Annals of Pure and Applied Logic 163 (2012) 68–84

Contents lists available at SciVerse ScienceDirect

Annals of Pure and Applied Logic

journal homepage: www.elsevier.com/locate/apal

Inclusion and exclusion dependencies in team semantics — On some logics of imperfect information

Pietro Galliani *

Faculteit der Natuurwetenschappen, Wiskunde en Informatica, Institute for Logic, Language and Computation, Universiteit van Amsterdam, P.O. Box 94242, 1090 GE AMSTERDAM, The Netherlands

ARTICLE INFO

Article history:
Received 7 June 2011
Received in revised form 5 August 2011
Accepted 17 August 2011
Available online 25 September 2011
Communicated by P.J. Scott

MSC:
03B60
03C80
03C85

Keywords:
Dependence
Independence
Imperfect information
Team semantics
Model theory

ABSTRACT

We introduce some new logics of imperfect information by adding atomic formulas corresponding to *inclusion* and *exclusion* dependencies to the language of first order logic. The properties of these logics and their relationships with other logics of imperfect information are then studied. As a corollary of these results, we characterize the expressive power of independence logic, thus answering an open problem posed in Grädel and Väänänen, 2010 [9].

1. Introduction

The notions of dependence and independence are among the most fundamental ones considered in logic, in mathematics, and in many of their applications. For example, one of the main aspects in which modern predicate logic can be thought of as superior to medieval term logic is that the former allows for quantifier alternation, and hence can express certain complex patterns of dependence and independence between variables that the latter cannot easily represent.

Logics of imperfect information are a family of logical formalisms whose development arose from the observation that not all possible patterns of dependence and independence between variables may be represented in first order logic. Among these logics, *dependence logic* [20] is perhaps the one most suited for the analysis of the notion of dependence itself, since it isolates it by means of *dependence atoms* which correspond, in a very exact sense, to functional dependencies of the exact kind studied in database theory. The properties of this logic, and of a number of variants and generalizations thereof, have been the object of much research in recent years, and we cannot hope to give here an exhaustive summary of the known results. We will content ourselves, therefore, to recall in Section 2.1 those that will be of particular interest for the rest of this work.

Independence logic [9] is a recent variant of dependence logic. In this new logic, the fundamental concept that is being added to the first order language is not *functional dependence*, as for the case of dependence logic proper, but

* Tel.: +31 020 525 8260; fax: +31 20 525 5206.
E-mail address: pgallian@gmail.com.

0168-0072/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.apal.2011.08.005

informational independence; as we will see, this is achieved by considering *independence atoms* $y \perp_x z$, whose informal meaning corresponds to the statement according to which, for any fixed value of x, the sets of the possible values for y and z are independent. Just as dependence logic allows us to reason about the properties of functional dependence, independence logic does the same for this notion. Much is not known at the moment about independence logic; in particular, one open problem mentioned in [9] concerns the *expressive power* of this formalism over open formulas.

In this work, we will find an answer to this problem; and furthermore, as a means to do so, we will study some logics obtained by extending the language of first order logic along the same lines of dependence or independence logic.

2. Dependence and independence logic

2.1. Dependence logic

Dependence logic [20] is, together with independence-friendly (IF) logic [10,19], one of the most widely studied logics of imperfect information. In brief, it can be described as the extension of first order logic obtained by adding *dependence atoms* $=(t_1 \ldots t_n)$ to its language, with the informal meaning of "The value of the term t_n is functionally determined by the values of the terms $t_1 \ldots t_{n-1}$".

We will later recall the full definition of the *team semantics* of dependence logic, an adaptation of Hodges' compositional semantics for IF logic [12], and one of the three equivalent semantics for dependence logic described in [20]. It is worth noting already here, though, that the key difference between Hodges semantics and the usual Tarskian semantics is that in the former the satisfaction relation $\models$ associates to every first order model[1] M and formula ϕ a set of *teams*, that is, a set of sets of assignments, instead of just a set of assignments as in the latter.

As discussed in [13], the fundamental intuition behind Hodges' semantics is that a team is a representation of an *information state* of some agent: given a model M, a team X, and a suitable formula ϕ, the expression $M \models_X \phi$ asserts that, from the information that the "true" assignment s belongs to the team X, it is possible to infer that ϕ holds, or, in game-theoretic terms, that the verifier has a strategy τ which is winning for all plays of the game $G(\phi)$ which start from any assignment $s \in X$.

The satisfaction conditions for dependence atoms are then given by the following semantic rule **TS-dep**.

Definition 2.1 (*Dependence Atoms*)**.** Let M be a first order model, let X be a team over it, let $n \in \mathbb{N}$, and let $t_1 \ldots t_n$ be terms over the signature of M and with variables in $\mathrm{Dom}(X)$. Then the following holds.

TS-dep: $M \models_X =(t_1 \ldots t_n)$ if and only if, for all $s, s' \in X$ such that $t_i\langle s\rangle = t_i\langle s'\rangle$ for $i = 1 \ldots n-1$, $t_n\langle s\rangle = t_n\langle s'\rangle$.

This rule corresponds closely to the definition of *functional dependency* commonly used in database theory [4]: more precisely, if $X(t_1 \ldots t_n)$ is the relation $\{(t_1\langle s\rangle, \ldots, t_n\langle s\rangle) : s \in X\}$ then

$$M \models_X =(t_1 \ldots t_n) \Leftrightarrow X(t_1 \ldots t_n) \models \{t_1 \ldots t_{n-1}\} \rightarrow t_n,$$

where the right-hand expression states that, in the relation $X(t_1 \ldots t_n)$, the value of the last term t_n is a function of the values of $t_1 \ldots t_{n-1}$.

The following known results will be of some use for the rest of this work.

Theorem 2.2 (*Locality [20]*)**.** *Let M be a first order model and let ϕ be a dependence logic formula over the signature of M with free variables in $\vec{v}$. Then, for all teams X with domain $\vec{w} \supseteq \vec{v}$, if X' is the restriction of X to $\vec{v}$, then*

$$M \models_X \phi \Leftrightarrow M \models_{X'} \phi.$$

As an aside, it is worth pointing out that the above property does not hold for most variants of IF logic: for example, if $\mathrm{Dom}(M) = \{0, 1\}$ and $X = \{(x := 0, y := 0), (x := 1, y := 1)\}$, it is easy to see that $M \models_X (\exists z/y)z = y$, even though for the restriction X' of X to $\mathrm{Free}((\exists z/y)z = y) = \{y\}$ we have that $M \not\models_{X'} (\exists z/y)z = y$. This is a typical example of *signalling* [10,14], one of the most peculiar and, perhaps, problematic aspects of IF logic.

Theorem 2.3 (*Downwards Closure Property [20]*)**.** *Let M be a model, let ϕ be a dependence logic formula over the signature of M, and let X be a team over M with domain $\vec{v} \supseteq \mathrm{Free}(\phi)$ such that $M \models_X \phi$. Then, for all $X' \subseteq X$,*

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

Theorem 2.4 (*Dependence Logic Sentences and Σ_1^1 [20]*)**.** *For every dependence logic sentence ϕ, there exists a Σ_1^1 sentence Φ such that*

$$M \models_{\{\emptyset\}} \phi \Leftrightarrow M \models \Phi.$$

Conversely, for every Σ_1^1 sentence Φ, there exists a dependence logic sentence ϕ such that the above holds.

[1] In all of this paper, I will assume that first order models have at least two elements in their domain.