Notre Dame Journal of Formal Logic 52(1), 55-65, 2011.

A Remark on Negation in Dependence Logic*

Juha Kontinen[†] Jouko Väänänen [‡]

Abstract

We show that for any pair ϕ and ψ of contradictory formulas of dependence logic there is a formula θ of the same logic such that $\phi \equiv \theta$ and $\psi \equiv \neg\theta$. This generalizes a result of J. Burgess [2].

1 Introduction

Dependence logic [11] arises from first-order logic by addition of *dependence atoms*

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

the intuitive meaning of which is that the value of x_n is completely determined by the values of $x_1, \ldots, x_{n-1}$. J. Burgess [2] observed (in the equivalent context of Henkin sentences [6]) that if two sentences ϕ and ψ of dependence logic have no models in common, then there is a sentence θ of dependence logic such that $\phi \equiv \theta$ and $\psi \equiv \neg\theta$. In this paper we generalize this to formulas with free variables.

As Burgess points out, his result indicates that negation is not a semantic operation, that is, knowing the class of models of ϕ is not enough for knowing the class of models of $\neg\phi$. In this sense conjunction and disjunction are different: once we know the classes of models of ϕ and ψ we know exactly which models satisfy $\phi \wedge \psi$ or $\phi \vee \psi$. Likewise, existential and universal quantifiers are semantic operations: once the class of teams satisfying $\phi(x)$ is given, the classes of models of $\exists x\phi(x)$ and $\forall x\phi(x)$ are completely determined.

*The first author was supported by grant 127661 of the Academy of Finland. The second author was partially supported by grant 40734 of the Academy of Finland and by the EUROCORES LogICCC LINT programme. The authors are grateful for the generous support of the Mittag-Leffler Institute, where part of the work reported here was carried out.

†Department of Mathematics and Statistics, University of Helsinki, Finland.

‡Department of Mathematics and Statistics, University of Helsinki, Finland and ILLC, University of Amsterdam, The Netherlands.

Thus we have to conclude that there is something special about negation. This should not come as a surprise, given that the expressive power of sentences of dependence logic is exactly existential second-order logic (Σ_1^1). The negation of dependence logic is not the classical Boolean negation although we have the equivalences:

$$\begin{aligned} \neg(\phi \vee \psi) &\equiv \neg\phi \wedge \neg\psi \\ \neg(\phi \wedge \psi) &\equiv \neg\phi \vee \neg\psi \\ \neg\neg\phi &\equiv \phi \\ \neg\exists x\phi &\equiv \forall x\neg\phi \\ \neg\forall x\phi &\equiv \exists x\neg\phi \end{aligned} \tag{2}$$

which hold for sentences and formulas with free variables alike. A clear indication of the non-classical nature of the negation in dependence logic is the fact that the negations of the usual definitions of concepts such as "infinity", "even cardinality", "equicardinality", "non-wellfoundedness" and "incompleteness (of a linear order)" are all non-determined [10].

In this paper we show that negation has similar qualities when applied to formulas with free variables. We show that if two formulas ϕ and ψ of dependence logic have only the empty team in common, then there is a formula θ of dependence logic such that $\phi \equiv \theta$ and $\psi \equiv \neg\theta$.

The interest in extending the result of Burgess from sentences to formulas stems from the following: We can think of formulas of dependence logic as descriptions of properties of teams. Teams are more or less the same thing as databases. Thus formulas describe properties of databases. The atomic formula (1) describes the property of a database that a certain field (x_n) is completely determined by a combination of other fields ($x_1, \ldots, x_{n-1}$). In computer science this is called a *functional dependency* [1]. The composite formulas of dependence logic simply describe more complex dependencies than the mere functional dependencies. Our result then shows that if two database properties, definable in dependence logic (i.e., downwards monotone NP by Theorems 2.10 and 2.11), are such that only the empty database has both, then we can capture (either) one of the properties with a formula of dependence logic in such a way that its negation hits exactly the other property.

This non-semantic quality of negation disappears if we define the semantic value of a formula as the set of pairs of teams (X, Y) such that X satisfies the formula and Y its negation. Then the semantic value of the negation can of course be easily computed by simply reversing the pairs (X, Y) into (Y, X).

2 Preliminaries

In this section we define Dependence Logic ($\mathcal{D}$) and recall some basic results on dependence logic.

Definition 2.1 ([11])**.** *The syntax of $\mathcal{D}$ extends the syntax of* FO*, defined in terms of $\vee$, $\wedge$, $\neg$, $\exists$ and $\forall$, by new atomic (dependence) formulas of the form*

$$=\!(t_1, \ldots, t_n), \tag{3}$$

where $t_1, \ldots, t_n$ are terms. If L is a vocabulary, we use $\mathcal{D}[L]$ to denote the set of formulas of $\mathcal{D}$ based on L.

The intuitive meaning of the dependence formula (3) is that the value of the term t_n is determined by the values of the terms $t_1, \ldots, t_{n-1}$. As singular cases we have

$$=\!(),$$

which we take to be universally true, and

$$=\!(t),$$

which declares that the value of the term t depends on nothing, i.e., is constant. We let $\top$ be the formula $=\!()$ and $\bot$ be $\neg =\!()$.

The set $\mathrm{Fr}(\phi)$ of free variables of a formula $\phi \in \mathcal{D}$ is defined as for first-order logic, except that we have the new case

$$\mathrm{Fr}(=\!(t_1, \ldots, t_n)) = \mathrm{Var}(t_1) \cup \cdots \cup \mathrm{Var}(t_n),$$

where $\mathrm{Var}(t_i)$ is the set of variables occurring in term t_i. If $\mathrm{Fr}(\phi) = \emptyset$, we call ϕ a sentence.

In order to define the semantics of $\mathcal{D}$, we first need to define the concept of a *team*. Let $\mathfrak{A}$ be a model with domain A. In this article, all models $\mathfrak{A}$ are assumed to have a domain with at least two elements. *Assignments* of $\mathfrak{A}$ are finite mappings from variables into A. The value of a term t in an assignment s is denoted by $t^{\mathfrak{A}}\langle s\rangle$. If s is an assignment, x a variable, and $a \in A$, then $s(a/x)$ denotes the assignment (with domain $\mathrm{dom}(s) \cup \{x\}$) which agrees with s everywhere except that it maps x to a.

Let A be a set and $\{x_1, \ldots, x_k\}$ a finite (possibly empty) set of variables. A *team* X of A with domain $\mathrm{dom}(X) = \{x_1, \ldots, x_k\}$ is any set of assignments from the variables $\{x_1, \ldots, x_k\}$ into the set A. We denote by $rel(X)$ the k-ary relation of A corresponding to X

$$rel(X) = \{(s(x_1), \ldots, s(x_k)) : s \in X\}.$$

If X is a team of A, and $F\colon X \to A$, we use $X(F/x_n)$ to denote the team $\{s(F(s)/x_n) : s \in X\}$ and $X(A/x_n)$ the team $\{s(a/x_n) : s \in X \text{ and } a \in A\}$.

We are now ready to define the semantics of dependence logic. In Definition 2.2 below, we only consider formulas in negation normal form, i.e., negation is allowed to appear only in front of atomic formulas. However, in this article we allow negation of dependence logic to appear freely in formulas. For a formula $\psi = \neg\phi$, where ϕ is not atomic, $\mathfrak{A} \models_X \psi$ is taken as a shorthand for $\mathfrak{A} \models_X \psi^*$, where ψ^* is acquired by transforming ψ into negation normal form using the equivalences of (2).

Definition 2.2 ([11])**.** *Let $\mathfrak{A}$ be a model and X a team of A. The satisfaction relation $\mathfrak{A} \models_X \varphi$ is defined as follows:*

1. $\mathfrak{A} \models_X t_1 = t_2$ *iff for all* $s \in X$ *we have* $t_1^{\mathfrak{A}}\langle s\rangle = t_2^{\mathfrak{A}}\langle s\rangle$.
2. $\mathfrak{A} \models_X \neg t_1 = t_2$ *iff for all* $s \in X$ *we have* $t_1^{\mathfrak{A}}\langle s\rangle \neq t_2^{\mathfrak{A}}\langle s\rangle$.
3. $\mathfrak{A} \models_X =\!(t_1, \ldots, t_n)$ *iff for all* $s, s' \in X$ *such that* $t_1^{\mathfrak{A}}\langle s\rangle = t_1^{\mathfrak{A}}\langle s'\rangle, \ldots, t_{n-1}^{\mathfrak{A}}\langle s\rangle = t_{n-1}^{\mathfrak{A}}\langle s'\rangle$, *we have* $t_n^{\mathfrak{A}}\langle s\rangle = t_n^{\mathfrak{A}}\langle s'\rangle$.
4. $\mathfrak{A} \models_X \neg =\!(t_1, \ldots, t_n)$ *iff* $X = \emptyset$.
5. $\mathfrak{A} \models_X R(t_1, \ldots, t_n)$ *iff for all* $s \in X$ *we have* $(t_1^{\mathfrak{A}}\langle s\rangle, \ldots, t_n^{\mathfrak{A}}\langle s\rangle) \in R^{\mathfrak{A}}$.
6. $\mathfrak{A} \models_X \neg R(t_1, \ldots, t_n)$ *iff for all* $s \in X$ *we have* $(t_1^{\mathfrak{A}}\langle s\rangle, \ldots, t_n^{\mathfrak{A}}\langle s\rangle) \notin R^{\mathfrak{A}}$.
7. $\mathfrak{A} \models_X \psi \wedge \phi$ *iff* $\mathfrak{A} \models_X \psi$ *and* $\mathfrak{A} \models_X \phi$.
8. $\mathfrak{A} \models_X \psi \vee \phi$ *iff* $X = Y \cup Z$ *such that* $\mathfrak{A} \models_Y \psi$ *and* $\mathfrak{A} \models_Z \phi$.
9. $\mathfrak{A} \models_X \exists x_n \psi$ *iff* $\mathfrak{A} \models_{X(F/x_n)} \models \psi$ *for some* $F\colon X \to A$.
10. $\mathfrak{A} \models_X \forall x_n \psi$ *iff* $\mathfrak{A} \models_{X(A/x_n)} \psi$.

Above, we assume that the domain of X contains the variables free in ϕ. Finally, a sentence ϕ is true in a model $\mathfrak{A}$ ($\mathfrak{A} \models \phi$) if $\mathfrak{A} \models_{\{\emptyset\}} \phi$.

From Definition 2.2 it follows that many familiar propositional equivalences of connectives do not hold in dependence logic. For example, the idempotence of disjunction fails, which can be used to show that the distributivity laws of disjunction and conjunction do not hold in dependence logic either. We refer to Section 3.3 of [11] for a detailed exposition on propositional equivalences of connectives in dependence logic.

Another important distinction between first-order logic and dependence logic is that $\mathfrak{A} \not\models_X \phi$ does not always imply that $\mathfrak{A} \models_X \neg\phi$.

Example 2.3. *Let $\mathfrak{A}$ be a model with $A = \{0, 1, 2\}$. Consider the following team X of A:*

	x_0	x_1	x_2
s_0	1	2	2
s_1	2	1	2
s_2	0	1	2

(4)

By Definition 2.2 part 1, we have that $\mathfrak{A} \not\models_X x_0 = x_2$, since $s_0(x_0) \neq s_0(x_2)$, and $\mathfrak{A} \not\models_X \neg x_0 = x_2$, since $s_1(x_0) = s_1(x_2)$. On the other hand, it holds that $\mathfrak{A} \models_X \exists x_0(x_0 = x_2)$: let $F \colon X \to A$ be the mapping $F(s_0) = 2$, $F(s_1) = 2$, $F(s_2) = 2$. Then $Y = X(F/x_0)$ is the team

	x_0	x_1	x_2
s_0	2	2	2
s_1	2	1	2
s_2	2	1	2

(5)

By Definition 2.2 part 1, we have that $\mathfrak{A} \models_Y x_0 = x_2$, hence, by Definition 2.2 part 9, $\mathfrak{A} \models_X \exists x_0(x_0 = x_2)$.

Example 2.4. *Let $\mathfrak{A}$ be a model. Consider the following sentence ϕ*

$$\phi := \forall x =\!(x).$$

Now ϕ is true in $\mathfrak{A}$ if $\mathfrak{A} \models_{\{\emptyset\}} \phi$, and $\mathfrak{A} \models_{\{\emptyset\}} \phi$ if and only if $\mathfrak{A} \models_Y =\!(x)$, where

$$Y = \{\emptyset\}(A/x) = \{s(a/x) : s \in \{\emptyset\} \text{ and } a \in A\}.$$

Therefore, for every $a \in A$, Y contains an assignment s with domain $\{x\}$ such that $s(x) = a$. Since $|A| \geq 2$, $\mathfrak{A} \not\models_Y =\!(x)$, whence $\mathfrak{A} \not\models_{\{\emptyset\}} \phi$. On the other hand, note that $\mathfrak{A} \models_{\{\emptyset\}} \neg\phi$ if $\mathfrak{A} \models_{\{\emptyset\}} \exists x(\neg =\!(x))$ if and only if $\mathfrak{A} \models_{\{\emptyset\}(F/x)} \neg =\!(x)$, for some $F \colon \{\emptyset\} \to A$. Since for all F, the team $\{\emptyset\}(F/x)$ is non-empty, Definition 2.2 part 4 implies that, for all $F \colon \{\emptyset\} \to A$, $\mathfrak{A} \not\models_{\{\emptyset\}(F/x)} \neg =\!(x)$, and hence $\mathfrak{A} \not\models_{\{\emptyset\}} \neg\phi$.

A sentence ϕ is called *determined* in a model $\mathfrak{A}$ if either $\mathfrak{A} \models \phi$ or $\mathfrak{A} \models \neg\phi$. Otherwise ϕ is called *non-determined* in $\mathfrak{A}$. Next we define the notions of logical consequence and equivalence for formulas of dependence logic.

Definition 2.5. *Let ϕ and ψ be formulas of dependence logic. The formula ψ is a* logical consequence *of ϕ,*

$$\phi \Rightarrow \psi,$$

if for all models $\mathfrak{A}$ *and teams* X*, with* $\mathrm{Fr}(\phi) \cup \mathrm{Fr}(\psi) \subseteq \mathrm{dom}(X)$*, and* $\mathfrak{A} \models_X \phi$ *we have* $\mathfrak{A} \models_X \psi$*. The formulas* ϕ *and* ψ *are* logically equivalent*,*

$$\phi \equiv \psi,$$

if $\phi \Rightarrow \psi$ *and* $\psi \Rightarrow \phi$*.*

It is worth noting that $\phi \equiv \psi$ does not in general entail that $\neg\phi \equiv \neg\psi$. For the purposes of this paper, formulas ϕ and ψ are said to be *contradictory* if, for all $\mathfrak{A}$ and X, $\mathfrak{A} \models_X \phi$ and $\mathfrak{A} \models_X \psi$ implies $X = \emptyset$. We have to allow ϕ and ψ to agree on $X = \emptyset$ because, over any model $\mathfrak{A}$, the empty team satisfies all formulas of dependence logic.

Proposition 2.6. *For all models* $\mathfrak{A}$ *and formulas* ϕ *of dependence logic, it holds that* $\mathfrak{A} \models_\emptyset \phi$*.*

Proof. See Lemma 3.9 in [11]. □

Let X be a team with domain $\{x_1, \ldots, x_k\}$ and $V \subseteq \{x_1, \ldots, x_k\}$. Denote by $X \upharpoonright V$ the team $\{s \upharpoonright V : s \in X\}$ with domain V. The following lemma shows that the truth of a formula depends only on the interpretations of the variables occurring free in the formula.

Lemma 2.7. *Suppose* $V \supseteq \mathrm{Fr}(\phi)$*. Then* $\mathfrak{A} \models_X \phi$ *if and only if* $\mathfrak{A} \models_{X \upharpoonright V} \phi$*.*

Proof. See Lemma 3.27 in [11]. □

The following fact (Fact 11.1 in [7], see Proposition 3.10 in [11]) is also a very basic property of all formulas of dependence logic:

Proposition 2.8 (Downward closure)**.** *Let* ϕ *be a formula of dependence logic,* $\mathfrak{A}$ *a model, and* $Y \subseteq X$ *teams. Then* $\mathfrak{A} \models_X \phi$ *implies* $\mathfrak{A} \models_Y \phi$*.*

On the other hand, the expressive power of sentences of $\mathcal{D}$ coincides with that of existential second-order sentences (Σ^1_1):

Theorem 2.9. *For every sentence* ϕ *of* $\mathcal{D}$ *there is a sentence* Φ *of* Σ^1_1 *such that*

$$\text{For all models } \mathfrak{A}\text{: } \mathfrak{A} \models_{\{\emptyset\}} \phi \iff \mathfrak{A} \models \Phi. \tag{6}$$

Conversely, for every sentence Φ *of* Σ^1_1 *there is a sentence* ϕ *of* $\mathcal{D}$ *such that (6) holds.*

Proof. Using the method of [12, 5] (See Theorems 6.2 and 6.15 in [11]). □

However, Theorem 2.9 does not – a priori – tell us anything about formulas with free variables. An upperbound for the complexity of formulas of $\mathcal{D}$ is provided by the following result showing that formulas of dependence logic can be compositionally translated into sentences of Σ_1^1 [11].

Theorem 2.10. *Let L be a vocabulary and ϕ a $\mathcal{D}[L]$-formula with free variables $v_1, \ldots, v_r$. Then there is a $L \cup \{R\}$-sentence ψ of Σ_1^1, in which R appears only negatively, such that for all model $\mathfrak{A}$ and teams X with domain $\{v_1, \ldots, v_r\}$:*

$$\mathfrak{A} \models_X \phi \iff (\mathfrak{A}, rel(X)) \models \psi.$$

In [9] it was shown that also the converse holds.

Theorem 2.11. *Let L be a vocabulary, R r-ary, and $R \notin L$. Then for every $L \cup \{R\}$-sentence ψ of Σ_1^1, in which R appears only negatively, there is a L-formula ϕ of dependence logic with free variables $v_1, \ldots, v_r$ such that, for all $\mathfrak{A}$ and X with domain $\{v_1, \ldots, v_r\}$:*

$$\mathfrak{A} \models_X \phi \iff (\mathfrak{A}, rel(X)) \models \psi \vee \forall \overline{x} \neg R(\overline{x}). \tag{7}$$

Proof. See Theorem 4.10 in [9]. □

Theorem 2.11 shows that formulas of dependence logic correspond in a precise way to the negative fragment of Σ_1^1 and are therefore very expressive. On the other hand, if we restrict attention to formulas that do not contain dependence atomic formulas as subformulas, we lose much of the expressive power.

Definition 2.12. *A formula ϕ of $\mathcal{D}$ is called a first-order formula if it does not contain dependence atomic formulas as subformulas.*

Theorem 2.13. *Let ϕ be a first-order formula of dependence logic. Then for all $\mathfrak{A}$ and X:*

1. $\mathfrak{A} \models_{\{s\}} \phi \Leftrightarrow \mathfrak{A} \models_s \phi$.
2. $\mathfrak{A} \models_X \phi \Leftrightarrow$ *for all* $s \in X$: $\mathfrak{A} \models_s \phi$.

In this article our goal is to generalize the following result of J. Burgess [2] to cover also open formulas.

Theorem 2.14. *Suppose that ϕ and ψ are contradictory sentences of dependence logic. Then there is a sentence $\theta \in \mathcal{D}$ such that $\phi \equiv \theta$ and $\psi \equiv \neg\theta$.*

The proof of Theorem 2.14 is based on the following direct consequence of Theorem 2.9 and the Craig Interpolation Theorem [4]:

Theorem 2.15. *Let ϕ and ψ be contradictory sentences of dependence logic. Let the vocabulary of ϕ be L and the vocabulary of ψ be L'. Then there is a first-order sentence θ in the vocabulary $L \cap L'$ such that $\phi \Rightarrow \theta$ and $\psi \Rightarrow \neg\theta$.*

Proof. See Theorem 6.7 in [11]. □

3 The main result

The negation of dependence logic is not the classical negation and hence knowing the models of ϕ does not completely determine the models of $\neg\phi$. Recall that, by Theorem 2.11, the formulas of dependence logic correspond to the downwards monotone classes of Σ^1_1. However, Theorem 2.11 does not, in its formulation, say anything about negations of open formulas. In fact, for every ϕ, constructed in the proof of Theorem 2.11, it is immediate that $\neg\phi$ is logically equivalent to a first-order formula of $\mathcal{D}$.

The following result shows that the analogue of Theorem 2.14 does indeed extend to all formulas of dependence logic.

Theorem 3.1. *Let L be a vocabulary and let ϕ and ψ be contradictory L-formulas of dependence logic with free variables $v_1, \ldots, v_k$. Then there is a L-formula θ of dependence logic with the same free variables such that $\phi \equiv \theta$ and $\psi \equiv \neg\theta$.*

Proof. Let us first assume that there is a L-formula η with free variables $v_1, \ldots, v_k$ such that

$$\phi \Rightarrow \eta, \tag{8}$$

and

$$\psi \Rightarrow \neg\eta. \tag{9}$$

Define $\hat{\phi} := \phi \vee \forall x =\!(x)$ and $\hat{\psi} := \psi \vee \forall x =\!(x)$. In Example 2.4, it was shown that for every $\mathfrak{A}$ (recall that $|A|$ is at least 2) and $X \neq \emptyset$, $\mathfrak{A} \not\models_X \forall x =\!(x)$ and also $\mathfrak{A} \not\models_X \neg\forall x =\!(x)$. This implies that, in every model $\mathfrak{A}$, $\mathfrak{A} \not\models_X \neg\hat{\phi}$ and $\mathfrak{A} \not\models_X \neg\hat{\psi}$ for all non-empty teams X. We now claim that the formula θ below is as wanted

$$\theta := \hat{\phi} \wedge (\neg\hat{\psi} \vee \eta).$$

Note that $\neg\theta \equiv \neg\hat{\phi} \vee (\hat{\psi} \wedge \neg\eta)$. We first show that $\phi \equiv \theta$. Let $\mathfrak{A}$ be a model and $X \neq \emptyset$. Note that it suffices to consider teams $X \neq \emptyset$ since, by Proposition 2.6, all formulas of dependence logic are satisfied by the empty team. Let us assume $\mathfrak{A} \models_X \phi$. Then, by (8), $\mathfrak{A} \models_X \eta$, and hence $\mathfrak{A} \models_X \theta$. Assume then that $\mathfrak{A} \models_X \theta$. Then $\mathfrak{A} \models_X \hat{\phi}$. Now since $\mathfrak{A} \not\models_Y \forall x =\!(x)$ for all $Y \neq \emptyset$, we must have $\mathfrak{A} \models_X \phi$.

Let us then show that $\neg\theta \equiv \psi$. Assume $\mathfrak{A} \models_X \psi$ and $X \neq \emptyset$. Then $\mathfrak{A} \models_X \hat{\psi}$. Now, by (9), it holds that $\mathfrak{A} \models_X \hat{\psi} \wedge \neg\eta$, hence we have $\mathfrak{A} \models_X \neg\theta$. For the converse, assume that $\mathfrak{A} \models_X \neg\theta$. Recall that $\neg\theta \equiv \neg\hat{\phi} \vee (\hat{\psi} \wedge \neg\eta)$. Now $\neg\hat{\phi} \equiv \neg\phi \wedge \neg\forall x =\!(x)$ and since $\mathfrak{A} \not\models_Y \neg\forall x =\!(x)$ for all $Y \neq \emptyset$, we have $\mathfrak{A} \not\models_Y \neg\hat{\phi}$ for all $Y \neq \emptyset$. The only possibility is then that $\mathfrak{A} \models_X \hat{\psi} \wedge \neg\eta$ implying $\mathfrak{A} \models_X \hat{\psi}$. Since $\mathfrak{A} \not\models_Y \neg\forall x =\!(x)$ for all $Y \neq \emptyset$, we have $\mathfrak{A} \models_X \psi$ as wanted.

It now suffices to show that for each pair ϕ and ψ of contradictory formulas, we can construct a formula η satisfying (8) and (9).

Without loss of generality, we may assume that both ϕ and ψ are satisfied in some model $\mathfrak{A}$ by some $X \neq \emptyset$. Note that if, e.g., $\phi \equiv \bot$, we can choose $\eta = \phi$ trivially. By Theorem 2.15 there is a sentence ϕ' (analogously ψ') of $\Sigma^1_1[L \cup R]$, where R k-ary, such that for all models $\mathfrak{A}$ and teams X with domain $\{v_1, \ldots, v_k\}$ it holds that

$$\mathfrak{A} \models_X \phi \Leftrightarrow (\mathfrak{A}, rel(X)) \models \phi'. \tag{10}$$

$$\mathfrak{A} \models_X \psi \Leftrightarrow (\mathfrak{A}, rel(X)) \models \psi'. \tag{11}$$

By Proposition 2.6, the sentences ϕ' and ψ' are not contradictory: if $X = \emptyset$, then $(\mathfrak{A}, rel(X)) \models \phi'$ and $(\mathfrak{A}, rel(X)) \models \psi'$. However, we can easily exclude these models by considering the sentences $\phi^* = \phi' \wedge \chi$ and $\psi^* = \psi' \wedge \chi$, instead where $\chi \equiv \exists\overline{x}R(\overline{x})$. By our assumption, both ϕ^* and ψ^* are still satisfiable.

Now, applying the Craig Interpolation Theorem [4], we get a first-order sentence η such that $\phi^* \Rightarrow \eta$ and $\psi^* \Rightarrow \neg\eta$ (cf. Theorem 2.15). Let now $\eta^*(\overline{v})$ be the first-order formula $\eta(P(\overline{t}) \setminus \overline{v} = \overline{t})$, i.e., the formula in which all occurrences of subformulas of the form $P(t_1, \ldots, t_k)$ are replaced by formulas

$$\bigwedge_{1 \leq i \leq k} v_i = t_i.$$

Using induction on the construction of η^* it holds that for all $\mathfrak{A}$ and $\overline{a} \in A^k$

$$(\mathfrak{A}, \{\overline{a}\}) \models \eta \Leftrightarrow \mathfrak{A} \models_s \eta^*, \tag{12}$$

where $s(v_i) = a_i$, for $1 \leq i \leq k$. We can now interpret η^* as a first-order formula of dependence logic. We claim that, for all models $\mathfrak{A}$ and teams X, the formula η^* satisfies the clauses (8) and (9). Let us first show clause (8). Let $\mathfrak{A}$ be a model and $X \neq \emptyset$. Suppose that $\mathfrak{A} \models_X \phi$. By the downward closure, we get that, for all $s \in X$, $\mathfrak{A} \models_{\{s\}} \phi$. Therefore, by (10), for all $s \in X$, $(\mathfrak{A}, \{(s(v_1), \ldots, s(v_k))\}) \models \phi^*$. It follows that, for all $s \in X$,

$(\mathfrak{A}, \{(s(v_1), \ldots, s(v_k))\}) \models \eta$, and, by (12), that, for all $s \in X$, $\mathfrak{A} \models_s \eta^*$. Now Theorem 2.13 part 2 implies that $\mathfrak{A} \models_X \eta^*$ as wanted.

Let us then show that the clause (9) holds. Let $\mathfrak{A}$ be a model and $X \neq \emptyset$. Suppose that $\mathfrak{A} \models_X \psi$. By the downward closure, we get that, for all $s \in X$, $\mathfrak{A} \models_{\{s\}} \psi$. Therefore, by (11), for all $s \in X$, $(\mathfrak{A}, \{(s(v_1), \ldots, s(v_k))\}) \models \psi^*$ and thus, for all $s \in X$, $(\mathfrak{A}, \{(s(v_1), \ldots, s(v_k))\}) \models \neg\eta$. Again, by (12), we get that, for all $s \in X$, $\mathfrak{A} \models_s \neg\eta^*$. Now Theorem 2.13 part 2 implies that $\mathfrak{A} \models_X \neg\eta^*$ as wanted. □

As discussed in the Introduction, Theorem 3.1 shows that the negation of dependence logic is not a semantic operation, i.e., knowing the models of ϕ does not tell us practically anything about $\neg\phi$. It is straightforward to show that the connectives $\rightarrow$ and $\leftrightarrow$, defined in terms of $\neg$, $\wedge$, and $\vee$ in the usual way, are also non-semantical operations.

Corollary 3.2. *Let ξ_1 and ξ_2 be formulas of dependence logic. There are formulas ϕ_i and ψ_i, for $1 \leq i \leq 2$, such that $\phi_1 \equiv \psi_1$ and $\phi_2 \equiv \psi_2$, but*

$$(\phi_1 \rightarrow \phi_2) \equiv \xi_1$$
$$(\psi_1 \rightarrow \psi_2) \equiv \xi_2$$
$$(\phi_1 \leftrightarrow \phi_2) \equiv \xi_1$$
$$(\psi_1 \leftrightarrow \psi_2) \equiv \xi_2$$

Proof. Let $\phi_2 = \psi_2 = \bot$. By Theorem 3.1, we can find formulas ϕ_1 and ψ_1 such that

$$\phi_1 \equiv \psi_1 \equiv \bot,$$

$\neg\phi_1 \equiv \xi_1$, and $\neg\psi_1 \equiv \xi_2$. Now it holds that

$$\phi_1 \rightarrow \phi_2 \equiv \neg\phi_1 \vee \bot \equiv \neg\phi_1 \equiv \xi_1.$$

Analogously, we get that $\psi_1 \rightarrow \psi_2 \equiv \xi_2$. Finally, note that

$$\begin{aligned} \phi_1 \leftrightarrow \phi_2 &\equiv (\neg\phi_1 \vee \bot) \wedge (\neg\phi_2 \vee \bot) \\ &\equiv \xi_1 \wedge (\top \vee \bot) \equiv \xi_1. \end{aligned}$$

Analogously, we get that $\psi_1 \leftrightarrow \psi_2 \equiv \xi_2$. □

Corollary 3.2 shows that if we a given an implication (respectively an equivalence) of which we do not know either the antecedent or the consequent, but only the classes of models that satisfy the antecedent and the consequent, then we cannot say anything about the class of models satisfying the implication.

4 The case of IF logic

In this section we formulate Theorem 3.1 for Independence Friendly Logic (IF logic). We first briefly recall the syntax and semantics of IF logic.

The syntax of IF logic (as defined in [3]) extends the syntax of FO by slashed quantifiers $(\exists x/W)$ and $(\forall x/W)$, where W is a finite set of variables. The intuitive meaning, e.g., of a formula $(\exists x/\{y\})\phi$ is that "there exists x, independently of y, such that ϕ". Compositional semantics, similar to Definition 2.2, was defined for IF logic in [7]. In the context of IF logic one usually talks about trumps instead of teams, i.e., a trump for a formula $\phi(x_1,\ldots,x_n)$, with free variables $x_1,\ldots,x_n$, corresponds to a team with domain $\{x_1,\ldots,x_n\}$. The set $\mathrm{Fr}(\phi)$ of free variables of a IF-formula ϕ is defined otherwise as for first-order logic, except that we have the new cases: $\mathrm{Fr}((\exists x/W)\psi)) = W \cup (\mathrm{Fr}(\psi) \setminus \{x\})$ and $\mathrm{Fr}((\forall x/W)\psi)) = W \cup (\mathrm{Fr}(\psi) \setminus \{x\})$. We refer to the Appendix of [3] for the truth definition of IF logic and only discuss its similarities and differences to Definition 2.2.

In IF logic, atomic formulas and connectives $\wedge$, $\vee$, and $\neg$ are treated just like in Definition 2.2. With respect to trumps with a fixed domain

$$\{x_1,\ldots,x_n\},$$

the meaning of a formula of the form $(\exists x/W)\phi$, where $W \subseteq \{x_1,\ldots,x_n\}$, is that "there is an x, depending only on variables in the set $\{x_1,\ldots,x_n\} \setminus W$, such that ϕ". This can be expressed in dependence logic as

$$\exists x(=\!(x_{j_1},\ldots,x_{j_r},x) \wedge \phi), \tag{13}$$

where $\{x_{j_1},\ldots,x_{j_r}\} = \{x_1,\ldots,x_n\} \setminus W$. Note that if we consider trumps over variables $\{x_1,\ldots,x_{n+m}\}$, the variables $x_{n+1},\ldots,x_{n+m}$ need to be added to formula (13). This simple observation actually marks a difference between IF logic and $\mathcal{D}$, since, unlike with $\mathcal{D}$, the truth of an IF-formula may depend on the interpretations of variables that do not occur in the formula. For example, the truth of the formula ϕ

$$\phi = \exists x/\{y\}(x = y) \tag{14}$$

in a trump X with domain $\{x, y, z\}$ depends on the values of z in X, although z does not occur in ϕ. This observation also implies that it is not possible to define a compositional meaning-preserving translation of formulas either from IF logic into $\mathcal{D}$, or from $\mathcal{D}$ into IF logic.

We shall next show that a version of Theorem 3.1 can be proved for IF-logic. As a corollary, we get a (non-compositional) translation of formulas between IF logic and dependence logic.

Theorem 4.1. *Let L be a vocabulary and let ϕ and ψ be contradictory L-formulas of* IF *logic with free variables $v_1, \ldots, v_k$. Then there is a L-formula θ of* IF *logic with the same free variables such that for all models $\mathfrak{A}$ and trumps X with domain $\{v_1, \ldots, v_k\}$: $\mathfrak{A} \models_X \phi \Leftrightarrow \mathfrak{A} \models_X \theta$ and $\mathfrak{A} \models_X \psi \Leftrightarrow \mathfrak{A} \models_X \neg\theta$.*

Proof. All the properties of dependence logic used in the proof of Theorem 3.1 also hold for IF logic. In particular, the analogue of Theorem 2.10 for IF-formulas can be found in [8]. The formula $\forall x =\!(x)$ can be replaced, e.g., by the formula

$$\forall x \exists y/\{x\}(x = y),$$

which (and its negation) is non-determined in all structures of cardinality greater than 1. □

Unlike with dependence logic (Proposition 2.7), it is not clear that Theorem 4.1 holds without the restriction to trumps with domain $\{v_1, \ldots, v_k\}$. It is an open question whether the version of Theorem 4.1 holds in which "for all X with domain $\{v_1, \ldots, v_k\}$" is replaced by "for all X with $\{v_1, \ldots, v_k\} \subseteq \mathrm{dom}(X)$".

Theorem 4.1 can be used to show the following (non-compositional) translation of formulas between IF logic and dependence logic.

Corollary 4.2. *Let L be a vocabulary. For every L-formula $\phi \in \mathcal{D}$ ($\phi \in$ IF) with free variables $v_1, \ldots, v_k$ there is a L-formula ϕ^* of* IF *logic (respectively $\phi^* \in \mathcal{D}$) with the same free variables such that for all models $\mathfrak{A}$ and X with domain $\{v_1, \ldots, v_k\}$:*

$$\begin{aligned} \mathfrak{A} \models_X \phi &\Leftrightarrow \mathfrak{A} \models_X \phi^*, \\ \mathfrak{A} \models_X \neg\phi &\Leftrightarrow \mathfrak{A} \models_X \neg\phi^*. \end{aligned}$$

Proof. We show how to translate $\phi \in \mathcal{D}[L]$ into a IF$[L]$-formula ϕ^*. By Theorem 2.10, there are sentences ψ_+ and ψ_- of $\Sigma^1_1[L \cup \{R\}]$, in which R appears only negatively, that are equivalent (in the sense of Theorem 2.10) to ϕ and $\neg\phi$, respectively. By Theorem 5.2 in [9] (i.e., the analogue of Theorem 2.11 for IF logic), there are IF$[L]$-formulas Ψ_+ and Ψ_- equivalent to ψ_+ and ψ_-, with $v_1, \ldots, v_k$ free. By Theorem 4.1 there is a IF$[L]$-formula ϕ^* such that for all models $\mathfrak{A}$ and trumps X with domain $\{v_1, \ldots, v_k\}$: $\mathfrak{A} \models_X \phi^* \Leftrightarrow \mathfrak{A} \models_X \Psi_+$ and $\mathfrak{A} \models_X \neg\phi^* \Leftrightarrow \mathfrak{A} \models_X \Psi_-$. By the construction, ϕ^* is a correct translation for ϕ. For the converse, we first apply the translation of IF$[L]$-formulas into $\Sigma^1_1[L \cup \{R\}]$-sentences (see [8]), and then Theorems 2.11 and 3.1. □

References

[1] W. W. Armstrong. Dependency structures of data base relationships. In *IFIP Congress*, pages 580–583, 1974.

[2] J. P. Burgess. A remark on Henkin sentences and their contraries. *Notre Dame J. Formal Logic*, 44(3):185–188 (electronic) (2004), 2003.

[3] P. Cameron and W. Hodges. Some combinatorics of imperfect information. *J. Symbolic Logic*, 66(2):673–684, 2001.

[4] W. Craig. Linear reasoning. a new form of the herbrand-gentzen theorem. *J. Symb. Log.*, 22(3):250–268, 1957.

[5] H. B. Enderton. Finite partially-ordered quantifiers. *Z. Math. Logik Grundlagen Math.*, 16:393–397, 1970.

[6] L. Henkin. Some remarks on infinitely long formulas. In *Infinitistic Methods (Proc. Sympos. Foundations of Math., Warsaw, 1959)*, pages 167–183. Pergamon, Oxford, 1961.

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

[8] W. Hodges. Some strange quantifiers. In *Structures in logic and computer science*, volume 1261 of *Lecture Notes in Comput. Sci.*, pages 51–65. Springer, Berlin, 1997.

[9] J. Kontinen and J. Väänänen. On definability in dependence logic. *Journal of Logic, Language and Information*, 18(3):317–332, 2009.

[10] J. Väänänen. A Remark on Nondeterminacy in IF Logic. *Acta Philosophica Fennica*, 78:71–77, 2006.

[11] J. Väänänen. *Dependence logic: A New Approach to Independence Friendly Logic*, volume 70 of *London Mathematical Society Student Texts*. Cambridge University Press, Cambridge, 2007.

[12] W. J. Walkoe, Jr. Finite partially-ordered quantification. *J. Symbolic Logic*, 35:535–555, 1970.