical validity. The complete logic of weak correlation and independence would be the *pure modal logic of $I_X y$*. Stronger notions of correlation or partial dependence between x and y arise when we put further constraints on how far the local value of x constrains that of y, with full functional dependence in the limit.

Open problem *Axiomatize the pure modal logic of independence. Is it decidable?.*

Remark 6.12. The preceding analysis also suggests more general *comparative informational assertions* $X \geq_Y Z$ ('X carries at least as much information about Y as Z does'):

$$s \models X \geq_Y Z \quad \text{iff} \quad Inf_s(X, Y) \subseteq Inf_s(Z, Y).$$

All the above notions of dependence and independence are definable in terms of these comparative assertions. E.g., the conditional independence statement $I_X Y | Z$ says that Z is at least (in fact, just) as informative about Y as $X \cup Z$ is, which can be expressed formally as the equivalence $I_X Y | Z \leftrightarrow Z \geq_Y (X \cup Z)$.

Open problem *Axiomatize the logic of comparative informational assertions.*

6.4 Dynamics and model change

Typically, epistemic events carrying new information can change a current model. One may learn the current value of some variable, or more general facts. There can also be non-informational reasons for changing a current model, say, with a shift of a current dynamical system. A few instances of the dynamics of dependence models will be discussed here, using methods from dynamic-epistemic logic, [7], [28], [15].

Learning current values. One can update a knowledge base after learning the true values of a set of variables X at (the current assignment) $s \in A$. This changes the model $\mathbf{M} = (M, A)$ to the submodel $\mathbf{M}|X = M|\{t \in A : s =_X t\}$ that retains only the assignments that agree with s on all X-values. Now interpret the dynamic modality $[X]\varphi$ as follows:

$$\mathbf{M}, s \models [X]\varphi \quad \text{iff} \quad \mathbf{M}|X, s \models \varphi$$

This modality occurs in epistemic logic under the name of "public inspection of a value", [32].

Fact 6.13. *The logic* LFD *with the modalities $[X]\varphi$ is completely axiomatizable and decidable.*

Proof. It suffices to observe that the following recursion axioms are valid:

(a) $[X]P\mathbf{y} \leftrightarrow P\mathbf{y}$, (b) $[X]\neg\psi \leftrightarrow \neg[X]\psi$, (c) $[X](\alpha \wedge \beta) \leftrightarrow ([X]\alpha \wedge [X]\beta)$,

(d) $[X]\mathbb{D}_Y \psi \leftrightarrow \mathbb{D}_{X \cup Y}[X]\psi$, (e) $[X]D_Y z \leftrightarrow D_{X \cup Y} z$

Used iteratively in a standard dynamic-epistemic style, these reduce each formula in the extended dynamic language to an equivalent base formula of LFD. □

Learning new facts. Another form of information update happens when learning a new *true fact* φ about the current assignment s. Semantically, an update with a formula φ transforms the model $\mathbf{M} = (M, A)$ into the relativized submodel $\mathbf{M} \,|\, \varphi = M \,|\, \{s \in A : s \models \varphi\}$ retaining only the assignments satisfying φ. In the syntax, this is reflected by dynamic modalities $[\varphi]\psi$, with a semantic truth condition given by:

$$\mathbf{M}, s \models [\varphi]\psi \;\; \text{iff} \;\; \mathbf{M}, s \models \varphi \text{ implies } \mathbf{M}|\varphi, s \models \psi$$

Remark 6.14. The logic of this type of update modality ('public announcement logic') is a well-known pilot system of information update. But updating a dependence model can mean different things. Going to a submodel with fewer assignments typically adds to the existing dependencies. In epistemic scenarios, this increase is fine, and in fact useful.[49] However, if the dependence model is a state space for some known current process, changes in the dependence structure create a new process, and this needs to be motivated by other considerations.[50]

A dynamic-epistemic analysis still works for the new extended setting, but there is no longer any reduction to the base language of LFD. Admittedly, the dependence modalities after an update can be reduced to the original ones in a similar way to the well-known recursion law for epistemic modalities:

$$[\varphi]\mathbb{D}_X\psi \;\leftrightarrow\; (\varphi \to \mathbb{D}_X(\varphi \to [\varphi]\psi))$$

But the new dependencies in the updated model $\mathbf{M}\,|\,\varphi$ can only be 'pre-encoded' in the original model $\mathbf{M}$ by means of a *conditional dependence operator* $D_X^\varphi y$, with a semantics given by:

$$s \models D_X^\varphi y \;\; \text{iff} \;\; \forall t \in A\,(\, t \models \varphi \text{ and } s =_X t \text{ imply } s =_y t\,).$$

This is illustrated in the following recursion equivalence, whose validity is easy to check:

$$[\varphi]D_X y \;\leftrightarrow\; (\varphi \to D_X^\varphi y)$$

Of course, conditional dependence needs a recursion law in its turn, and the following is valid:

$$[\varphi]D_X^\alpha y \;\leftrightarrow\; (\varphi \to D_X^{\varphi \wedge [\varphi]\alpha} y)$$

The logic with this update modality can be reduced to its static base logic (with conditional dependence operators) via such recursion laws.

But in this case, the static base logic itself is no longer a routine extension of LFD. The difficulty lies in the following result.

Fact 6.15. *The conditional dependence atom is not definable in* LFD*.*

Proof. For simplicity, consider a language with two variables x, y and one atom Pxy. Take a dependence model $\mathbf{M}$ with just two admissible assignments s, t where $s(x) = s(y) = 0, t(x) = 0, t(y) = 1$, while the binary predicate P holds only of $(0, 0)$ in the underlying first-order model. As for non-trivial dependence atoms in this language, at both assignments, the formulas $D_y x, \neg D_x y$

---

[49]Suppose the truth values of p and q are unknown, whence observing one of them says nothing about the other. But after learning a new fact $p \leftrightarrow q$ imposing a dependency, observing the truth value of one of p, q automatically gives the other. Thus, new dependencies speed up information flow.

[50]A typical model change different from information update is changing the space of relevant variables. This happens, e.g., when analyzing two correlated variables by introducing a new variable on which both depend.

are true. Now extend $\mathbf{M}$ to a model $\mathbf{M}'$ with a third assignment u such that $u(x) = 0$, $u(y) = 2$, while P now also holds of $(0, 2)$. In $\mathbf{M}'$, $D_y x, \neg D_x y$ are true at all of s, t, u.

Now, it is easy to prove by induction that the map F sending the two assignments s and u in $\mathbf{M}'$ to s in $\mathbf{M}$, and the assignment t in $\mathbf{M}'$ to t in $\mathbf{M}$ has the following property. For any LFD formula φ and any assignment v, $\mathbf{M}', v \models \varphi$ iff $\mathbf{M}, F(v) \models \varphi$.[51]

But, conditional dependence sees a difference here: the formula $D_x^{Pxy} y$ is true at s in the model $\mathbf{M}$ (the restriction leaves only the assignment s), but not in $\mathbf{M}'$, since both s, u remain after the restriction. So $D_x^{Pxy} y$ cannot be definable in terms of LFD formulas. $\qquad\square$

Open problem *Axiomatize the modal logic of conditional dependence. Is it decidable?*

Remark 6.16 (Enlarging models)**.** Natural updates can just as well extend current dependence models with new assignments, thereby possibly giving up dependencies that used to hold.

Broader dynamic perspectives. An update perspective suggests extending the semantics of LFD from considering just single dependence models to *families* of these.

Definition 6.17. *A* dependence universe $\mathfrak{U}$ *is a family of dependence models.*

Epistemically, each model in $\mathfrak{U}$ can be seen as a candidate for the true structure of the world, and the dependence universe then represents a 'space of inquiry'. But a dependence universe might also be a family of available processes one can switch between. Either way, just as a dependence model, a dependence universe need not be a full power set of some sort: many possible structures may be missing from the space of inquiry or process repertoire.[52]

A natural extension of LFD describes triples $(\mathfrak{U}, \mathbf{M}, S)$ of a dependence universe $\mathfrak{U}$, a dependence model $\mathbf{M} \in \mathfrak{U}$, and one or more binary relations on dependence models $S \subseteq \mathfrak{U} \times \mathfrak{U}$, expressing relevant changes of the dependence models. It contains the dependence modalities interpreted as before, but also modalities accessing alternative models via the relations S.

Example 6.18. Truth conditions for a bimodal epistemic dependence language:

$\mathfrak{U}, \mathbf{M}, S, s \models \mathbb{D}_X \varphi$ iff $\mathfrak{U}, \mathbf{M}, S, s' \models \varphi$ holds for all $s' \in A$ with $s =_X s'$

$\mathfrak{U}, \mathbf{M}, S, s \models \langle S \rangle \varphi$ iff $\mathfrak{U}, \mathbf{M}', S, s \models \varphi$ holds for some $\mathbf{M}' \in \mathfrak{U}$ with $S \mathbf{M} \mathbf{M}'$ and $s \in A'$

A natural relation S for exploring dependence universes is that of *submodel*. Call its downward-looking modality $[\downarrow]\varphi$. Now part of the semantics of LFD can be internalized in this richer logic.

Example 6.19. *The following two principles are valid in dependence universes:*

(a) $D_X y \to [\downarrow] D_X y$, *(b) $\mathbb{D}_X [\downarrow] \varphi \to [\downarrow] \mathbb{D}_X \varphi$.*

Open problem *What is the complete logic of LFD plus the downward and upward submodel modalities on dependence universes?*

This concludes our exploration of logical operators that extend the basic language of LFD.

---

[51]The general fact here is that F is a modal 'p-morphism', cf. [22], in a sense appropriate to dependence models.

[52]In an epistemic perspective, gaps encode information about how the process of inquiry may go. Related structures occur in dynamic-epistemic logic under the heading of 'protocol models', [15]. One can think of protocols with gaps as higher-order dependencies or correlations on how information can be acquired.

7 Dependence in concrete settings

The dependence semantics and logic of this paper are simple, and many notions of dependence in actual use add further features. This section presents a few cases, mainly to show that they fit with the basic LFD perspective, while also highlighting their interesting more specialized structures that call for further logical investigation.

7.1 Databases

This paper started with a simple database example, which nevertheless does not do justice to the more sophisticated structures studied in database theory, [1]. Much of this theory is in terms of first-order logic and its low-complexity fragments, and in this light, LFD can be seen as an attempt at capturing some high-level features of databases in a modal style. Indeed, various kinds of dependence and independence in databases can be represented in LFD-style languages, especially with the extensions introduced in Section 6.

As a further point, databases consist of 'facts' and 'rules'. Rules are hard-wired regularities, telling us how to close the database under inferences. For semantic dependencies in a model, this suggests a natural distinction: some are 'accidental', others are 'essential'. This distinction cannot be seen inside dependence models, it requires an additional external decision which regularities are important and which ones are not. A semantic setting for getting at the distinction are the dependence universes of Section 6.6. Accidental dependencies $D_X y$ just hold in the current model, while essential ones continue to hold even under relevant updates of that model, which can be expressed using modalities such as $[\downarrow]\varphi$ and $[\uparrow]\varphi$.

7.2 Vector spaces

The next example comes from linear algebra where dependence is the fundamental notion behind a wide range of applications to computation, defining geometrical dimension, and much more. A vector y depends on a set of vectors $\mathbf{x}$ if y can be written as a linear combination of the $z \in \mathbf{x}$. This notion is not primarily semantic, but it rather ties in with the equivalent functional definability perspective on dependence of Section 2.3. However, given the special operations used in linear algebra, there are interesting valid properties beyond those provided by LFD.

Example 7.1. The Steinitz Exchange Principle, [55], reads as follows in the LFD language:

$$D_{(X \cup \{y\})} z \to (D_X z \vee D_{(X \cup \{z\})} y)$$

The reason for its validity in linear algebra is that, if z is a linear combination of X, y, then either the coefficient for y is 0, and the first disjunct holds, or that coefficient is not 0, and then one can divide by it, obtaining a formula expressing y as a linear combination of X, z.

The Steinitz principle is not valid in LFD: a counter-example on dependence models occurs in Example 2.8, where we have $D_{\{z,x\}} y, \neg D_z y$ and $\neg D_{\{z,y\}} x$. However, in the spirit of Section 5.3, one can ask for a modal *correspondence* result: which constraint on assignments in dependence models ensures the validity of the Steinitz principle? What comes to mind is a principle about existence of inverses for implicit functions that truly depend on their arguments, but a precise solution remains to be found. In addition, there is a natural question of axiomatization.

Open problem *Axiomatize the complete theory of LFD-style assertions about dependence between vectors. Is it just the basic proof system* **LFD** *plus Steinitz Exchange?*

Remark 7.2. *Matroid Theory* studies abstract linear dependence and independence. Matroids are finite families of sets of vectors satisfying conditions implying the uniqueness of finite dimension. Matroids can be represented as dependence frames for LFD, [37], but there is an issue of the best logical framework. In the matroid setting, *sets* of variables are the central notion, and LFD does not describe such sets in an abstract algebraic way, except by brute enumeration. It would be of interest to develop a modal perspective on Matroid Theory.

7.3 Topologizing LFD: the logic of continuous dependence

In empirical contexts, the exact values of most variables are never accessible. Then, the existence of a functional dependence in the sense of LFD is a moot point, of only theoretical importance. What matters is whether there is a *knowable* dependence: given what can be known in principle, by measurements of any precision, about the value of x, can the value of y be found with any desired degree of precision?

Making sense of this intuition calls for a *topological* setting, with its intuitions of approximation and continuity. This section outlines such a *logic of continuous dependence* LCD, though a full presentation and development is postponed to our forthcoming paper [8].

A variable y *depends continuously on* x at an assignment s if the value of y at s is determined to any desired degree of approximation by some (possibly better) degree of approximation of the value of x at s. Epistemically, this means that one can know the value of y with any desired accuracy, if given a sufficiently accurate estimate of the value of x.[53]

This suggests having a topology τ on the set of objects O, to capture approximations of values $s(x)$ as open neighborhoods $U \in \tau$ with $s(x) \in U$. Global dependence $D_x^{\mathbf{M}} y$ in such a *topo-dependence model* $\mathbf{M} = (M, A, \tau)$ requires existence of a continuous map from x-values to y-values, while local dependence $D_x^s y$ is given by:

$$s \models D_x y \quad \text{iff} \quad \forall V \in \tau(s(y)) \, \exists U \in \tau(s(x)) \, \forall t \in A \, (t(x) \in U \to t(y) \in V),$$

where $\tau(o) = \{U \in \tau : o \in U\}$ is the family of open neighborhoods of an object $o \in O$. This can be generalized to dependence $D_X y$ on a *set* X of variables, by using the *product topology* on $O^{|X|}$. Intuitively, $D_X y$ holds at an assignment s if all assignments that assign to X values that are close enough to their current ones also assign to y a close enough value to its current one.

The natural analogue semantic clause for simple dependence modalities is:

$$s \models \mathbb{D}_x \varphi \quad \text{iff} \quad \exists U \in \tau(s(x)) \, \forall t \in A \, (t(x) \in U \to t \models \varphi).$$

The definition can be generalized to set-based dependence modalities $\mathbb{D}_X \varphi$ using a product topology, but we skip details here. Intuitively, $\mathbb{D}_x \varphi$ holds at an assignment s if φ holds at all admissible assignments that assign to x values that are 'close enough' to their current value $s(x)$. This connects to the well-known topological-interior semantics for the modal logic $S4$: an assignment s satisfies $\mathbb{D}_x \varphi$ iff the current x-value $s(x)$ is in the *interior* of the set $\{t(x) : t \models \varphi\}$ of all x-values of φ-assignments. Philosophically, the interior semantics points to an *evidential conception of knowledge*: φ is knowable from X if there exist some potential pieces of evidence about X that entails φ.

---

[53]A stronger notion, perhaps closer to the stated epistemic intuition about approximation, would be continuous dependence at some open neighborhood of the current point. This can be defined in the language to follow.

As for defined notions, $A\varphi := \mathbb{D}_\emptyset \varphi$ is still the universal modality over all assignments in A. Global dependence, defined as before by $AD_x y$, now expresses the existence of a *continuous* map $f : O \to O$ with $s(y) = f(s(x))$ for all $s \in A$. Other defined operators acquire a different meaning. The formula $D_\emptyset y$ used to mean in LFD that y is constant, taking only one value. But in a topological setting it expresses a more complex condition on the 'specialization pre-order', which only reduces to constancy in the presence of the separation axiom T_1.

More details, including decidability and a complete axiomatization, as well as further extensions to include uniform continuity and links with Domain Theory, will be presented in [8]. For now, we note that the proof calculus for LCD involves modal logic S4 rather than S5 for its dependence modalities. Moreover, even LFD principles that remain valid as they stand now express something subtly different in a topological setting. In particular, the Transfer Axiom $(D_X Y \wedge \mathbb{D}_Y \varphi) \to \mathbb{D}_X \varphi$ turns out to capture the *continuity* of dependence.[54]

Remark 7.3 (Point-free alternatives). Dependence in the logic LCD strengthens the notion of dependence in LFD: the functions made explicit in Section 2.3 are now to be *continuous*. But the intuitions behind the topological view seem independent from the existence of point-to-point functions. They rather talk about *correlating evidence*, i.e., open sets, whether or not there is some underlying set of sharp limit points and functions between these. The better framework, then, might be a *point-free* topology, with the notion of dependence suitably adapted to direct correlations between open sets that induce continuous functions under some appropriate mathematical construction of points.

7.4 Dynamical systems

Many real-life dependencies have a temporal aspect. Even the simple propositional example of Remark 6.8 suggests a network dynamics where propositions can become true or false, and dependencies involve a time delay. [55] This suggests a temporal universe of assignments occurring over time, with dependencies such as

$$s_{t+1}(y) = s_t(x).$$

Now one might reduce this to a static setting by adding temporal variables, using function terms as in Section 6.1. But it seems more natural to turn dependence models into *dynamical systems* where assignments are global states that can occur and repeat over the permissible evolutions of the system. A logic for this should combine LFD with a temporal language.

Consider a dependence model $\mathbf{M} = (M, A)$ with an assignment-changing next-state map $g : A \to A$. The *dynamical system defined by* $(\mathbf{M}, g)$ is the family of functions $\{g^n\}_{n \in N}$, with g^n the n-fold composition of g with itself. A simple language of dynamic dependence adds three items to the syntax of LFD: a *next* operator $\bigcirc \varphi$, an *n-th step dependence* operator $D_X^{(n)} y$ for each $n \in N$, and a *henceforth* operator $*\varphi$. Their semantics has these clauses:

$$s \models \bigcirc \varphi \ \text{ iff } \ g(s) \models \varphi$$

---

[54]In the topological reading, the Transfer Axiom literally tells us that, if there is a continuous function F mapping X-values into the corresponding Y-values, then $s(Y) \in Int\{t(Y) : t \models \varphi\}$ implies $F^{-1}(s(Y)) = s(X) \in Int\{t(X) : t \models \varphi\} = Int(F^{-1}\{t(Y) : t \models \varphi\})$ for all φ. This is the syntactic counterpart of the topological definition of continuity: inverse images of open sets by continuous maps are open.

[55]The same is true for situation-theoretic scenarios of information flow, [11], [19], and for ubiquitous strategies in iterated strategic games, such as Tit-for-Tat or Copy-Cat: what you do now is what I will do next, [64].

44

$$s \models D_X^{(n)} y \ \text{ iff } \ s =_X t \text{ implies } g^n(s) =_y g^n(t) \text{ for all } t \in A$$

$$s \models *\varphi \ \text{ iff } \ s \models \bigcirc^n \varphi \text{ for all } n \in N \quad (\text{with } \bigcirc^n \varphi \text{ the } n\text{-th iteration of } \bigcirc)$$

In particular, $D_X^{(1)} y$ says that the current values of the variables in X uniquely determine the next-step value of y. This is just what is needed to formalize Tit-for-Tat or Copy-Cat.[56]

As for valid reasoning, dynamic analogues of Reflexivity, Monotonicity and Transitivity are easy to formulate. There is also a valid dynamic analogue of the Transfer Axiom:

$$\left(D_X^{(n)} Y \wedge \bigcirc^n \mathbb{D}_Y \varphi \right) \to \mathbb{D}_X \bigcirc^n \varphi$$

Open problem *Axiomatize dynamic dependence logic completely. Is this logic decidable?*[57]

Remark 7.4 (Topology once more). Dynamical systems usually have a state space endowed with a topology. This richer setting gives rise to 'dynamical topo-dependence models' (M, A, τ) with a next-step map $g : A \to A$ that is continuous. Then, for instance, extending the system LCD with temporal operators, for a finite total set of variables V, the commutation axiom $\bigcirc \mathbb{D}_V \varphi \to \mathbb{D}_V \bigcirc \varphi$ expresses the continuity of the next-step function g.

7.5 Games

Dependence also occurs in game theory, [64], though with an additional flavor. While LFD speaks about dependence of *values*, game theory talks about dependence of *actions*. The notions are related, but games pose some interesting new features for logical dependence analysis, [16], [18].

Example 7.5. (Choice and dependence). Consider an extensive game of perfect information with two players A, E that have two moves 'left' and 'right' at each turn. A moves first, then it is E's turn. The four histories in the game tree can be viewed as assignments to two variables x, y, with x the action chosen by A, and y by E. The available actions for each player are independent from those of the other: both $I_x y$ and $I_y x$ hold in the sense of Section 6.3.

Now let E choose a strategy, i.e., in this simple game: a move at each of her two possible decision nodes. This restricted play introduces a functional dependence: the LFD statement $D_x y$, which was false before, will now come to hold. Thus, committing to a choice, or a strategy in general, changes the current dependence model for the game to one where appropriate dependence statements come to hold.[58]

Extensive game trees can be associated with dependence models whose variables stand for successive actions by the players.[59] Moreover, the action perspective introduces the dependence dynamics of Section 6.4. Making a choice *makes a dependence statement $D_X y$ true* by removing

---

[56] The syntax chosen here for purposes of illustration is a bit cumbersome. Direct functional notations, such as Ox for the value of x at the next state, will be more perspicuous in practice.

[57] In response to an earlier version of this paper, completeness and decidability results for the temporal dependence logic of dynamical systems have been claimed in [56].

[58] It is often said that committing to a strategy makes one's actions independent from those by the other player, since the strategy was chosen beforehand. However, this seems a confusion between pre-game deliberation and in-game play: as the game is played, a strategy does follow the particular moves chosen by the other players.

[59] To make this work, some issues have to be solved, since strategies produce forests rather than sub-trees, [16]. Also, in extensive games, variables should depend on 'earlier' variables, not on those for later stages.

some assignments from a given model $\mathbf{M}$, to obtain a submodel $\mathbf{N}$ satisfying (a) the statement $D_X y$, but also (b) the following 'X-richness' constraint, for the given set $X \subseteq V$ of variables:

$$\mathbf{M} \restriction X = \mathbf{N} \restriction X$$

(where $\mathbf{M} \restriction X$ is the restriction of all assignments in $\mathbf{M}$ to the domain X). The 'X-rich submodels' $\mathbf{N}$ satisfying this constraint can be viewed in both first-order and modal terms. An interesting question is which syntactic types of statement are preserved when moving from $\mathbf{M}$ to $\mathbf{N}$.

Open problem Develop the dynamic dependence logic of strategic choice.

Next consider extensive games with *imperfect information*. Here is a simple illustration.

Example 7.6 (Imperfect information games). In Example 7.5, now assume that E cannot observe A's move. Then E's epistemic uncertainty relation holds between the two mid-points of the game tree. This game has been discussed widely for its combination of action and knowledge, a typical feature of games with imperfect information and their links with modal logics, [16].

In game theory, a strategy must be *uniform*, assigning the same move at points that E cannot distinguish epistemically. In the above example, this leaves only strategies 'always left' and 'always right'.[60] The intuition behind uniform strategies is that they can only appeal to things that players *know*. This knowledge is encoded by the current equivalence class of their epistemic equivalence relation. This is LFD dependence combined with the epistemic representation in Section 3.4. The choice of move by a strategy depends on the variable for the agent's knowledge state. In general, this perspective will work with distinct variables for agents and moves, corresponding to the knowledge and the action modalities whose interplay is crucial to reasoning about games with imperfect information.

But, if games are very regular, say, just choosing values for stage variables $x, y, \dots$ from some fixed set, epistemic uncertainty relations match up directly with dependence relations for sets of variables. Assume, as is common in epistemic-temporal logic, [65], that players can observe some events that have taken place, but not others. Then their equivalence relation on histories will be equality for the values of their *observable variables* only. In games, strategies for a player now have to choose actions that depend in the LFD sense on the values of the observed variables for that player. In general, in this sort of two-player game with imperfect observation, players A and E partition all the variables.[61]

This discussion by no means exhausts the topic of games with either perfect or imperfect information from a dependence-logical perspective, and in general, as stated before, we will need combinations of epistemic logic for players's knowledge and LFD for their actions.

7.6   Causality

A final important arena for dependence is causality. *Causal graphs*, [66, 43], impose correlations between variables, restricting the simultaneous assignments of values that represent possible states of world. This is reminiscent of the dependence graphs in Section 2, and indeed, one common notion of 'causal influence' of a variable x on an endogenous variable y found in [66] can be simply represented in LFD as $D_{V-\{y\}} y \wedge \neg D_{V-\{y,x\}} y$, with V the set of all variables. But the match is not

---

[60]In contrast, in IF Logic these strategies are said to act *independently* from what player A does, [47].

[61]There are also several richer logical languages at the interface of imperfect information games and combined epistemic-dependence models: see [16], Ch. 21, and [18].

one-to-one. Not all relational facts in dependence graphs represent causal connections: singling out the truly causal ones requires a separate decision. Vice versa, causal graphs do not have a unique associated dependence model: they are schemata for many models. Even so, the representation in Section 2 may extend to causality, now also analyzing various types of dependencies that can occur in dependence models.

Conversely, several themes in the theory of causal graphs resonate in the present framework. For instance, LFD with function terms may be considered a modal companion to the logic for causality in [43], that manipulates explicit equations between variables in causal graphs. Also, the crucial notion of 'interventions' in causal graphs has an obvious counterpart in updates of dependence models that fix values, as in Section 6.5. Even so, there may be an essential surplus to the notion of causal dependence that transcends the resources of the LFD framework. In this sense, see [9, 10] for a formalism that combines features of Dependence Logic with an interventionist approach to causality, and see [76] for a similar combination of epistemic logic and causal models.

Many further concrete notions of (in-)dependence occur in the literature. There is essential dependence and independence in natural language, [47], metaphysics, [34], [50], proof theory, [70], ceteris paribus reasoning, [17], social choice theory, logics of agency, and many other fields. A complete list is beyond the scope of this paper, but a confrontation with LFD seems worthwhile in many of these cases.

8   Related work

In this section, some of many other approaches to dependence are listed in historical order, with comments on connections to the LFD framework.

Armstrong axioms. The basic structural properties of functional dependence used in this paper were identified by Armstrong [5], in the form of the postulates of Inclusion (cf. Definition 2.4, Example 5.2a), Transitivity (cf. Definition 2.4) and Additivity (cf. Example 5.2b). By Fact 2.5, the first two together are equivalent with the conjunction of our Projection and Transitivity properties (as well as with the conjunction of Reflexivity, Monotonicity and Transitivity), while Armstrong's Additivity is absorbed into our definition of $D_X Y$ as an abbreviation for $\bigwedge_{y \in Y} D_X y$. Armstrong gave a representation theorem showing that these axioms are complete for (global) database dependence. Section 2 of this paper presents a different proof, yielding a stronger representation theorem (Proposition 2.7), for both global and local dependence. Similar abstract structural axioms for independence, given in [36], underlie the modal independence logic in Section 6.3.

CRS logic. As explained in our Introduction, LFD is a direct continuation of generalized assignment semantics CRS for first-order logic, for which we have given several references. The origins of CRS lie in relational and cylindric algebra, [63]. The decidability of CRS can be shown by first-order translation into the 'Guarded Fragment' GF, [3], while the first-order translation for LFD in Section 3.2 does not map into GF. As we have noted, it is an open problem whether one can prove decidability for LFD via a known decidable fragment of FOL.

Independence-friendly logic. Dependence pervades game-theoretic semantics for logical systems. Strategies in evaluation games for FOL correspond with Skolem functions that express dependence in the sense of Section 2.3. A further innovation was 'Independence-Friendly Logic' (IF-logic, for short), [47], where the player for the existential quantifier may have imperfect information about the objects chosen by the player for the universal quantifier, cf. Section 7.5. A

compositional semantics for IF-logic uses evaluation on sets of assignments, [48], allowing for choices of values independently from the values for specified other variables. [62] These sets are like LFD dependence models, but without designated single assignments and local dependence. Moreover, in contrast with LFD, IF-logic is second-order and non-axiomatizable. For a complete mathematical development of IF-logic, see [58].

A comparison between LFD and IF-logic poses a challenge, already noted for CRS vs. IF-logic in [13]. IF-logic sees first-order logic as tied to linear *dependencies* between quantifiers, and incorporates 'branching quantifiers', thereby moving up to second-order complexity. In contrast, CRS sees FOL as too much tied to *independence*, and weakens it to a decidable logic that allows for both dependence and independence of variables. One obvious difference is that IF-logic takes standard FOL as is, and adds syntax for independence. We made some remarks on the connection of LFD with FOL in Section 3.2, and we have more precise results – but a deeper treatment is a topic for a separate paper. But perhaps more importantly here, in the terminology of Section 7.5, while LFD analyzes what might be called *value dependencies* between variables, IF-logic describes what might be called *choice dependencies* between quantifiers. It is easy to see formally that LFD cannot express choice dependencies, and our brief discussion of games showed that we would need additional modalities over dependence universes. Even so, LFD and IF-logic also share some traits, and cross-overs between the two are worth exploring.

For instance, one can *enrich* LFD with natural forms of branching quantification on dependence models. For instance, the natural reading of the simplest Henkin formula

$$\begin{bmatrix} \forall x_1 & \exists y_1 \\ \forall x_2 & \exists y_2 \end{bmatrix} P x_1 x_2 y_1 y_2$$

in a dependence model $\mathbf{M} = (O, I, A)$ is the assertion that

$$\exists F : O^{(x_1)} \to O \, \exists G : O^{(x_2)} \to O \, \forall (o_1, o_2) \in O^{(x_1, x_2)} \, (o_1, o_2, F(o_1), G(o_2)) \in I(P) \cap O^{(x_1, x_2, y_1, y_2)}.$$

In other words: the witnessing functions have domains restricted to the corresponding admissible values and return a tuple combination that not only satisfies P, but is actually realized by some admissible assignment of values to (x_1, x_2, y_1, y_2). This seems to be the natural generalization to branching quantifiers of the semantic reading of the LFD formula $\forall_x \exists_y P x y$ explained in Section 3.2. Similarly, one can define LFD versions of the slash quantifiers from IF-logic. As happened in the case of CRS versus FOL as well as with other logical systems, [4], this strategy of generalizing to models with admissible tuples of values might well lower the complexity of IF-logic. But again, we leave details to a further publication.

Independence and randomness. An innovative abstract first-order logic for probabilistic independence is presented in [55], emerging from the study of randomness. The calculus contains several axioms at the abstraction level of LFD, but also more specialized principles such as the Steinitz Axiom for linear dependence discussed in Section 7.2. For another broad approach to elementary qualitative principles for dependence and independence, see the measurement-theoretical analysis of probabilistic reasoning in [62].

Dependence logic. Väänänen's dependence logic DL [71] was the first to introduce explicit dependence atoms (for global dependence), a crucial device that we have adopted in LFD. The

---

language of DL is an extension of the language of first-order logic, but interpreted over *sets* of assignments (called 'teams'), instead of single assignments (as in FOL), or combinations of a current assignment and a surrounding team (as in LFD). This 'set lifting' semantics was first suggested in this context in [48].[63] Interpreting on sets of assignments *lifts* the meanings of standard propositional connectives, resulting in a richer vocabulary with a non-classical logic. Moreover, the DL interpretation of the first-order quantifiers ranges over sets of assignments, yielding a form of second-order quantification. By now, there is an extensive body of theory on variations, extensions, and fragments of the DL framework, which we cannot survey here. The reader is referred to the original source [71] and to the extensive survey article [35] in the Stanford Electronic Encyclopedia of Philosophy.

Comparing LFD with DL, one striking difference is between the 'team semantics' for DL on sets of assignments with global functional dependencies, and the local semantics of LFD with single assignments inside teams, giving the central place to local dependence. This difference may seem slight, but as observed earlier, the set lifting brings with it some pressure towards non-classical logics.[64] Another major difference is the view of quantifiers, as briefly discussed above in connection with IF-logic, and from a more classical angle, in Section 3.2. LFD quantifiers range over values available inside one dependence model, thus respecting all current dependence constraints. In contrast, DL quantifiers can evaluate in *new* teams (i.e., *other* dependence models), thus 'freeing' the quantified variables from the constraints of the old team. From the minimal standpoint of LFD, such 'freeing quantifiers' are naturally viewed as composites of *two* different notions of logical interest: a dynamic modality for relevant kinds of model change in a dependence universe, followed by a model-internal LFD quantifier.[65]

Both views have their attractions. The LFD quantifiers fit well with an *epistemic* interpretation (in which the current team comprises all the possibilities compatible with one's background knowledge), as well as with applications to *complete databases* (where the current team stands for a complete state space, listing all the states that can be generated by some dynamic process). Other dependence models then only come in via informational update, or via process change. But in many other settings, e.g., applications to more general *partial* databases, dynamical updating, open systems, etc., the stronger DL quantifiers will be just what is needed, capturing interesting properties that go beyond the resources of LFD.[66]

Finally, in terms of technical comparisons, there are various questions that can be asked. One is the point, already mentioned in the Introduction, that the lower complexity of LFD (inevitably) comes at a price of lower expressive power. Thus, it would be of interest to match LFD (minus local dependence atoms) and its extensions with some low-complexity fragments of DL, and in particular fragments with restricted forms of quantification. Given the modal nature of our formalism, it might seem at first sight that the appropriate comparison is with the system of 'Modal Dependence

---

[63]Set lifting as a general device has a long history in logical semantics, resulting in the theory of 'complex algebras' which distinguish, amongst other things, different 'inner' and 'outer' variants of Boolean operations, [24]. E.g., the disjunction of DL is the inner version of Boolean union.

[64]For a system preserving classical semantics in a set-lifted setting, cf. the possibility semantics of [49].

[65]Valid laws for 'freeing quantifiers' depend on the model change relation chosen plus the choice of dependence universes. E.g., quantifiers become second-order when the dependence universe contains all possible variants of the current dependence model, i.e., all sets of assignments. But as remarked in Section 6.4, one might also allow gaps in dependence universes, creating higher-level dependencies, and lowering complexity of the logic.

[66]Many further themes in this paper have counterparts in the literature on more expressive DL formalisms. For instance, dependence plus independence in a sense close to that of Section 6.3 is studied in [41] and its follow-up literature, cf. [35]. In addition, studies of dependence with a classical logic base are found in [38], [51].

Logic' in [72], or other propositional logics of dependence [73]. However, both of these are purely propositional languages, with no variables over objects and no quantification over them, however restricted. In fact, since CRS is known to have tight connections with the Guarded Fragment of FOL, it would be more natural to expect its extension LFD with dependence atoms to have interesting connections with some corresponding fragment of DL. As a particular instance, how is LFD related to the recent Guarded Fragment versions of DL introduced in [40]? Another interesting line to pursue might be the earlier-mentioned 'deconstruction' of DL quantifiers into dynamic modalities plus LFD modalities, which could lead to a richer intermediate theory with various modalities over dependence universes. Indeed, it seems worthwhile to look for formalisms in between LFD and DL. The dependence universes in Section 6.4 are an instance, since they represent a Henkin move of not considering all sets of assignments, but just certain subfamilies, creating what might be called higher-order dependencies when moving between dependence models. We suspect that this will make the logic first-order, since one can describe this setting in a *three-sorted first-order language* with variables over objects, assignments, and dependence models.[67]

Logics of questions. We have noted at various places that LFD has informational interpretations in terms of implications between questions, as discussed in a general dynamic-epistemic setting in [6]. The logic of questions has a long history with classical sources such as [12] and [46]. The handbook article [44] surveys many themes on the logic side, and [42] surveys themes focusing on natural language semantics. For a modern perspective on dependence in terms of inquisitive logic of questions, cf. [25]. For a comprehensive treatment of inquisitive logic, we refer to the modern source [26].

Extended epistemic logic. Connections between LFD and epistemic logics were explained in Section 4. We cannot survey all points of contact, but general background can be found in [27]. As a special case, [57] is an early study of epistemic models with our equality-based accessibility relations. Many of our dependence themes are reflected in analogies between LFD and recent work on 'extended epistemic predicate logics' where agents can know not just propositions, but also objects, [67, 75, 6]. These logics add 'knowing wh'-constructions to propositional 'knowing that', and can be seen as well-chosen often decidable fragments of epistemic predicate logic. Specific analogies include our theme of functional definability in Section 2.3 and 'knowledge of functions', studied in [30], the conditional knowledge of objects and facts by epistemic agents in Section 6.4 and [75, 6], and on logics for public 'inspection' of values, [32].

9   Conclusion

Dependence has a ubiquitous semantic sense of determination of values for some variables by those of others. We have presented a decidable classical logic LFD for reasoning about functional dependence, together with complete axiomatizations. The proofs come in both first-order and modal style. Conceptually, these two complementary perspectives connect to the two manifestations of dependence highlighted throughout this paper: 'ontic' in the world or in some dynamical system, and 'informational' connecting to knowledge and questions. Further language extensions, as well as richer semantical settings, have been discussed in some detail.

Many open problems have been identified in this extension process, reflecting mainly its semantic and model-theoretic spirit. But we have also shown that there is room for a purely proof-theoretic

---

[67]This is just one option. See [53] for an alternative way of reducing the complexity of DL.

analysis of LFD and its extensions, and perhaps as a compromise between the model theory and proof theory: an analysis in *universal algebra* would be illuminating.

Going beyond these standard logical perspectives, one can think of dependence *information-theoretically*, in terms of values of dependent variables adding no Kolmogorov complexity to the given ones. But perhaps the greatest challenge left unaddressed here is tying the qualitative logical LFD analysis to *probabilistic* notions of correlation and dependence.[68]

A point of entry may be the analogy of dependence with consequence relations noted in Section 2. LFD-style dependence goes by universal quantification over all assignments. But as we observed, one can soften this, as in non-monotonic default logics, by going to models where the semantic dependence holds only in the most plausible cases, or only with high probability in some qualitative sense, [31]. In that case, the agenda for LFD becomes wide open again.

References

[1] S. Abiteboul, R. Hull & V. Vianu, 1994, *Foundations of Databases*, Pearson, London.

[2] S. Abramsky & J. Väänänen, 2009, From IF to BI: A Tale of Dependence and Separation, *Synthese* 167:2, 207–230.

[3] H. Andréka, J. van Benthem & I. Németi, 1998, Modal Languages and Bounded Fragments of Predicate Logic, *Journal of Philosophical Logic*, 27:3, 217–274.

[4] H. Andréka, J. van Benthem, N. Bezhanishvili & I. Németi, 2014, Changing a Semantics: Opportunism or Courage?, in M. Manzano, I. Sain & E. Alonso, eds., *The Life and Work of Leon Henkin*, Birkhaueser Verlag, 307–337.

[5] W. Armstrong, 1974, Dependency Structures of Database Relationships, *Proceedings IFIP Conference*, 580–583.

[6] A. Baltag, 2016, To Know is to Know the Value of a Variable, *Adv. in Modal Logic 2016*, 135–155.

[7] A. Baltag, L. Moss & S. Solecki, 1998, The Logic of Public Announcements, Common Knowledge, and Private Suspicions, *Proceedings TARK 98*, 43–56.

---

[68]There might even seem to be an essential mismatch, as *independence* is complexity-increasing in LFD, while it is complexity-decreasing in probabilistic computation. Compare the undecidability of modal logics that have commutation axioms, with the beneficial use of commutation results like Fubini's Theorem in probabilistic reasoning. This mismatch dissolves, however, by making a distinction. Regular mathematical structures simplify computation, but their logical *theory* is more complex than that of arbitrary structures.

[8] A. Baltag, A & J. van Benthem, 2020. The Logic of Continuous Dependence and Knowability. Manuscript, ILLC, University of Amsterdam.

[9] F. Barbero & G. Sandu. Interventionist Counterfactuals on Causal Teams. Proceedings 3rd Workshop on Formal Reasoning about Causation, Responsibility, and Explanations in Science and Technology, *Electronic Proceedings in Theoretical Computer Science* 286: 16–30, 2019.

[10] F. Barbero & G. Sandu. Team semantics for interventionist counterfactuals: observations vs. interventions. To appear in the *Journal of Philosophical Logic*, 2020.

[11] J. Barwise & J. Seligman, 1995, *Information Flow. The Logic of Distributed Systems*, Cambridge University Press, Cambridge UK.

[12] N. Belnap & Steele, 1976, *The Logic of Questions and Answers*, Yale University Press, New Haven.

[13] J. van Benthem, 1996, *Exploring Logical Dynamics*, CSLI Publications, Stanford University.

[14] J. van Benthem, 2005, Guards, Bounds, and Generalized Semantics, *Journal of Logic, Language and Information*, 14:3, 263–279.

[15] J. van Benthem, 2011, *Logical Dynamics of Information and Interaction*, Cambridge University Press, Cambridge UK.

[16] J. van Benthem, 2014, *Logic in Games*, The MIT Press, Cambridge MA.

[17] J. van Benthem, P. Girard & O. Roy, 2009, Everything Else Being Equal: A Modal Logic for Ceteris Paribus Preferences, *Journal of Philosophical Logic* 38:1, 83–125.

[18] J. van Benthem & D. Klein, 2019, Logics for Analyzing Games, *Stanford On-Line Encyclopedia of Philosophy*.

[19] J. van Benthem & M-C Martinez, 2008, The Stories of Logic and Information, *Handbook of the Philosophy of Information*, Elsevier, Amsterdam, 2017–2080,

[20] J. van Benthem & S. Minica, 2012, Toward a Dynamic Logic of Questions, *Journal of Philosophical Logic*, 41:4, 633–669.

[21] M. van den Berg, 1996, *The Internal Structure of Discourse*, Dissertation 96-03, ILLC, University of Amsterdam.

[22] P. Blackburn, M. de Rijke & Y. Venema, 2000, *Modal Logic*, Cambridge University Press, Cambridge.

[23] G. Brewka, 1991, *Nonmonotonic Reasoning: Logical Foundations of Commonsense*, Cambridge University Press, Cambridge.

[24] Ch. Brink, 1993, Power Structures, *Algebra Universalis* 30, 177–216.

[25] I. Ciardelli, 2016, Dependency as Question Entailment, in *Dependence Logic: Theory and Applications*, Springer, 129–182.

[26] I. Ciardelli, J. Groenendijk & F. Roelofsen, 2019, *Inquisitive Semantics*, Oxford University Press, Oxford.

[27] H. van Ditmarsch, J. Halpern, W. van der Hoek & B. Kooi, 2015, *Handbook of Epistemic Logic*, College Publications, London.

[28] H. van Ditmarsch, W. van der Hoek & B. Kooi, 2007, *Dynamic Epistemic Logic*, Springer Science Publishers, Dordrecht.

[29] H. van Ditmarsch, W. van der Hoek & B. Kooi, 2009, Knowing More - from Global to Local Correspondence, *Proc. of IJCAI-09*, 955–960.

[30] Y. Ding, 2016, Epistemic Logic with Functional Dependency Operator, *Studies in Logic*, 9:4, 55–84.

[31] Y. Ding, W. Holliday & Th. Icard, 2020, Logics of Imprecise Comparative Probability, to appear in *Journal of Approximate Reasoning*.

[32] J. van Eijck, M. Gattinger & Y. Wang, 2017, Knowing Values and Public Inspection, *Proceedings 7th Indian Conference on Logic and its Applications*, Kanpur, 77–90.

[33] R. Fagin, J. Halpern, Y. Moses & M. Vardi, 1995, *Reasoning About Knowledge*, The MIT Press, Cambridge MA.

[34] K. Fine & N. Tennant, 1983, A Defense of Arbitrary Objects, *Aristotelean Society Supplementary Volume* 57:1, 55–90.

[35] P. Galliani, 2018, Dependence Logic, *Stanford Encyclopedia of Philosophy*, Stanford Univ.

[36] D. Geiger, A. Paz & J. Pearl, 1991, Axioms and Algorithms for Inferences Involving Probabilistic Independence, *Information and Computation*, 91:1, 128–141.

[37] D. Gonzalez, 2019, An Exploration of Matroids and Modal Logic, Dept of Philosophy, Stanford Univ.

[38] V. Goranko & A. Kuusisto, 2018, Logics for Propositional Determinacy and Independence, *The Review of Symbolic Logic*, 11:3, 470–506.

[39] E. Graedel, 1999, On the Restraining Power of Guards, *Journal of Symbolic Logic*, 64, 1719–1742.

[40] E. Graedel & M. Otto, 2020, Guarded Teams: The Horizontally Guarded Case, *Proceedings 28th Annual Conference on Computer Science Logic*, Leibniz Center for Informatics, Schloss Dagstuhl, 22:1–22:17.

[41] E. Graedel & J. Väänänen, 2013, Dependence and Independence, *Studia Logica*, 101:2, 399-410.

[42] J. Groenendijk & M. Stokhof, 1997, Questions, *Handbook of Logic and Language*, Elsevier, Amsterdam, 1009–1053.

[43] J. Halpern, 2016, *Actual Causality*, The MIT Press, Cambridge MA.

[44] D. Harrah, 2002, The Logic of Questions, *Handbook of Philosophical Logic*, Springer, Dordrecht, 61–145.

[45] L. Henkin, D. Monk & A. Tarski, 1971, *Cylindric Algebra*, Part I, North-Holland, Amsterdam.

[46] J. Hintikka, 1976, *The Semantics of Questions and the Questions of Semantics*, North-Holland, Amsterdam.

[47] J. Hintikka & G. Sandu, 1997, Game-Theoretical Semantics, *Handbook of Logic and Language*, Elsevier, Amsterdam, 361–410.

[48] W. Hodges, 1997, Compositional Semantics for a Language of Imperfect Information, *Logic Journal of the IGPL*, 5:4, 539–563.

[49] W. Holliday, 2020, Possibility Semantics, in *New Directions in Logic*, College Publications, London.

[50] L. Humberstone, 2019, Explicating Logical Independence, *Journal of Philosophical Logic*, 49 (2020), 135—218.

[51] L. Humberstone, 2019, Supervenience, Dependence, Disjunction, *Logic and Logical Philosophy*, 28:1, 3–135.

[52] D. de Jongh & L. Chagrova, 1995, The Decidability of Dependency in Intuitionistic Propositional Logic, *Journal of Symbolic Logic*, 60:2, 498–504.

[53] J. Kontinen & F. Yang, 2019, Logics for First-order Team Properties, *Proceedings 26th WOLLIC*, LNCS 11541, 392–413.

[54] R. Koudijs, 2020. Characterization Theorems for LFD, working paper, Institute for Logic, Language and Computation, University of Amsterdam.

[55] M. van Lambalgen, 1992, Independence, Randomness, and the Axiom of Choice, *Journal of Symbolic Logic*, 57:4, 1274–1304.

[56] D. Li, 2020, LFD for Dynamical Systems, working paper, Institute for Logic, Language and Computation, University of Amsterdam.

[57] A. Lomuscio & M. Ryan, 1998, Ideal Agents Sharing (Some!) Knowledge, *Proceedings ECAI 2008*, John Wiley, Hoboken NJ, 557–561.

[58] A. Mann, G. Sandu & M. Sevenster, 2011, *Independence-Friendly Logic*, Cambridge University Press, Cambridge UK.

[59] M. Marx, 2001. Tolerance Logic, *Journal of Logic, Language and Information*, 10:3, 353–373.

[60] M. Marx, 2006, Complexity of Modal Logic, *Handbook of Modal Logic*, Elsevier Science, Amsterdam, 139–179,

[61] M. Marx & Y. Venema, 1997. *Multi-Dimensional Modal Logic*, Springer, Dordrecht.

[62] L. Narens, 2007, *Theories of Probability. An Examination of Logical and Qualitative Foundations*, World Scientific, Singapore.

[63] I. Németi, 1985, The Equational Theory of Cylindric Relativized Set Algebras is Decidable, Preprint 63/85, Mathematical Institute, Hungarian Academy of Sciences, Budapest.

[64] M. Osborne & A. Rubinstein, 1994, *A Course in Game Theory*, MIT Press, Cambridge MA.

[65] R. Parikh & R. Ramanujam, 2003, A Knowledge-Based Semantics of Messages, *Journal of Logic, Language and Information*, 12, 453–467.

[66] J. Pearl, 2009, *Causality: Models, Reasoning and Inference*, Cambridge University Press.

[67] J. Plaza, 2007, Logics of public communications, *Synthese*, 158:2, 165–179.

[68] P. Pützstück, 2020, *Exploring LFD*, Bachelor Thesis, Department of Informatics, RWTH Aachen.

[69] D. Scott, 1971, On Engendering an Illusion of Understanding, *Journal of Philosophy*, 68:21, 787–807.

[70] A. Troelstra & H. Schwichtenberg, 2000, *Basic Proof Theory*, Cambridge University Press, Cambridge.

[71] J. Väänänen, 2007, *Dependence Logic: A New Approach to Independence Friendly Logic*, Cambridge University Press, Cambridge.

[72] J. Väänänen, 2008, Modal Dependence Logic, *New Perspectives on Games and Interaction (Texts in Logic and Games)*, Amsterdam University Press, 237-–254.

[73] F. Yang & J. Väänänen, 2016, Propositional Logics of Dependence, *Annals of Pure and Applied Logic*, 167(7): 557—589.

[74] Y. Venema, 1995, Cylindric Modal Logic, *Journal of Symbolic Logic* 60:2, 591–623.

[75] Y. Wang, 2018, Beyond Knowing That: A New Generation of Epistemic Logics, *Jaakko Hintikka on Knowledge and Game Theoretical Semantics*, Springer, Dordrecht, 499-533.

[76] K. Xie, *Where Causality, Conditionals and Epistemology Meet*, PhD dissertation, ILLC, University of Amsterdam, 2020.

Appendix A: Modal proofs of LFD decidability and completeness

To study LFD as a modal logic, we need to generalize the 'standard' relational models introduced in Section 3.4 to a wider class of relational models. Viewing LFD as a modal language in the usual sense, with modalities $\mathbb{D}_X \varphi$ and atomic formulas $P\mathbf{x}$ and $D_X y$, our general relational models will be just ordinary Kripke models for this language. This move allows us to apply to them well-known notions and methods in modal logic, such as p-morphisms, unraveling, and filtration. In the following we will assume familiarity with these standard modal techniques. See [22] for definitions and explanations.

So there are two main differences between general relational models and the standard models introduced earlier: (a) each relation $=_X$ for sets $X \subseteq V$ is taken as primitive, without being reduced to an intersection of basic relations $=_x$, and (b) $D_X y$ is treated as just another atom, whose semantics is given by a valuation (although one subject to restrictions).

A1. Relational semantics

Definition A.1. *A* relational model *is a structure* $\mathbf{M} = (A, =_X, D_X y, P\mathbf{x})$, *where: A is a set of possible worlds ("abstract assignments"); $=_X \subseteq A \times A$ are binary relations on worlds, one for each set $X \subseteq V$ of variables; $D_X^s y \subseteq \mathcal{P}(V) \times V$ are relations between sets of variables X and variables y, one for each world $s \in A$; and P^s are n-ary relations on variables, one for each n-ary predicate P and each world s. These ingredients are required to satisfy four conditions:*

(1) *all relations $=_X$ are equivalence relations on A;*

(2) *all relations D^s satisfy Projection and Transitivity;*

(3) *if $s =_X t$ and $D_X^s y$ for all $y \in Y$, then $s =_Y t$ and $D_X^t y$ for all $y \in Y$;*

(4) *if $s =_X t$ and $P^s \mathbf{y}$ for some $\{y_1, \ldots, y_m\} \subseteq X$, then $P^t \mathbf{y}$;*

(5) *$=_\emptyset$ is the global relation on A (relating every two worlds)*

The semantics of LFD on relational models is just as on dependence models, except that the abstract relations $s =_X t$, $D_X^s y$ and $P^s \mathbf{x}$ are used instead of their concrete counterparts.

Fact A.2. Standard *relational models in the sense of Section 3.4 are exactly those relational models satisfying the following two* additional conditions:

(5) *if $s =_X t$ and $s =_Y t$, then $s =_{X \cup Y} t$.*

(6) *if $s =_X t$ implies $s =_y t$ holds for all $t \in A$, then $D_X^s y$.*

A2. Equivalence between relational models and dependence models

We now show that *the logic of relational models is the same as the logic of dependence models.*

To go from dependence models to relational models: we can just use the equivalence between dependence models and standard relational models (cf. Fact 3.15 and Fact 3.17).

But to go the other way, from relational models to dependence models, we need a representation of relational models in terms of standard ones:

Proposition A.3. *Every relational model is a p-morphic image of some* standard *relational model (in the sense of Section 3.4).*

Proof. The proof is essentially a variation of modal *unravelling*, making infinitely many copies of each world.[69]

Let $\mathbf{M} = (A, =_X, D, P)$ be a relational model, and let $s_0 \in A$ be any designated world. To construct a standard relational model $\mathbf{M}^{st}$, take as *worlds* the set A^{st} of all 'histories', i.e. all finite sequences $h = (s_0, X^1, s_1, \ldots, X^n, s_n)$, with $n \geq 0$ and $s_o, \ldots, s_n \in A$ satisfying $s_{k-1} =_{X^k} s_k$ for all $k = 1, n$. We denote by $last(h) := s_n$ the last state in history h, and by $\rightarrow_X$ the natural *one-step relation* on histories, given by $h \rightarrow_X h'$ iff $h' = (h, X, s')$ (with $last(h) =_X s' = last(h')$). The one-step relations structure A^{st} can be viewed as a *tree* with root (s_0) (where s_0 is the designated world), in which any two nodes h, h' are connected by a unique non-redundant path.

To structure this as a relational model, we define a *new one-step relation* $\overset{=}{\rightarrow}_X$, incorporating all the one-step relations labelled by sets that locally determine X:

$$h \overset{=}{\rightarrow}_X h' \quad \text{iff} \quad h \rightarrow_Y h' \text{ for some } Y \text{ with } last(h) \models D_Y X.$$

Then the *required equivalence relations* $=_X$ on worlds/histories in A^{st} can be taken to be *the reflexive-transitive-symmetric closure of the relations* $\overset{=}{\rightarrow}_X$. To check the claims below, it may be useful to note that $h =_X h'$ holds iff the unique non-redundant path from h to h' consists only of steps of the form $h_n \overset{=}{\rightarrow}_{Y^n} h_{n+1}$, or $h_n \overset{=}{\leftarrow}_{Y^n} h_{n+1}$, with $last(h_n) \models D_Y X$.

Finally, the *valuation* on atoms is given by truth at the last world in the history (in the original model):

$$D^h_X y \quad \text{iff} \quad last(h) \models D_X y, \qquad P^h \mathbf{x} \quad \text{iff} \quad last(h) \models P\mathbf{x}.$$

The fact that this definition yields a standard relational model $\mathbf{M}^{st}$ is an easy verification.

To finish the proof, we define a map $f : A^{st} \to A$, by putting $f(h) := last(h)$ for all $h \in A^{st}$. It is easy to check that f is a surjective p-morphism f from $\mathbf{M}^{st}$ to $\mathbf{M}$. (Surjectivity follows from the fact that every world $s \in A$ satisfies $s_0 =_\emptyset s$, by condition 5 on relational models, hence $h = (s_0, \emptyset, s)$ is a history with $f(h) = last(h) = s$.) $\qquad \square$

Combining Fact 3.15, Propositions 3.17 and A.3, plus the preservation of modal formulas under surjective p-morphisms (and so under surjective homomorphisms), yields the following:

Corollary A.4. *(Modal equivalence of relational and dependence models) The same LFD formulas are valid on dependence models, relational models and standard relational models.*

---

[69]Note the similarity of the unraveled model in this proof and the tree construction in the proof of Theorem 4.8. Indeed, the two decidability proofs are based on similar ideas, but there are also notable differences. The proof in Section 4 is based on a syntactic construction ("type models") and is very elaborate; the proof of the key Truth Lemma is a very syntactic complex induction on formulas. In contrast, the proof in this section is purely semantic, and it offers a shortcut, by relying on known results and techniques in Modal Logic.

A3. Decidability via relational models

The preceding detour into abstract relational models and the above Corollary A.4 on modal equivalence can be used to give a second, more general proof of decidability using the Modal Logic concept of *filtration* [22].

Proposition A.5. *The language LFD has the Strong Finite Relational Model Property: if φ is satisfied in some relational model $\mathbf{M}$, then it is satisfied in a* finite *relational model, whose size is bounded by a computable function of φ. As a consequence, the logic LFD is* decidable.

Proof. Start with the singleton $F = \{\varphi\}$, and construct the finite set of formulas $\Phi = \Phi_F$ as in Section 4.1 (whose size was bounded by a computable function of φ).

The *filtrated model* $\mathbf{M}^f$ has as *worlds* the equivalence classes $[s]$ of original worlds $s \in A$ modulo Φ-equivalence $\equiv_\Phi$ (with respect to all formulas in Φ). Note that there are only finitely many such classes (their number is bounded by a computable function $F(\varphi)$).

To define the *relations* $=_X$ in the filtrated model, we take the following 'dependent filtration':

$$[s] =_X [t] \quad \text{iff} \quad (s \models \theta \text{ iff } s \models \theta) \text{ for all } \theta \in \Phi \text{ with } Free(\theta) \subseteq \{y \in V : s \models D_X y\}.$$

This is well defined (independent from the choice of representatives), and the definition implies that $\{y \in V : s \models D_X y\} = \{y \in V : t \models D_X y\}$ whenever $[s] =_X [t]$.

As for *valuation*: the truth values at $[s]$ for atoms $D_X y, P\mathbf{x} \in \Phi$ are inherited from the original truth values at s in $\mathbf{M}$. The resulting finite relational model $\mathbf{M}^f$ is a filtration of $\mathbf{M}$ in the usual sense. By the standard Filtration Lemma, $[s]$ will satisfy φ in $\mathbf{M}^f$.

As usual, the Strong Finite Relational Model Property provides an obvious algorithm for deciding satisfiability on relational models (and thus by Corollary A.4 also on dependence models). Given formula φ generate all the relational models (up to isomorphism) of size $\leq F(\varphi)$; check whether φ is satisfied in any of these models. If so, φ is satisfiable; else, it is not. □

It should be noted that in general the filtrated model is typically a *non-standard* relational model, not a dependence model.

A4. Completeness via relational models

Completeness of LFD with respect to dependence models follows from Corollary A.4 together with the following result:

Lemma A.6. *The calculus* **LFD** *is sound and strongly complete wrt general relational models.*

Proof. Soundness is immediate: the conditions on relational models were chosen to validate the matching axioms. For completeness, take the usual Henkin-style 'canonical model' for LFD, considered as a basic modal logic. This canonical model is a relational model, and the calculus is strongly complete for this model. □

Appendix B: Restricted cut elimination and subformula property

As announced, it is convenient to absorb Weakening into the logical rules (cf. [70] for this technique), while simultaneously restricting Projection and Transitivity to variables that actually occur in the conclusion, and also restricting Cut to dependence atoms between actually occurring variables. This can be done by first modifying the axioms to

(a) $\Gamma, \varphi \vdash \varphi, \Delta$ (b) $\Gamma \vdash D_X x, \Delta$ where $x \in X \subseteq Var(\Gamma \cup \Delta)$,

while introduction rules are made 'cumulative', by repeating principal formulas in the premises.

For instance, the left-introduction rule $(\mathbb{D}_L)$ becomes

$$\frac{\mathbb{D}_X \varphi, \varphi, \Gamma \vdash \Delta}{\mathbb{D}_X \varphi, \Gamma \vdash \Delta}$$

Transitivity needs special treatment: in addition to being made cumulative, it has to be restricted to relevant formulas, becoming the rule of 'Restricted Transitivity':

$$\frac{\Gamma \vdash \Delta, D_X Y, D_X Z \qquad \Gamma \vdash \Delta, D_Y Z, D_X Z}{\Gamma \vdash \Delta, D_X Z} \quad \text{where } Y \subseteq Var(\Gamma \cup \Delta) \cup X \cup Z$$

Likewise, the right-introduction rule $(\mathbb{D}_R)$ needs to be modified to:

$$\frac{\Gamma \vdash \Delta, \varphi, \mathbb{D}_X \varphi}{D_X Y, \Gamma, \Gamma' \vdash \Delta, \Delta', \mathbb{D}_X \varphi} \quad \text{where } Free(\Gamma \cup \Delta) \subseteq Y$$

Finally, we replace Cut by a restricted version (in which we also absorbed Weakening):

$$\text{(DA Cut)} \frac{\Gamma \vdash \Delta, D_X y \qquad D_X y, \Gamma \vdash \Delta}{\Gamma \vdash \Delta} \text{ where } X \cup \{y\} \subseteq Var(\Gamma \cup \Delta)$$

A *restricted-cut proof* is a proof that uses only these modified rules. The following observation shows how LFD allows for a tighter management of variables than FOL:

Lemma B.7. *(Elimination of irrelevant variables)*

- *If $\Gamma \vdash D_X Y, \Delta$ has a restricted-cut proof, and $Z = X \cap (Var(\Gamma) \cup Y \cup Var(\Delta))$, then $\Gamma \vdash D_Z Y, \Delta$ has a restricted-cut proof.*

- *If $\Gamma \vdash \mathbb{D}_X \varphi, \Delta$ has a restricted-cut proof, and $Z = X \cap (Var(\Gamma) \cup Var(\varphi) \cup Var(\Delta))$, then $\Gamma \vdash \mathbb{D}_Z \varphi, \Delta$ has a restricted-cut proof.*

Using this lemma and a cursory inspection of the above modified rules, we obtain:

Lemma B.8. *(Subformula/Subterm Property) Let $\mathcal{P}$ be a restricted-cut proof of the sequent $\Gamma \vdash \Delta$. Each formula θ in $\mathcal{P}$ is either of the form $D_X Y$ with $X \cup Y \subseteq Var(\Gamma \cup \Delta)$, or it is a subformula of some formula in $\Gamma \cup \Delta$. In particular, only variables $x \in Var(\Gamma \cup \Delta)$ occur in $\mathcal{P}$.*

Finally, we can prove our Restricted Cut Elimination theorem:

Every provable sequent has a restricted-cut proof (which thus involves only subformulas of the sequent formulas, or dependence atoms for variables in the sequent)

Proof. To show this, first gradually eliminate Transitivity and Projection in favor of their modified versions, using the above lemma when necessary. Similarly replace all other rules except Cut by their cumulative versions. Finally, eliminate unrestricted cuts in the usual way, by successively removing topmost maximal-rank cuts from a given proof of a sequent $\Gamma \cup \Delta$. Here, since DA Cut is permitted, one need not worry about cut-formulas of the form $D_X Y$, with all variables occurring in the original sequent $\Gamma \cup \Delta$. As a result, the additional axioms and rules for dependency are innocuous: the cut-formula to be removed was never introduced by such rules. The only case that presents any novelty is that of a dual-quantifier cut-formula $\mathbb{D}_X \varphi$ that is principal in both antecedent and succedent: having been freshly introduced on both sides. $\qquad\square$