Chapter 4
On Dependence Logic

Pietro Galliani and Jouko Väänänen

Abstract Dependence logic extends the language of first order logic by means of *dependence atoms* and aims to establish a basic theory of dependence and independence underlying such seemingly unrelated subjects as causality, random variables, bound variables in logic, database theory, the theory of social choice, and even quantum physics. In this work we summarize the setting of dependence logic and recall the main results of this rapidly developing area of research.

4.1 Introduction

The goal of dependence logic is to establish a basic theory of dependence and independence underlying such seemingly unrelated subjects as causality, random variables, bound variables in logic, database theory, the theory of social choice, and even quantum physics. There is an avalanche of new results in this field demonstrating remarkable convergence. The concepts of (in)dependence in the different fields of

J. Väänänen's research partially supported by grant 40734 of the Academy of Finland and the EUROCORES LogICCC LINT programme.
P. Galliani's research partially supported by the EUROCORES LogICCC LINT programme, by the Väisälä Foundation and by by Grant 264917 of the Academy of Finland.

P. Galliani (✉)
Department of Mathematics and Statistics, University of Helsinki, Helsinki, Finland
e-mail: pgallian@gmail.com

J. Väänänen
Department of Mathematics and Statistics, University of Helsinki, Helsinki, Finland
e-mail: jouko.vaananen@helsinki.fi

J. Väänänen
Institute for Logic, Language and Computation, University of Amsterdam, Amsterdam,
The Netherlands

A. Baltag and S. Smets (eds.), *Johan van Benthem on Logic and Information Dynamics*, Outstanding Contributions to Logic 5,
DOI: 10.1007/978-3-319-06025-5_4, © Springer International Publishing Switzerland 2014

humanities and sciences have surprisingly much in common and a common logic is starting to emerge.

Dependence logic [29] arose from the compositional semantics of Wilfrid Hodges [19] for the independence friendly logic [18, 25]. In dependence logic the basic semantic concept is *not* that of an assignment s satisfying a formula ϕ in a model $\mathfrak{M}$,

$$\mathfrak{M} \models_s \phi,$$

as in first order logic, but rather the concept of a *set* S of assignments satisfying ϕ in $\mathfrak{M}$,

$$\mathfrak{M} \models_S \phi.$$

Defining satisfaction relative to a *set* of assignments opens up the possibility to express dependence phenomena, roughly as passing in propositional logic from one valuation to a Kripke model leads to the possibility to express modality. The focus in dependence logic is not on truth values but on variable values. We are interested in dependencies between individuals rather than between propositions.

In [3] Johan van Benthem writes:

> "*Sets of assignments S encode several kinds of* '*dependence*'
> *between variables. There may not be one single intuition.*
> '*Dependence*' *maymean functional dependence*
> (*if two assignments agree in S onx*, *they also agree ony*),
> *but also other kinds of* '*correlation*' *among value ranges.* (4.1)
> ...
> *Different dependence relations may have different mathematical*
> *properties and suggest different logical formalisms.*"

This is actually how things have turned out. For a start, using the concept of functional dependence it is possible, as Wilfrid Hodges [19] demonstrated, to define compositionally[1] the semantics of independence friendly logic, the extension of first order logic by the quantifier

$$\exists x/y\phi \qquad \text{i.e. “there is an } x\text{, independently of } y\text{, such that } \phi\text{”},$$

as follows: Suppose S is a team of assignments, a "plural state", in a model $\mathfrak{M}$. Then

$$\mathfrak{M} \models_S \exists x/y\phi$$

if and only if there is another set S' such that

$$\mathfrak{M} \models_{S'} \phi$$

[1] Before [19] it was an open question whether a compositional semantics can be given to independence friendly logic.

and the following "transition"-conditions hold:

- If $s \in S$, then there is $s' \in S'$ such that if z is a variable other than x, then $s(z) = s'(z)$.
- If $s' \in S'$, then there is $s \in S$ such that if z is a variable other than x, then $s(z) = s'(z)$.
- If $s, s' \in S'$ and $s(z) = s'(z)$ for all variables other than y or x, then $s(x) = s'(x)$.

In a sense, independence friendly logic is a logical formalism suggested by the functional dependence relation, but its origin is in game theoretical semantics, not in dependence relations. With dependence logic the situation is different. It was directly inspired by the functional dependence relation introduced by Wilfrid Hodges.

Peter van Emde Boas pointed out to the second author in the fall of 2005 that the functional dependence behind dependence logic is known in database theory [2]. This led the second author to realize—eventually—that the dependence we are talking about here is not just about variables in logic but a much more general phenomenon, covering such diverse areas as algebra, statistics, computer science, medicine, biology, social science, etc.

As Johan van Benthem points out in (4.1), there are different dependence intuitions. Of course the same is true of intuitions about independence. For some time it was not clear what would be the most natural concept of *independence*. There was the obvious but rather weak form of independence of x from y as dependence of x on some variable z other than y. Eventually a strong form of independence was introduced in [15], which has led to a breakthrough in our understanding of dependence relations and their role.

We give an overview of some developments in dependence logic (Sect. 4.2) and independence logic (Sect. 4.3). This is a tiny selection, intended for a newcomer, from a rapidly growing literature on the topic. Furthermore, in Sect. 4.4 we discuss conditional independence atoms and we prove a novel result—that is, that conditional and non-conditional independence logic are equivalent. Finally, in Sect. 4.6 we briefly discuss an application of our logics to belief representation.

4.2 Functional Dependence

The approach of [29] is that one should look for the strongest concept of dependence and use it to define weaker versions. Conceivably one could do the opposite, start from the weakest and use it to define stronger and strong concepts. The weakest dependence concept—whatever it is—did not offer itself immediately, so the strongest was more natural to start with. The wisdom of focusing in the extremes lies in the hope that the extremes are most likely to manifest simplicity and robustness, which would make them susceptible to a theoretical study.

Let us start with the strongest form of dependence, functional dependence. We use the vector notation $\vec{x}$ for finite sequences $x_1, \ldots, x_n$ of variables.[2] We add to first order logic[3] new atomic formulas

$$=\!(\vec{y}, \vec{x}), \tag{4.2}$$

with the intuitive meaning

the $\vec{y}$ totally determine the $\vec{x}$.

In other words, the meaning of (4.2) is that the values of the variables $\vec{y}$ functionally determine the values of the variables $\vec{x}$. We think of the atomic formulas (4.2) on a par with the atomic formula $x = y$. In particular, the idea is that the formula (4.2) is a purely logical expression, not involving any non-logical symbols, in particular no function symbol for the purported function manifesting the functional dependence.

The best way to understand the concept (4.2) is to give it exact semantics: To this end, suppose $\mathfrak{M}$ is a model. Suppose S is a set of assignments into M (or a *team* as such sets are called). We define:

Definition 4.1 The team S satisfies $=\!(\vec{y}, \vec{x})$ in $\mathfrak{M}$, in symbols

$$\mathfrak{M} \models_S =\!(\vec{y}, \vec{x})$$

if

$$\forall s, s' \in S(s(\vec{y}) = s'(\vec{y}) \to s(\vec{x}) = s'(\vec{x})). \tag{4.3}$$

One may ask, why not define the meaning of $=\!(y, x)$ as “there is a function which maps y to x”? The answer is that if we look at the meaning of $=\!(y, x)$ under *one* assignment s, then there *always* is a function f mapping $s(y)$ to $s(x)$, namely the function $\{(s(y), s(x))\}$, and if we look at the meaning of $=\!(y, x)$ under *many* assignments, a team, then (4.3) is indeed equivalent to the statement that there is a function mapping $s(y)$ to $s(x)$ for all s in the team.

A special case of $=\!(\vec{y}, \vec{x})$ is $=\!(\vec{x})$, the *constancy atom*. The intuitive meaning of this atom is that the value of $\vec{x}$ is constant in the team. It results from $=\!(\vec{y}, \vec{x})$ when $\vec{y}$ is the empty sequence.

Functional dependence has been studied in database theory and some basic properties, called **Armstrong’s Axioms** have been isolated [2]. These axioms state the following properties of $=\!(\vec{y}, \vec{x})$:

(A1) $=\!(\vec{x}, \vec{x})$. Anything is functionally dependent of itself.

(A2) If $=\!(\vec{y}, \vec{x})$ and $\vec{y} \subseteq \vec{z}$, then $=\!(\vec{z}, \vec{x})$. Functional dependence is preserved by increasing input data.

[2] Or attributes, something that has a value.

[3] The basic ideas can be applied to almost any logic, especially to modal logic.

(A3) If $\vec{y}$ is a permutation of $\vec{z}$, $\vec{u}$ is a permutation of $\vec{x}$, and $=(\vec{z}, \vec{x})$, then $=(\vec{y}, \vec{u})$. Functional dependence does not look at the order of the variables.

(A4) If $=(\vec{y}, \vec{z})$ and $=(\vec{z}, \vec{x})$, then $=(\vec{y}, \vec{x})$. Functional dependences can be transitively composed.

The following result is well-known in the database community and included in textbooks of database theory[4]:

Theorem 4.2 [2] *The axioms* (A1)–(A4) *are complete in the sense that a relation* $=(\vec{y}, \vec{x})$ *follows by the rules* (A1)–(A4) *from a set* Σ *of relations of the same form if and only if every team which satisfies* Σ *satisfies* $=(\vec{y}, \vec{x})$.

Proof Suppose $=(\vec{y}, \vec{x})$ does not follow by the rules from a set Σ of atoms. Let V be the set of variables z such that $=(\vec{y}, z)$ follows by the rules from Σ. Let W be the remaining variables in $\Sigma \cup \{=(\vec{y}, \vec{x})\}$. Thus $\vec{x} \cap W \neq \emptyset$. Consider the model $\{0, 1\}$ of the empty vocabulary and the team

The variables in V				The variables in W			
0	0	...	0	0	...	...	0
0	0	...	0	1	1	...	1

The atom $=(\vec{y}, \vec{x})$ is not true in this team, because $\vec{y} \subseteq V$ and $\vec{x} \cap W \neq \emptyset$. Suppose then $=(\vec{v}, \vec{w})$ is one of the assumptions. If each v is in V, then so is each w so they all get value 0. On the other hand, if some v is in W, it gets in this team two values, so it cannot violate dependence. □

We now extend the truth definition (Definition 4.1) to the full first order logic augmented by the dependence atoms $=(\vec{x}, \vec{y})$. To this end, let $s(a/x)$ denote the assignment which agrees with s except that it gives x the value a. We define for formulas which have negation in front of atomic formulas only:

$$\left.\begin{aligned}
&\mathfrak{M} \models_S x = y && \iff \forall s \in S(s(x) = s(y)).\\
&\mathfrak{M} \models_S \neg x = y && \iff \forall s \in S(s(x) \neq s(y)).\\
&\mathfrak{M} \models_S R(x_1, \ldots, x_n) && \iff \forall s \in S((s(x_1), \ldots, s(x_n)) \in R^{\mathfrak{M}}).\\
&\mathfrak{M} \models_S \neg R(x_1, \ldots, x_n) && \iff \forall s \in S((s(x_1), \ldots, s(x_n)) \notin R^{\mathfrak{M}}).\\
&\mathfrak{M} \models_S \phi \wedge \psi && \iff \mathfrak{M} \models_S \phi \text{ and } \mathfrak{M} \models_S \psi.\\
&\mathfrak{M} \models_S \phi \vee \psi && \iff \text{There are } S_1 \text{ and } S_2 \text{ such that}\\
& && \quad\; S = S_1 \cup S_2, \mathfrak{M} \models_{S_1} \phi, \text{ and } \mathfrak{M} \models_{S_2} \psi.\\
&\mathfrak{M} \models_S \exists x \phi && \iff \mathfrak{M} \models_{S'} \phi \text{ for some } S' \text{ such that}\\
& && \quad\; \forall s \in S\, \exists a \in M(s(a/x) \in S')\\
&\mathfrak{M} \models_S \forall x \phi && \iff \mathfrak{M} \models_{S'} \phi \text{ for some } S' \text{ such that}\\
& && \quad\; \forall s \in S\, \forall a \in M(s(a/x) \in S')
\end{aligned}\right\} \quad (4.4)$$

It is easy to see that for formulas not containing any dependence atoms, that is, for pure first order formulas ϕ,

[4] See e.g. [26].

$$\mathfrak{M} \models_{\{s\}} \phi \iff \mathfrak{M} \models_s \phi$$

and

$$\mathfrak{M} \models_S \phi \iff \forall s \in S(\mathfrak{M} \models_s \phi),$$

where $\mathfrak{M} \models_s \phi$ has its usual meaning. This shows that the truth conditions (4.4) agree with the usual Tarski truth conditions for first order formulas. Thus considering the "plural state" S rather than individual "states" s makes no difference for first order logic, but it makes it possible to give the dependence atoms $=(\vec{x}, \vec{y})$ their intended meaning.

What about axioms for non-atomic formulas of dependence logic? Should we adopt new axioms, apart from the Armstrong Axioms [A1–A4]? There is a problem! Consider the sentence

$$\exists x \forall y \exists z (=(z, y) \land \neg z = x). \tag{4.5}$$

It is easy to see that this sentence is satisfied by a team in a model $\mathfrak{M}$ if and only M is infinite. As a result, by general considerations going back to Gödel's Incompleteness Theorem, the semantic consequence relation

$$\phi \models \psi \iff \forall \mathfrak{M} \forall S(\mathfrak{M} \models_S \phi \rightarrow \mathfrak{M} \models_S \psi)$$

is non-arithmetical. Thus there cannot be any completeness theorem in the usual sense. However, this does not prevent us from trying to find axioms and rules which are as complete as possible. This is what is done in [24], where a complete axiomatization is given for *first order* consequences of dependence logic sentences. The axioms are a little weaker than standard first order axioms when applied to dependence formulas, but on the other hand there are two special axioms for the purpose of dealing with dependence atoms as parts of formulas in a deduction. Rather than giving all details (which can be found in [24]) we give just an example of the use of both new rules.

Suppose we are given ϵ, x, y and f, and we have already concluded, in the middle of some argument, the following:

if $\epsilon > 0$, *then there is* $\delta > 0$ *depending only on* ϵ *such that*
if $|x - y| < \delta$, *then* $|f(x) - f(y)| < \epsilon$.

By merely logical reasons we should be able to conclude

There is $\delta > 0$ *depending only on* ϵ *such that*
if $\epsilon > 0$ *and* $|x - y| < \delta$, *then* $|f(x) - f(y)| < \epsilon$.

Note that "*depending only on* ϵ" has moved from inside the implication to outside of it. The new rule of dependence logic, isolated in [24], which permits this, is called

Dependence Distribution Rule. Neither first order rules nor Armstrong's Axioms give this because neither of them gives any clue of how to deal with dependence atoms as parts of bigger formulas.

Here is another example of inference in dependence logic: Suppose we have arrived at the following formula in the middle of some argument:

$$\begin{aligned}&\textit{For every } x \textit{ and every } \epsilon > 0 \textit{ there is } \delta > 0 \textit{ depending only on } \epsilon\\ &\textit{such that for all } y, \textit{ if } |x - y| < \delta, \textit{ then } |f(x) - f(y)| < \epsilon.\end{aligned}$$

On merely logical grounds we should be able to make the following conclusion:

$$\begin{aligned}&\textit{For every } x \textit{ and every } \epsilon > 0 \textit{ there is } \delta > 0\\ &\textit{such that for all } y, \textit{ if } |x - y| < \delta, \textit{ then } |f(x) - f(y)| < \epsilon,\\ &\textit{and moreover, for any other } x' \textit{ and } \epsilon' > 0 \textit{ there is } \delta' > 0\\ &\textit{such that for all } y', \textit{ if } |x' - y'| < \delta', \textit{ then } |f(x') - f(y')| < \epsilon\\ &\textit{and if } \epsilon = \epsilon', \textit{ then } \delta = \delta'.\end{aligned}$$

The new rule, isolated in [24] which permits this step is called *Dependence Elimination Rule*, because the dependence atom "*depending only on* ϵ" has been entirely eliminated. The conclusion is actually first order, that is, without any occurrence of dependence atoms.

The first author [11] has given an alternative complete axiomatization, not for first order consequences of dependence sentences, but for dependence logic consequences of first order sentences. Clearly, more results about partial axiomatizations of the logical consequence relation in dependence logic can be expected in the near future.

An important property of dependence logic is the *downward closure* [20]: If $\mathfrak{M} \models_S \phi$ and $S' \subseteq S$, then $\mathfrak{M} \models_{S'} \phi$. It is a trivial matter to prove this by induction on the length of the formula. Once the downward closure is established it is obvious that we are far from having a negation in the sense of classical logic. Intuitively, dependence is a restriction of freedom (of values of variables in assignments). When the team gets smaller there is even less freedom. This intuition about the nature of dependence prevails in all the logical operations of dependence logic. Since dependence formulas are easily seen to be representable in existential second order logic, the following result shows that downward closure is really *the* essential feature of dependence logic:

Theorem 4.3 [23] *Let us fix a vocabulary L and an n-ary predicate symbol $S \notin L$. Then:*

- *For every L-formula $\phi(x_1, ..., x_n)$ of dependence logic there is an existential second order $L \cup \{S\}$-sentence $\Phi(S)$, closed downward with respect to S, such that for all L-structures M and all teams X:*

$$\mathfrak{M} \models_X \phi(x_1, \ldots, x_n) \iff \mathfrak{M} \models \Phi(X). \tag{4.6}$$

- *For every existential second order $L \cup \{S\}$-sentence $\Phi(S)$, closed downward with respect to S, there exists an L-formula $\phi(x_1, \ldots, x_n)$ of dependence logic such that* (4.6) *holds for all L-structures M and all teams $X \neq \emptyset$.*

This shows that dependence logic is maximal with respect to the properties of being expressible in existential second order logic and being downward closed. This theorem is also the source of the main model theoretical properties of dependence logic. The Downward Löwenheim-Skolem Theorem, the Compactness Theorem and the Interpolation Theorem are immediate corollaries. Also, when the above theorem is combined with the Interpolation Theorem of first order logic, we get the fact that dependence logic sentences ϕ for which there exists a dependence logic sentence ψ such that for all $\mathfrak{M}$

$$\mathfrak{M} \models \psi \iff \mathfrak{M} \not\models \phi$$

are first order definable. So not only does dependence logic not have the classical negation, the only sentences that have a classical negation are the first order sentences.

4.3 Independence Logic

Independence logic was introduced in [15]. Before going into the details, let us look at the following precedent:

In [3] Johan van Benthem suggested, as an example of an "other kind of correlation" than functional dependence, the following dependence relation for a team S in a model $\mathfrak{M}$:

$$\exists a \in M \exists b \in M(\{s(x) : s \in S, s(y) = a\} \neq \{s(x) : s \in S, s(y) = b\}). \tag{4.7}$$

The opposite of this would be

$$\forall a \in M \forall b \in M(\{s(x) : s \in S, s(y) = a\} = \{s(x) : s \in S, s(y) = b\}), \tag{4.8}$$

which is a kind of independence of x from y, for if we take $s \in S$ and we are told what $s(y)$ is, we have learnt nothing about $s(x)$, because for each $a \in M$ the set

$$\{s(x) : s \in S, s(y) = a\}$$

is the same. This is the idea behind the independence atom $\vec{x} \perp \vec{y}$: the values of $\vec{x}$ should not reveal anything about the values of $\vec{y}$ and vice versa. More exactly, suppose $\mathfrak{M}$ is a model and S is a team of assignments into M. We define:

Definition 4.4 A team S satisfies the atomic formula $\vec{x} \perp \vec{y}$ in $\mathfrak{M}$ if

$$\forall s, s' \in S \exists s'' \in S(s''(\vec{y}) = s(\vec{y}) \wedge s''(\vec{x}) = s'(\vec{x})). \tag{4.9}$$

We can immediately observe that a constant variable is independent of every variable, including itself. To see this, suppose x is constant in S. Let y be any variable, possibly $y = x$. If $s, s' \in S$ are given, we need $s'' \in S$ such that $s''(x) = s(x)$ and $s''(y) = s'(y)$. We can simply take $s'' = s'$. Now $s''(x) = s(x)$, because x is constant in S. Of course, $s''(y) = s'(y)$. Conversely, if x is independent of every variable, it is clearly constant, for it would have to be independent of itself, too. So we have

$$=\!(\vec{x}) \iff \vec{x} \perp \vec{x}.$$

We can also immediately observe the symmetry of independence, because criterion (4.9) is symmetrical in x and y. More exactly, $s''(y) = s(y) \wedge s''(x) = s'(x)$ and $s''(x) = s'(x) \wedge s''(y) = s(y)$ are trivially equivalent.

Dependence atoms were governed by Armstrong's Axioms. Independence atoms have their own axioms introduced in the context of random variables in [14]:

Definition 4.5 The following rules are the **Independence Axioms**

1. $\vec{x} \perp \emptyset$ (Empty Set Rule).
2. If $\vec{x} \perp \vec{y}$, then $\vec{y} \perp \vec{x}$ (Symmetry Rule).
3. If $\vec{x} \perp \vec{y}\vec{z}$, then $\vec{x} \perp \vec{y}$ (Weakening Rule).
4. If $\vec{x} \perp \vec{x}$, then $\vec{x} \perp \vec{y}$ (Constancy Rule).
5. If $\vec{x} \perp \vec{y}$ and $\vec{x}\vec{y} \perp \vec{z}$, then $\vec{x} \perp \vec{y}\vec{z}$ (Exchange Rule).

Note that $xy \perp xy$ is derivable from $x \perp x$ and $y \perp y$, by means of the Empty Set Rule, the Constancy Rule and the Exchange Rule.

It may seem that independence must have much more content than what these four axioms express, but they are actually complete in the following sense[5]:

Theorem 4.6 (Completeness of the Independence Axioms, [14]) *If T is a finite set of independence atoms of the form $\vec{u} \perp \vec{v}$ for various $\vec{u}$ and $\vec{v}$, then $\vec{y} \perp \vec{x}$ follows from T according to the above rules if and only if every team that satisfies T also satisfies $\vec{y} \perp \vec{x}$.*

Proof We adapt the proof of [14] into our framework. Suppose $\vec{x} \perp \vec{y}$ follows semantically from Σ but does not follow by the above rules. W.l.o.g. Σ is closed under the rules. We may assume that $\vec{x}$ and $\vec{y}$ are minimal, that is, if $\vec{x}\,' \subseteq \vec{x}$ and $\vec{y}\,' \subseteq \vec{y}$ and at least one containment is proper, then if $\vec{x}\,' \perp \vec{y}\,'$ follows from Σ semantically, it also follows by the rules. It is easy to see that if $\Sigma \models u \perp u$, then $\Sigma \vdash u \perp u$.

Suppose $\vec{x} = (x_1, \ldots, x_l)$ and $\vec{y} = (y_1, \ldots, y_m)$. Let $\vec{z} = (z_1, \ldots, z_k)$ be the remaining variables. W.l.o.g., $l \geq 1$ and $m \geq 1$, $x_1 \perp x_1 \notin \Sigma$, and $x_1 \notin \{y_1, \ldots, y_m\}$.

[5] This was originally proved for random variables in [14] and then adapted for databases in [22].

We construct a team S in a 2-element model $M = \{0, 1\}$ of the empty vocabulary as follows: We take to S every $s : \vec{x}\vec{y}\vec{z} \to M$, which satisfies $s(u) = 0$ for u such that $u \perp u \in \Sigma$ and in addition

$$s(x_1) = \text{ the number of ones in } s[\{x_2, \ldots, x_l, y_1, \ldots, y_m\}] \bmod 2$$

Claim 1 $\vec{x} \perp \vec{y}$ is not true in S. Suppose otherwise. Consider the following two assignments in S:

	x_1	other x_i	y_1	other y_i	other
s	1	0	1	0	0
s'	0	0	0	0	0

If s'' is such that $s''(\vec{x}) = s(\vec{x})$ and $s''(\vec{y}) = s'(\vec{y})$, then $s'' \notin S$. Claim 1 is proved.

Claim 2 S satisfies all the independence atoms in Σ. Suppose $\vec{v} \perp \vec{w} \in S$. If either $\vec{v}$ or $\vec{w}$ contains only variables in Z, then the claim is trivial, as then either $\vec{v}$ or $\vec{w}$ has in S all possible binary sequences. So let us assume that both $\vec{v}$ and $\vec{w}$ meet $\vec{x}\vec{y}$. If $\vec{v}\vec{w}$ does not cover all of $\vec{x}\vec{y}$, then S satisfies $\vec{v} \perp \vec{w}$, because we can fix parity on the variable in $\vec{x}\vec{y}$ which does not occur in $\vec{v}\vec{w}$. So let us assume $\vec{v}\vec{w}$ covers all of $\vec{x}\vec{y}$. Thus $\vec{v} = \vec{x}'\vec{y}'\vec{z}'$ and $\vec{w} = \vec{x}''\vec{y}''\vec{z}''$, where $\vec{x}'\vec{x}'' = \vec{x}$ and $\vec{y}'\vec{y}'' = \vec{y}$. W.l.o.g., $\vec{x}' \neq \emptyset$ and $\vec{x}'\vec{y}' \neq \vec{x}\vec{y}$. By minimality $\vec{x}' \perp \vec{y}' \in \Sigma$ and $\vec{x}'' \perp \vec{y} \in \Sigma$. Since $\vec{v} \perp \vec{w} \in \Sigma$, a couple of applications of the Exchange and Weakening Rules gives $\vec{x}'\vec{y}' \perp \vec{x}''\vec{y}'' \in \Sigma$. But then $\vec{x}'\vec{x}'' \perp \vec{y}'\vec{y}'' \in \Sigma$, contrary to the assumption. □

We can use the conditions (4.4) to extend the truth definition to the entire *independence logic*, i.e. the extension of first order logic by the independence atoms. Can we axiomatize logical consequence in independence logic? The answer is again no, and for the same reason as for dependence logic: Recall that the sentence (4.5) characterizes infinity and ruins any hope to have a completeness theorem for dependence logic. We can do the same using independence atoms:

Lemma 4.7 *The sentence*

$$\exists z \forall x \exists y \forall u \exists v (xy \perp uv \wedge (x = u \leftrightarrow y = v) \wedge \neg v = z) \tag{4.10}$$

is true exactly in infinite models.

The conclusion is that the kind of dependence relation needed for expressing infinity can be realized either by the functional dependence relation or by the independence relation. Another such example is parity in finite models. The following two sentences, the first one with a dependence atom and the second with an independence atom, both express the evenness of the size of a finite model:

$$\forall x \exists y \forall u \exists v (=(u, v) \wedge (x = v \leftrightarrow y = u) \wedge \neg x = y)$$

$$\forall x \exists y \forall u \exists v (xy \perp uv \wedge (x = v \leftrightarrow y = u) \wedge \neg x = y)$$

The fact that we could express, at will, both infinity and evenness by means of either dependence atoms or independence atoms, is not an accident. Dependence logic and independence logic have overall the same expressive power:

Theorem 4.8 *The following are equivalent:*

(1) *K is definable by a sentence of the extension of first order logic by the dependence atoms.*
(2) *K is definable by a sentence of the extension of first order logic by the independence atoms.*
(3) *K is definable in existential second order logic.*

Proof The equivalence of (1) and (3), a consequence of results in [8] and [31], as observed in [20], is proved in [29]. So it suffices to show that (1) implies (2). We give only the main idea. Sentences referred to in (1) have a normal form [29]. Here is an example of a sentence in such a normal form

$$\forall x \forall y \exists v \exists w (=(x, v) \wedge =(y, w) \wedge \phi(x, y, v, w)),$$

where $\phi(x, y, v, w)$ is a quantifier free first order formula. This sentence can be expressed in terms of independence atoms as follows:

$$\forall x \forall y \exists v \exists w (xv \perp y \wedge yw \perp xv \wedge \phi(x, y, v, w)). \qquad \square$$

Note that independence, as we have defined it, is not the negation of dependence. It is rather a very strong denial of dependence. However, there are uses of the concepts of dependence and independence where the negation of dependence *is* the same as independence. An example is vector spaces.

There is an earlier common use of the concept of independence in logic, namely the independence of a set Σ of axioms from each other. This is usually taken to mean that no axiom is provable from the remaining ones. By Gödel's Completeness Theorem this means the same as having for each axiom $\phi \in \Sigma$ a model of the remaining ones $\Sigma \setminus \{\phi\}$ in which ϕ is false. This is not so far from the independence concept $\vec{y} \perp \vec{x}$. Again, the idea is that from the truth of $\Sigma \setminus \{\phi\}$ we can say nothing about the truth-value of ϕ. This is the sense in which Continuum Hypothesis (CH) is independent of ZFC. Knowing the ZFC axioms gives us no clue as to the truth or falsity of CH. In a sense, our independence atom $\vec{y} \perp \vec{x}$ is the familiar concept of independence transferred from the world of formulas to the world of elements of models, from truth values to variable values.

4.4 Conditional Independence

The independence atom $\vec{y} \perp \vec{x}$ turns out to be a special case of the more general atom $\vec{y} \perp_{\vec{x}} \vec{z}$, the intuitive meaning of which is that the variables $\vec{y}$ are totally independent of the variables $\vec{z}$ when the variables $\vec{x}$ are kept fixed (see [15]). Formally,

Definition 4.9 A team S satisfies the atomic formula $\vec{y} \perp_{\vec{x}} \vec{z}$ in $\mathfrak{M}$ if

$$\forall s, s' \in S(s(\vec{x}) = s'(\vec{x}) \rightarrow \exists s'' \in S(s''(\vec{x}\vec{y}) = s(\vec{x}\vec{y}) \wedge s''(\vec{z}) = s'(\vec{z}))).$$

Some of the rules that this "conditional" independence notion obeys are

Reflexivity: $\vec{x} \perp_{\vec{x}} \vec{y}$,
Symmetry: If $\vec{y} \perp_{\vec{x}} \vec{z}$, then $\vec{z} \perp_{\vec{x}} \vec{y}$,
Weakening: If $\vec{y}y' \perp_{\vec{x}} \vec{z}z'$, then $\vec{y} \perp_{\vec{x}} \vec{z}$,
First Transitivity: If $\vec{x} \perp_{\vec{z}} \vec{y}$ and $\vec{u} \perp_{\vec{z}\vec{x}} \vec{y}$, then $\vec{u} \perp_{\vec{z}} \vec{y}$,
Second Transitivity: If $\vec{y} \perp_{\vec{z}} \vec{y}$ and $\vec{z}\vec{x} \perp_{\vec{y}} \vec{u}$, then $\vec{x} \perp_{\vec{z}} \vec{u}$,
Exchange: If $\vec{x} \perp_{\vec{z}} \vec{y}$ and $\vec{x}\vec{y} \perp_{\vec{z}} \vec{u}$, then $\vec{x} \perp_{\vec{z}} \vec{y}\vec{u}$.

Are these axioms complete? More in general, is it possible to find a finite, decidable axiomatization for the consequence relation between conditional independence atoms?

The answer is negative. Indeed, in [16, 17] Hermann proved that the consequence relation between conditional independence atoms is undecidable; and as proved by Parker and Parsaye-Ghomi in [28], it is not possible to find a finite and complete axiomatization for these atoms. However, the consequence relation is recursively enumerable, and in [27] Naumov and Nicholls developed a proof system for it.

The logic obtained by adding conditional independence atoms to first order logic will be called in this paper *conditional independence logic*. It is clear that it contains (nonconditional) independence logic; and furthermore, as discussed in [15], it also contains dependence logic, since a dependence atom $=(\vec{x}, \vec{y})$ can be seen to be equivalent to $\vec{y} \perp_{\vec{x}} \vec{y}$. It is also easy to see that every conditional independence logic sentence is equivalent to some Σ_1^1 sentence, and therefore that conditional independence logic is equivalent to independence logic and dependence logic with respect to sentences.

But this leaves open the question of whether every conditional independence logic formula is equivalent to some independence logic one. In what follows, building on the analysis of the expressive power of conditional independence logic of [10],[6] we prove that independence logic and conditional independence logic are indeed equivalent.

In order to give our equivalence proof we first need to mention two other atoms, the inclusion atom $\vec{x} \subseteq \vec{y}$ and the exclusion atom $\vec{x} \mid \vec{y}$. These atoms correspond to the database-theoretic inclusion [4, 9] and exclusion [5] dependencies, and hold in a

[6] In that paper, conditional independence logic is simply called "independence logic". After all, the two logics are equivalent.

team if and only if no possible value for $\vec{x}$ is also a possible value for $\vec{y}$ and if every possible value for $\vec{x}$ is a possible value for $\vec{y}$ respectively. More formally,

Definition 4.10 A team S satisfies the atomic formula $\vec{x} \subseteq \vec{y}$ in $\mathfrak{M}$ if

$$\forall s \in S \exists s' \in S(s'(\vec{y}) = s(\vec{x}))$$

and it satisfies the atomic formula $\vec{x} \mid \vec{y}$ in $\mathfrak{M}$ if

$$\forall s, s' \in S(s(\vec{x}) \neq s'(\vec{y})).$$

As proved in [10],

1. Exclusion logic (that is, first order logic plus exclusion atoms) is equivalent to dependence logic;
2. Inclusion logic (that is, first order logic plus inclusion atoms) is not comparable with dependence logic, but is contained in (nonconditional) independence logic;
3. Inclusion/exclusion logic (that is, first-order logic plus inclusion and exclusion atoms) is equivalent to *conditional* independence logic (that is, first-order logic plus conditional independence atoms $\vec{y} \perp_{\vec{x}} \vec{z}$).

Thus, if we can show that exclusion atoms can be defined in terms of (nonconditional) independence atoms and of inclusion atoms, we can obtain at once that independence logic contains conditional independence logic (and, therefore, is equivalent to it). But this is not difficult: indeed, the exclusion atom $\vec{x} \mid \vec{y}$ is equivalent to the expression

$$\exists \vec{z}(\vec{x} \subseteq \vec{z} \wedge \vec{y} \perp \vec{z} \wedge \vec{y} \neq \vec{z}).$$

This can be verified by checking the satisfaction conditions of this formula. But more informally speaking, the reason why this expression is equivalent to $\vec{x} \mid \vec{y}$ is that it states that that every possible value of $\vec{x}$ is also a possible value for $\vec{z}$, that $\vec{y}$ and $\vec{z}$ are independent (and therefore, any possible value of $\vec{y}$ must occur together with any possible value of $\vec{z}$), and that $\vec{y}$ is always different from $\vec{z}$. Such a $\vec{z}$ may exist if and only if no possible value of $\vec{x}$ is also a possible value of $\vec{y}$, that is, if and only if $\vec{x} \mid \vec{y}$ holds.

Hence we may conclude at once that

Theorem 4.11 *Every conditional independence logic formula is equivalent to some independence logic formula.*

In [10] it was also shown the following analogue of Theorem 4.3:

Theorem 4.12 *Let us fix a vocabulary L and an n-ary predicate symbol $S \notin L$. Then:*

- *For every L-formula $\phi(x_1, \ldots, x_n)$ of conditional independence logic there is an existential second order $L \cup \{S\}$-sentence $\Phi(S)$ such that for all L-structures M and all teams X:*

$$\mathfrak{M} \models_X \phi(x_1, \ldots, x_n) \iff \mathfrak{M} \models \Phi(X). \tag{4.11}$$

- *For every existential second order $L \cup \{S\}$-sentence $\Phi(S)$ there exists an L-formula $\phi(x_1, \ldots, x_n)$ of conditional independence logic such that (4.11) holds for all L-structures M and all teams $X \neq \emptyset$.*

Due to the equivalence between independence logic and conditional independence logic, the same result holds if we only allow nonconditional independence atoms. In particular, this implies that over finite models independence logic captures precisely the NP properties of teams.

4.5 Further Expressivity Results

The results mentioned in the above section left open the question of the precise expressive power of inclusion logic. This was answered in [14], in which a connection was found between inclusion logic and positive greatest fixed point logic GFP$^+$. In brief, GFP$^+$ is the logic obtained by adding to the language of first order logic the operator

$$[\mathrm{gfp}_{R,\vec{x}} \psi(R, \vec{x})](\vec{t}),$$

which asserts that the value of $\vec{t}$ is in the greatest fixed point of the operator $O(R) = \{\vec{a} : \psi(R, \vec{a})\}$,[7] and further requiring that no such operator occurs negatively.

Over finite models, it is known by [21] that this logic is equivalent to the better-known *least fixed point logic* LFP, which captures PTIME over linearly ordered models [21, 30]. Thus, the same is true of inclusion logic: more precisely,

Theorem 4.13 *A class of linearly ordered finite models is definable in inclusion logic if and only if it can be recognized in polynomial time.*

As dependence logic is equivalent to existential second order logic over sentences, it follows at once that there exists a fragment of dependence logic which also captures PTIME. Which fragment may it be? This is answered—in a slightly different setting—by Ebbing, Kontinen, Müller and Vollmer in [7] by introducing the class of *D^*-Horn formulas* and proving that they capture PTIME over *successor structures*.[8]

The complexity-theoretic properties of further fragments of these logics have been studied in the papers [6] and [13], in which *hierarchy theorems* are developed for the restrictions of these logics to dependencies of certain maximum lengths or to maximum numbers of quantifiers. We will not however present here a complete summary of these results, and we refer the interested readers to these papers for the details.

[7] In order to guarantee that such a fixed point exist, R is required to appear only positively in ψ.

[8] That is, over finite structures with a built-in successor operator and two constants for the least and greatest elements.

Finally, yet another direction of investigation consists in the search for *weak* dependency notions, which if added to the language of first order logic do not increase its expressive power (with respect to sentences). This problem in studied in [12], in which a fairly general class of such dependencies is found. One particularly interesting result along these lines is that the contradictory negations of functional dependence, inclusion, exclusion and (conditional or non-conditional) independence atoms are all weak and do not increase the expressive power of first order logic. This is somewhat surprising, since as we saw the corresponding non-negated atoms greatly increase the expressivity of first order logic instead; and the study of the manner in which applying the contradictory negation acts on dependence notions and on the logics they generate is an intriguing and largely unexplored avenue of research.

4.6 Belief Representation and Belief Dynamics

Given a model $\mathfrak{M}$, a variable assignment s admits a natural interpretation as the representation of a possible *state of things*, where, for every variable v, the value $s(v)$ corresponds to a specific *fact* concerning the world. To use the example discussed in Chap. 7 of [11], let the elements of $\mathfrak{M}$ correspond to the participants to a competition: then the values of the variables x_1, x_2 and x_3 in an assignment s may correspond respectively to the first-, second- and the third-placed players.

With respect to the usual semantics for first order logic, a first order formula represents a *condition* over assignments. For example, the formula

$$\phi(x_1, x_2, x_3) := (\neg x_1 = x_2) \wedge (\neg x_2 = x_3) \wedge (\neg x_1 = x_3)$$

represents the (very reasonable) assertion according to which the winner, the second-placed player and the third-placed player are all distinct.

Now, a team S, being a set of assignments, represents a set of states of things. Hence, a team may be interpreted as the *belief set* of an agent α: $s \in S$ if and only if the agent α believes s to be possible. Moving from assignments to teams, it is possible to associate to each formula ϕ and model $\mathfrak{M}$ the family of teams $\{S : \mathfrak{M} \models_S \phi\}$, and this allows us to interpret formulas as *conditions over belief sets*: in our example, $\mathfrak{M} \models_S \phi(x_1, x_2, x_3)$ if and only if $\mathfrak{M} \models_s \phi(x_1, x_2, x_3)$ for all $s \in S$, that is, if and only if our agent α believes that the winner, the second-placed player and the third-placed player will all be distinct.

However, there is much that first order logic cannot express regarding the beliefs of our agent. For example, there is no way to represent the assertion that the agent α *knows* who the winner of the competition will be: indeed, suppose that a first order formula θ represents such a property, and let s_1 and s_2 be any two assignments with $s_1(x_1) \neq s_2(x_1)$, corresponding to two possible states of things which disagree with respect to the identity of the winner. Then, for $S_1 = \{s_1\}$ and $S_2 = \{s_2\}$, we should have that $\mathfrak{M} \models_{S_1} \theta$ and that $\mathfrak{M} \models_{S_2} \theta$: indeed, both S_1 and S_2 correspond to belief sets in which the winner is known to α (and is respectively $s_1(x_1)$ or $s_2(x_1)$). But since a team S satisfies a first order formula if and only if all of its assignments satisfy it, this implies that $M \models_{S_1 \cup S_2} \theta$; and this is unacceptable, because if our agent α believes both s_1 and s_2 to be possible then she does not know whether the winner will be $s_1(x_1)$ or $s_2(x_1)$.

How to represent this notion of knowledge? The solution, it is easy to see, consists in adding *constancy atoms* to our language: indeed, $\mathfrak{M} \models_S =\!(x_1)$ if and only if for any two assignments $s, s' \in S$ we have that $s(x_1) = s'(x_1)$, that is, if and only if all states of things the agent α consider possible agree with respect to the identity of the winner of the competition. What if, instead, our agent could infer the identity of the winner from the identity of the second- and third-placed participants? Then we would have that $\mathfrak{M} \models_S =\!(x_2 x_3, x_1)$, since any two states of things which the agent considered possible and which agreed with respect to the identity of the second- and third-placed participants would also agree with respect to the identity of the winner. More in general, a dependence atom $=\!(\vec{y}, \vec{x})$ describes a form of *conditional knowledge*: $\mathfrak{M} \models_S =\!(\vec{y}, \vec{x})$ if and only if S corresponds to the belief state of an agent who would be able to deduce the value of $\vec{x}$ from the value of $\vec{y}$.

On the other hand, independence atoms represent situations of *informational independence*: for example, if $\mathfrak{M} \models_S x_1 \perp x_3$ then, by learning the identity of the third-placed player, our agent could infer nothing at all about the identity of the winner. Indeed, suppose that, according to our agent, it is possible that A will win (that is, there is a $s \in S$ with $s(x_1) = A$) and it is possible that B will place third (that is, there is a $s' \in S$ such that $s'(x_3) = B$). Then, by the satisfaction conditions of the independence atom, there is also a $s'' \in S$ such that $s''(x_1) = A$ and $s''(x_3) = B$: in other words, it is possible that A will be the winner *and* B will place third, and telling our agent that B will indeed place third will not allow her to remove A from her list of possible winners.

Thus, it seems that dependence and independence logic, or at least fragments thereof, may be interpreted as *belief description languages*. This line of investigation is pursued further in [11]: here it will suffice to discuss the interpretation of the *linear implication*[9] $\phi \multimap \psi$, a connective introduced in [1] whose semantics is given by

$$\mathfrak{M} \models_S \phi \multimap \psi \Leftrightarrow \text{ for all } S' \text{ such that } \mathfrak{M} \models_{S'} \phi \text{ it holds that } \mathfrak{M} \models_{S \cup S'} \psi.$$

How to understand this connective? Suppose that our agent α, whose belief state is represented by the team S, interacts with another agent β, whose belief state is represented by the team S': one natural outcome of this interaction may be represented by the team $S \cup S'$, corresponding to the set of all states of things that α *or* β consider possible. Then stating that a team S satisfies $\phi \multimap \psi$ corresponds to asserting that whenever our agent α interacts with another agent β whose belief state satisfies ϕ, the result of the interaction will be a belief state satisfying ψ: in other words, using

[9] The name "linear implication" is due to the similarity between the satisfaction conditions of this connective and the ones of the implication of linear logic. Another similarity is the following Galois connection: $\theta \models \phi \multimap \psi \iff \theta \vee \phi \models \psi$ [1].

the linear implication connective allows us to formulate predictions concerning the future *evolution* of the belief state of our agent.

One can, of course, consider other forms of interactions between agents and further connectives; and quantifiers can also be given natural interpretations in terms of belief updates (the universal quantifier $\forall v$, for example, can be understood in terms of the agent α *doubting* her beliefs about v). But what we want to emphasize here, beyond the interpretations of the specific connectives, is that team-based semantics offers a very general and powerful framework for the representation of beliefs and belief updates, and that notions of dependence and independence arise naturally under such an interpretation. This opens up some fascinating—and, so far, relatively unexplored—avenues of research, such as for example a more in-depth investigation of the relationship between dependence/independence logic and dynamic epistemic logic (DEL) and other logics of knowledge and belief; and, furthermore, it suggests that epistemic and doxastic ideas may offer some useful inspiration for the formulation and analysis of further notions of dependence and independence.

4.7 Concluding Remarks

We hope to have demonstrated that both dependence and independence can be given a logical analysis by moving in semantics from single states s to plural states S. Future work will perhaps show that allowing limited transitions from one plural state to another may lead to decidability results concerning dependence and independence logic, a suggestion of Johan van Benthem.

Furthermore, we proved the equivalence between conditional independence logic and independence logic, thus giving a novel contribution to the problem of characterizing the relations between extensions of dependence logic.

Finally, we discussed how team-based semantics may be understood as a very general framework for the representation of beliefs and belief updates and how notions of dependence and independence may be understood under this interpretation. This suggests the existence of intriguing connections between dependence and independence logic and other formalisms for belief knowledge representation, as well as a possible application for this fascinating family of logics.

References

1. Abramsky S, Väänänen J (2009) From IF to BI. Synthese 167(2):207–230. doi:10.1007/s11229-008-9415-6
2. Armstrong WW (1974) Dependency structures of data base relationships. Inf Process 74
3. van Benthem J (1997) Modal foundations for predicate logic. Log J IGPL 5(2):259–286 (electronic). doi:10.1093/jigpal/5.2.259
4. Casanova MA, Fagin R, Papadimitriou CH (1982) Inclusion dependencies and their interaction with functional dependencies. In: Proceedings of the 1st ACM SIGACT-SIGMOD sympo-

sium on principles of database systems, PODS'82. ACM, New York, NY, USA, pp 171–176. doi:10.1145/588111.588141

5. Casanova MA, Vidal VMP (1983) Towards a sound view integration methodology. In: Proceedings of the 2nd ACM SIGACT-SIGMOD symposium on principles of database systems, PODS'83. ACM, New York, NY, USA, pp 36–47. doi:10.1145/588058.588065
6. Durand A, Kontinen J (2012) Hierarchies in dependence logic. ACM Trans Comput Log (TOCL) 13(4):31. doi:http://dx.doi.org/10.1145/2362355.2362359
7. Ebbing J, Kontinen J, Müller J-S, Vollmer H (2012) A fragment of dependence logic capturing polynomial time. arXiv:1210.3321. URL: http://arxiv.org/abs/1210.3321
8. Enderton HB (1970) Finite partially-ordered quantifiers. Math Log Q 16(8):393–397. doi:10.1002/malq.19700160802
9. Fagin R (1981) A normal form for relational databases that is based on domains and keys. ACM Trans Database Syst 6:387–415. doi:10.1145/319587.319592
10. Galliani P (2012) Inclusion and exclusion dependencies in team semantics: on some logics of imperfect information. Ann Pure Appl Log 163(1):68–84. doi:10.1016/j.apal.2011.08.005
11. Galliani P (2012) The dynamics of imperfect information. PhD thesis, University of Amsterdam, September 2012. URL: http://dare.uva.nl/record/425951
12. Galliani P (2013) Upwards closed dependencies in team semantics. In: Puppis G, Villa T (eds) Proceedings fourth international symposium on games, automata, logics and formal verification, vol 119 of EPTCS, pp 93–106. doi:http://dx.doi.org/10.4204/EPTCS.119
13. Galliani P, Hannula M, Kontinen J (2012) Hierarchies in independence logic. In: Rocca SRD (ed) Computer science logic 2013 (CSL 2013), Leibniz international proceedings in informatics (LIPIcs), vol 23. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, pp 263–280. doi:http://dx.doi.org/10.4230/LIPIcs.CSL.2013.263
14. Galliani P, Hella L (2013) Inclusion logic and fixed point logic. In: Rocca SRD (ed) Computer science logic 2013 (CSL 2013), Leibniz international proceedings in informatics (LIPIcs), vol 23. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, pp 281–295. doi:http://dx.doi.org/10.4230/LIPIcs.CSL.2013.281
15. Grädel E, Väänänen J (2013) Dependence and independence. Studia Logica 101(2):399–410. doi:10.1007/s11225-013-9479-2
16. Herrmann C (1995) Corrigendum: corrigendum to "on the undecidability of implications between embedded multivalued database dependencies" [Inf Comput 122:221–235 (1995)]. Inf Comput 204(12):1847–1851. doi:10.1016/j.ic.2006.09.002
17. Herrmann C (1995) On the undecidability of implications between embedded multivalued database dependencies. Inf Comput 122(2):221–235. doi:10.1006/inco.1995.1148
18. Hintikka J, Sandu G (1989) Informational independence as a semantic phenomenon. In: Fenstad JE, Frolov IT, Hilpinen R (eds) Logic, methodology and philosophy of science, pp. 571–589. Elsevier, Amsterdam. doi:10.1016/S0049-237X(08)70066-1
19. Hodges W (1997) Compositional semantics for a language of imperfect information. Log J IGPL 5(4):539–563 (electronic). doi:10.1093/jigpal/5.4.539
20. Hodges W (1997) Some strange quantifiers. In: Structures in logic and computer science, Lecture Notes in Computer Science, vol 1261. Springer, Berlin, pp 51–65. doi:10.1007/3-540-63246-8_4
21. Immerman N (1986) Relational queries computable in polynomial time. Inf Control 68(1):86–104
22. Kontinen J, Link S, Väänänen J (2013) Independence in database relations. In: Proceedings of the 20th International workshop (WOLLIC) on logic, language, information, and computation. Lecture notes in computer science, vol 8071. Springer, pp 152–156
23. Kontinen J, Väänänen J (2009) On definability in dependence logic. J Log Lang Inf 18(3):317–332. doi:10.1007/s10849-009-9082-0
24. Kontinen J, Väänänen J (2013) Axiomatizing first order consequences in dependence logic. Ann Pure Appl Log 164:1101–1117. URL: http://arxiv.org/abs/1208.0176
25. Mann AL, Sandu G, Sevenster M (2011) Independence-friendly logic: A game-theoretic approach. In: London Mathematical Society lecture note series, vol 386. Cambridge University Press, Cambridge. doi:10.1017/CBO9780511981418

26. Mannila H, Räihä KJ (1992) The design of relational databases. Addison-Wesley, Cambridge
27. Naumov P, Nicholls B (2013) Re axiomatization of conditional independence. In: TARK, 2013. http://www2.mcdaniel.edu/pnaumov/papers/2013.tark.nn.pdf
28. Parker D Jr, Parsaye-Ghomi K (1980) Inferences involving embedded multivalued dependencies and transitive dependencies. In: Proceedings of the 1980 ACM SIGMOD international conference on management of data, ACM, pp 52–57. doi:10.1145/582250.582259
29. Väänänen J (2007) Dependence logic. In: London Mathematical Society student texts, vol 70. Cambridge University Press, Cambridge. doi:10.1017/CBO9780511611193
30. Vardi MY (1982) The complexity of relational query languages. In: Proceedings of the fourteenth annual ACM symposium on theory of computing. ACM, pp 137–146
31. Walkoe WJ Jr (1970) Finite partially-ordered quantification. J Symbolic Log 35:535–555. doi:10.2307/2271440