

From IF to BI

A Tale of Dependence and Separation

Samson Abramsky

Oxford University Computing Laboratory

Jouko Väänänen

ILLC Amsterdam

Abstract. We take a fresh look at the logics of informational dependence and independence of Hintikka and Sandu and Väänänen, and their compositional semantics due to Hodges. We show how Hodges' semantics can be seen as a special case of a general construction, which provides a context for a useful completeness theorem with respect to a wider class of models. We shed some new light on each aspect of the logic. We show that the natural propositional logic carried by the semantics is the logic of Bunched Implications due to Pym and O'Hearn, which combines intuitionistic and multiplicative connectives. This introduces several new connectives not previously considered in logics of informational dependence, but which we show play a very natural rôle, most notably intuitionistic implication. As regards the quantifiers, we show that their interpretation in the Hodges semantics is forced, in that they are the image under the general construction of the usual Tarski semantics; this implies that they are adjoints to substitution, and hence uniquely determined. As for the dependence predicate, we show that this is definable from a simpler predicate, of constancy or dependence on nothing. This makes essential use of the intuitionistic implication. The Armstrong axioms for functional dependence are then recovered as a standard set of axioms for intuitionistic implication. We also prove a full abstraction result in the style of Hodges, in which the intuitionistic implication plays a very natural rôle.

1. Introduction

Our aim in this paper is to take a fresh look at the logics of informational dependence and independence [11, 12, 33], and their compositional semantics due to Wilfrid Hodges [13, 14]. We shall focus on Dependence Logic, introduced by the second author [33].

The main objective of Hodges' work was to provide a compositional model-theoretic semantics for the IF-logic of Hintikka and Sandu [11, 12], which matched their "game-theoretical semantics". This was achieved by lifting the standard Tarski semantics of first-order formulas, given in terms of satisfaction in a structure with respect to an assignment to the free variables, to satisfaction by **sets of assignments**.

We seek a deeper understanding of Hodges' construction:

© 2008 *Kluwer Academic Publishers. Printed in the Netherlands.*

- First and foremost, what is going on? Where does the Hodges construction come from? Is it canonical in any way? Why does it work? What structures are really at play here?
- Because of the equivalence of Dependence Logic (or variants such as IF-logic) under this semantics to (a significant fragment of) second-order logic, there is no hope for a completeness theorem. But we may get a useful completeness theorem with respect to a wider class of models. Understanding the general algebraic context for the semantics points the way to such a completeness notion.
- We can also look for **representation theorems**, with some infinitary ingredients.

The results of our investigation are quite surprising conceptually (at least to us). The main points can be summarized as follows.

- We find a general context for Hodges’ construction. We shall not treat it in full generality here, as the general account is best stated in the language of categorical logic [16, 22], and we wish to avoid undue technicalities. However, we will indicate the possibilities for a general algebraic semantics, as the basis for a useful completeness theorem.
- We find that the natural propositional logic associated with the Hodges construction is the **logic of Bunched Implication** of Pym and O’Hearn [21, 24], which combines intuitionistic and multiplicative linear connectives. Although this is a highly non-classical logic, it reduces to the bivalent semantics for **sentences**.
- This not only yields a more natural view of the strangely asymmetric notions of conjunction and disjunction in the Hodges semantics (one is intuitionistic, while “disjunction” is actually multiplicative **conjunction!**), it also brings into prominence some connectives not previously considered in the setting of IF-logic or Dependence logic, in particular **intuitionistic implication**. This enables a novel analysis of the Dependence predicate of [33], as a Horn clause with respect to a more primitive predicate of single-valuedness. The well-known Armstrong axioms for functional dependence [1] then fall out as a standard axiomatization of intuitionistic (but not classical!) implication.
- Intuitionistic implication also plays a natural rôle in our version of a full abstraction theorem in the sense of Hodges.

- The construction is shown to lift the interpretation of the standard quantifiers in a canonical way, so that **quantifiers are uniquely determined as the adjoints to substitution** [16], just as in the standard Tarski semantics of first-order logic. This is also extended to characterizations of the dependence-friendly quantifiers of [33] as adjoints.

The plan of the remainder of the paper is as follows. In the next section we provide background on branching quantifiers, IF-logic, dependence logic, and Hodges’ semantics. Then in section 3 we show how the Hodges semantics is an instance of a general algebraic construction, in which the connectives of BI-logic arise naturally. In section 4, we show that the interpretation of the quantifiers in the Hodges construction is the canonical lift of the standard interpretation of the quantifiers as adjoints, and hence is uniquely determined. We also use the intuitionistic implication to show how the dependence-friendly quantifiers can be interpreted as certain adjoints. In section 5, we show how the intuitionistic implication arises naturally in the proof of a full abstraction theorem. In section 6, we show how the dependence predicate can be analyzed in terms of a more primitive predicate of single-valuedness, using the intuitionistic implication. This turns the “Armstrong axioms” into standard theorems of intuitionistic implicational logic. The final section outlines some further directions.

2. Dependence, Independence and Information Flow

We begin with a standard example: the formal definition of continuity for a function $f : \mathbb{R} \rightarrow \mathbb{R}$ on the real numbers.

$$\forall x. \forall \epsilon. \exists \delta. \forall x'. |x - x'| < \delta \Rightarrow |f(x) - f(x')| < \epsilon.$$

This definition is often explained in current calculus courses in terms of an “epsilon-delta game”.¹ The Adversary proposes a number, ϵ , as a measure of how close we must stay to the value of $f(x)$; we must then respond with a number, δ , such that, whenever the input is within the interval $(x - \delta, x + \delta)$, the output does indeed pass the ϵ -test of closeness to $f(x)$. Clearly, the choice of δ will depend on that of ϵ ; the nesting of the quantifiers expresses this dependency.

This is the definition of **global continuity** of f , expressed in terms of **local continuity** at every point x . This means that the choice of δ will depend, not only on ϵ , but on x also. Now consider the definition of **uniform continuity**:

$$\forall \epsilon. \exists \delta. \forall x. \forall x'. |x - x'| < \delta \Rightarrow |f(x) - f(x')| < \epsilon.$$

Here δ still depends on ϵ , **but must be chosen independently of x** . This variation in dependency is tracked syntactically by the different order of the quantifiers. Indeed, it seems that it was only after the distinction between pointwise and uniform notions of continuity, and, especially, convergence, had been clarified in 19th-century analysis, that the ground was prepared for the introduction of predicate calculus.

More generally, dependence or independence of bounds on various parameters is an important issue in many results on estimates in number theory and analysis. Hodges quotes a nice example from one of Lang's books [15] in [13].

Intuitively, there is an evident relation between these notions and that of **information flow**. Dependence indicates a form of information flow; independence is the **absence** of information flow.

2.1. BEYOND FIRST-ORDER LOGIC

It turns out that mere rearrangement of the order of quantifiers in first-order formulas is not sufficient to capture the full range of possibilities for informational dependence and independence. This was first realized almost 50 years ago, with Henkin's introduction of **branching quantifiers** [8]. The simplest case is the eponymous **Henkin quantifier**:

$$\left(\begin{array}{l} \forall x \exists y \\ \forall u \exists v \end{array} \right) A(x, y, u, v).$$

The intention is that y must be chosen depending on x , but **independently** of the choice of u ; while v must be chosen depending on u , but **independently** of the choice of x . The meaning of this formula can be explicated by introducing **Skolem functions** f and g : an equivalent formula will be

$$\exists f. \exists g. \forall x. \forall u. A(x, f(x), u, g(u)).$$

Here the constraints on dependencies are tracked by the dependence of the Skolem functions on certain variables, but not on others. Note that the Skolemized sentence is **second-order**; in fact, it belongs to the Σ_1^1 fragment of second-order logic.² This second-order rendition of the meaning of the Henkin quantifier cannot be avoided, in the sense that the Henkin quantifier strictly increases the expressive power of first-order logic, and in fact the extension of first-order logic with the Henkin quantifier is equivalent in expressive power to the Σ_1^1 fragment [8].

2.1.0.1. Examples

1. Consider

$$\left(\begin{array}{l} \forall x \ \exists y \\ \forall u \ \exists v \end{array} \right) (A(x) \rightarrow B(y)) \wedge (B(u) \rightarrow A(v)) \wedge [(x = v) \leftrightarrow (y = u)].$$

This expresses that A and B are equinumerous sets.

2. Now consider

$$\begin{aligned} \exists v. \left(\begin{array}{l} \forall x_1 \ \exists y_1 \\ \forall x_2 \ \exists y_2 \end{array} \right) (A(x_1) \rightarrow A(y_1)) \wedge [(x_2 = y_1) \rightarrow (y_2 = x_1)] \\ \wedge A(v) \wedge (A(x_1) \rightarrow (y_1 \neq v)). \end{aligned}$$

This expresses that A is an infinite set.

These examples show that the Henkin quantifier **is not expressible in first-order logic**.

2.2. FURTHER DEVELOPMENTS

The next major development was the introduction of IF-logic (“independence-friendly logic”) by Hintikka and Sandu [11]. The intention of IF-logic is to highlight informational dependence and independence. It provides a linear syntax for expressing branching quantification (and more), e.g. the Henkin quantifier can be written in linear notation as:

$$\forall x. \exists y. \forall u. (\exists v/x). A(x, y, u, v)$$

The “slashed quantifier” $(\exists v/x)$ has the intended reading “there exists a v **not depending on** x ”. Note the strange syntactic form of this quantifier, with its “outward-reaching” scope for x .

2.2.0.2. Dependence Logic A simplified approach was introduced by the second author, and developed extensively in the recent monograph [33]. The main novelty in the formulation of the logic is to use an atomic **dependence predicate**³ $D(x_1, \dots, x_n, x)$ which holds if x **depends on** x_1, \dots, x_n , **and only on these variables**. We can then define “dependence-friendly quantifiers” as standard quantifiers guarded with the dependence predicate:

$$(\exists x \setminus x_1, \dots, x_n). \phi \equiv \exists x. (D(x_1, \dots, x_n, x) \wedge \phi).$$

This yields essentially the same expressive power as IF-logic.

2.3. COMPOSITIONALITY: HODGES' SEMANTICS

But, what does it all mean? Hintikka claimed that **a compositional semantics for IF logic could not be given** [9]. Instead he gave a “Game-Theoretical Semantics”, essentially reduction to Skolem form as above.

Wilfrid Hodges showed that it could [13, 14].⁴

Before giving Hodges' construction, it will be useful firstly to recall Tarski's solution to the problem of how to define the truth of a sentence in a first-order structure $\mathcal{M} = (A, \dots)$ with underlying set A .⁵ In order to do this, he had to deal with the more general case of open formulas. The idea was to define

$$\mathcal{M}, s \models_X \phi$$

where X is a finite set of variables including those occurring free in ϕ , and s is an assignment of elements of A to X .⁶ Typical clauses include:

$$\mathcal{M}, s \models_X \phi \wedge \psi \equiv \mathcal{M}, s \models_X \phi \text{ and } \mathcal{M}, s \models_X \psi$$

$$\mathcal{M}, s \models_X \neg\phi \equiv \mathcal{M}, s \not\models_X \phi$$

$$\mathcal{M}, s \models_X \forall v. \phi \equiv \forall a \in A. \mathcal{M}, s[v \mapsto a] \models_{X \cup \{v\}} \phi$$

$$\mathcal{M}, s \models_X \exists v. \phi \equiv \exists a \in A. \mathcal{M}, s[v \mapsto a] \models_{X \cup \{v\}} \phi$$

Here $s[v \mapsto a]$ is the assignment defined on $X \cup \{v\}$ as follows: $s[v \mapsto a](v) = a$, and $s[v \mapsto a](w) = s(w)$ for $w \neq v$.

This is the very prototype of a compositional semantic definition. Via Dana Scott, this idea led to the now-standard use of **environments** in denotational semantics [28]. Environments are nowadays ubiquitous in all forms of semantics in computer science [34, 19].

2.3.0.3. Teams Hodges' key idea was to see that one must lift the semantics of formulas from single assignments to **sets of assignments**. Notions of dependence of one variable on others are only meaningful among a set of assignments. Hodges called these sets “trumps”; we follow [33] in calling them **teams**.

We consider the semantics of Dependence logic [33]. Formulas are built up from atomic formulas and their negations and the dependence predicates, by conjunction, disjunction, and universal and existential quantification. The set of all individual variables is denoted \mathcal{V} . A **team on $X \subseteq \mathcal{V}$** is a set of Tarski assignments on X . We define the following operations on teams:

- If T is a team on X and $v \in \mathcal{V}$, then $T[v \mapsto A]$ is the team on $X \cup \{v\}$ defined by:

$$T[v \mapsto A] = \{t[v \mapsto a] \mid t \in T \wedge a \in A\}.$$

- If T is a team on X , $v \in \mathcal{V}$, and $f : T \longrightarrow A$, then $T[v \mapsto f]$ is the team on $X \cup \{v\}$ defined by:

$$T[v \mapsto f] = \{t[v \mapsto f(t)] \mid t \in T\}.$$

2.3.0.4. *The Satisfaction Relation* We define a satisfaction relation

$$\mathcal{M}, T \models_X \phi$$

where the free variables of ϕ are contained in X , and T is a team on X . (In practice, we elide \mathcal{M}).

Firstly, for literals L we have:

$$T \models_X L \equiv \forall t \in T. t \models_X L$$

where $t \models_X L$ is the standard Tarskian definition of satisfaction of an atomic formula or its negation in a structure with respect to an assignment.

2.3.0.5. *Connectives and Quantifiers* The clauses for connectives and quantifiers are as follows:

$$T \models_X \phi \wedge \psi \equiv T \models_X \phi \text{ and } T \models_X \psi$$

$$T \models_X \phi \vee \psi \equiv \exists U, V. ([U \models_X \phi \text{ and } V \models_X \psi] \wedge [T = U \cup V])$$

$$T \models_X \forall v. \phi \equiv T[v \mapsto A] \models_{X \cup \{v\}} \phi$$

$$T \models_X \exists v. \phi \equiv \exists f : T \longrightarrow A. T[v \mapsto f] \models_{X \cup \{v\}} \phi.$$

2.3.0.6. *Semantics of the dependence predicate* Given a set of variables X and $W \subseteq X$, we define the following notions:

- An equivalence relation on assignments on X :

$$s \simeq_W t \equiv \forall w \in W. s(w) = t(w).$$

- A function $f : A^X \longrightarrow A$ **depends only on** W , written $f : A^X \longrightarrow_W A$, if for some $g : A^W \longrightarrow A$, $f = g \circ p_{XW}$, where $p_{XW} : A^X \longrightarrow A^W$ is the evident projection. Note that if such a g exists, it is unique.

Now we can define:

$$T \models_X D(W, v) \equiv \forall s, t \in T. s \simeq_W t \Rightarrow s(v) = t(v)$$

Note that this expresses **functional dependence**, exactly as in database theory [1].

An equivalent definition can be given in terms of the dependency condition on functions:

$$T \models_X D(W, v) \equiv \exists f : T \longrightarrow_W A. \forall t \in T. t(v) = f(t).$$

Strictly speaking, this is the “positive part” of the definition as given in [33] following Hodges. There is also a negative part, which defines satisfaction for ϕ as for the positive definition, but with respect to the De Morgan dual ϕ^d of ϕ :

$$(\phi \vee \psi)^d = \phi^d \wedge \psi^d, \quad (\exists v. \phi)^d = \forall v. \phi^d, \quad \text{etc.}$$

This allows for a “game-theoretic negation”, which formally “interchanges the rôles of the players”. It is simpler, and from our perspective loses nothing, to treat this negation as a **defined operation**, and work exclusively with formulas in negation normal form as above.

The theory of dependence logic: metalogical properties, connections with second-order logic, complexity and definability issues, *et cetera*, is extensively developed in [33]. However, as explained in the Introduction, many basic questions remain. We shall now show how the Hodges semantics can be seen in a new light, as arising from a general construction.

3. The Hodges construction revisited

An important clue to the general nature of the construction is contained in the observation by Hodges [13] (and then in [33]) that the sets of teams denoted by formulas of IF-logic or Dependence logic are **downwards closed**: that is, if $T \in \llbracket \phi \rrbracket$ and $S \subseteq T$, then $S \in \llbracket \phi \rrbracket$. This is immediately suggestive of well-known constructions on ordered structures.

3.1. A GENERAL CONSTRUCTION

We recall a couple of definitions. A **commutative ordered monoid** is a structure $(M, +, 0, \leq)$, where (M, \leq) is a partially ordered set, and $(M, +, 0)$ is a commutative monoid (a set with an associative and commutative operation $+$ with unit 0), such that $+$ is monotone:

$$x \leq x' \wedge y \leq y' \Rightarrow x + y \leq x' + y'.$$

A **commutative quantale** is a commutative ordered monoid where the partial order is a complete lattice, and $+$ distributes over all suprema: $m + \bigvee_{i \in I} m_i = \bigvee_{i \in I} (m + m_i)$.

Let $(M, +, 0, \leq)$ be a commutative ordered monoid. Then $\mathcal{L}(M)$, the set of lower (or downwards-closed) sets of M , ordered by inclusion, is the **free commutative quantale** generated by M [20].⁷

A downwards closed subset of a partially ordered set P is a set S such that:

$$x \leq y \in S \Rightarrow x \in S.$$

Thus this notion generalizes the downwards closure condition on sets of teams, where we think of $\mathcal{P}(A^X)$, the set of all teams on a set of variables X , as the commutative ordered monoid $(\mathcal{P}(A^X), \cup, \emptyset, \subseteq)$. The following notation will be useful. Given $X \subseteq P$, where P is a partially ordered set, we define

$$\downarrow(X) = \{x \in P \mid \exists y \in X. x \leq y\},$$

the **downwards closure** of X . A set S is downwards closed if and only if $S = \downarrow(S)$.

As a commutative quantale, $\mathcal{L}(M)$ is a model of intuitionistic linear logic (phase semantics [35, 27, 6]).⁸ In particular, we have

$$\begin{aligned} A \otimes B &= \downarrow\{m + n \mid m \in A \wedge n \in B\} \\ A \multimap B &= \{m \mid \forall n. n \in A \Rightarrow m + n \in B\} \end{aligned}$$

We note that when the definition of \otimes , the multiplicative **conjunction**, is specialized to our concrete setting, it yields the definition of **disjunction** in the Hodges semantics!

The multiplicative implication \multimap has not been considered previously in the setting of IF-logic and Dependence logic. However, it is perfectly well defined, and is in fact uniquely specified as the adjoint of the linear conjunction:

$$A \otimes B \leq C \iff A \leq B \multimap C.$$

Note that linear implication automatically preserves downwards closure.

3.2. WHAT IS THE PROPOSITIONAL LOGIC OF DEPENDENCE?

In fact, $\mathcal{L}(M)$ carries a great deal of structure. Not only is it a commutative quantale (and hence carries an interpretation of linear logic), but it is also a **complete Heyting algebra**, and hence carries an interpretation of intuitionistic logic.

We have the clauses

$$\begin{aligned} m \models A \wedge B &\equiv m \models A \text{ and } m \models B \\ m \models A \vee B &\equiv m \models A \text{ or } m \models B \\ m \models A \rightarrow B &\equiv \forall n \leq m. \text{ if } n \models A \text{ then } n \models B \end{aligned}$$

The situation where we have both intuitionistic logic and multiplicative linear logic coexisting is the setting for **BI logic**, the “logic of Bunched Implications” of David Pym and Peter O’Hearn [21, 24], which forms the basis for **Separation logic** (Reynolds and O’Hearn) [26], an increasingly influential logic for verification. The construction $\mathcal{L}(M)$ is exactly the way a “forcing semantics” for BI-logic is converted into an algebraic semantics as a “BI-algebra”, *i.e.* a structure which is both a commutative quantale **and** a complete Heyting algebra [25]. $\mathcal{L}(M)$ is in fact the free construction of a complete BI-algebra over an ordered commutative monoid.

This provides one reason for proposing BI-logic as the right answer to the question posed at the beginning of this subsection. The compelling further evidence for this claim will come from the natural rôle played by the novel connectives we are introducing into the logic of dependence. This rôle will become apparent in the subsequent developments in this paper.

3.3. BID-LOGIC AND ITS TEAM SEMANTICS

We shall spell out the extended logical language we are led to consider, and its concrete team semantics, extending the Hodges-style semantics already given in section 2.

We call the extended language BID, for want of a better name. Formulas are built from atomic formulas and their negations, and dependence formulas, by the standard first-order quantifiers, and the following propositional connectives: the intuitionistic (or “additive”) connectives \wedge , \vee , \rightarrow , and the multiplicative connectives \otimes and \multimap .

3.3.0.7. *Team Semantics for BI Logic* The team semantics for BID-logic is as follows:

$$\begin{aligned} T \models A \wedge B &\equiv T \models A \text{ and } T \models B \\ T \models A \vee B &\equiv T \models A \text{ or } T \models B \\ T \models A \rightarrow B &\equiv \forall U \subseteq T. \text{ if } U \models A \text{ then } U \models B \\ T \models A \otimes B &\equiv \exists U, V. T = U \cup V \wedge U \models A \wedge V \models B \\ T \models A \multimap B &\equiv \forall U. [U \models A \Rightarrow T \cup U \models B] \end{aligned}$$

The clauses for atomic formulas and their negations and for the dependence formulas and quantifiers are as given in section 2.

As already noted, the semantics of \wedge and \otimes coincide with those given for conjunction and disjunction in section 2. The connectives \vee and \rightarrow , intuitionistic or additive disjunction and implication, and the multiplicative implication \multimap , are new as compared to IF-logic or Dependence logic.

3.4. THE SEMANTICS OF SENTENCES

It is worth spelling out the semantics of sentences explicitly. By definition, sentences have no free variables, and there is only one assignment on the empty set of variables, which we can think of as the empty tuple $\langle \rangle$. In the Tarski semantics, there are only two possibilities for the set of satisfying assignments of a sentence, \emptyset and $\{\langle \rangle\}$, which we can identify with **false** and **true** respectively. When we pass to the team semantics for BID-logic, there are three possibilities for down-closed set of teams to be assigned to sentences: \emptyset , $\{\emptyset\}$, or $\{\emptyset, \{\langle \rangle\}\}$. Thus the semantics of sentences is **trivalent** in general.

In his papers, Hodges works only with non-empty teams, and has bivalent semantics for sentences. However, there is no real conflict between his semantics and ours. Let BID^- be BID-logic without the linear implication. Note that BID^- properly contains Dependence logic, which is expressively equivalent to IF-logic [33].

Proposition 1 *Every formula in BID^- -logic is satisfied by the empty team; hence in particular every sentence of BID^- -logic has either $\{\emptyset\}$ or $\{\emptyset, \{\langle \rangle\}\}$ as its set of satisfying teams, and the semantics of sentences in BID^- -logic is bivalent.*

Proof A straightforward induction on formulas of BID^- -logic. \square

On the other hand, linear implication clearly violates this property. Note that the empty team satisfies $A \multimap B$ if and only if $A \subseteq B$. We obtain as an immediate corollary:

Proposition 2 *Linear implication is not definable in BID^- -logic, and a fortiori is not definable in Dependence logic or IF-logic.*

3.5. THE GENERAL HODGES CONSTRUCTION

We shall briefly sketch, for the reader conversant with categorical logic, the general form of the construction.

The standard Tarski semantics of first-order logic is a special case of Lawvere's notion of hyperdoctrine [16]. We refer to [22] for a lucid

expository account. Construing \mathcal{L} as a functor in the appropriate fashion, we can give a general form of the Hodges construction as a functor from classical hyperdoctrines to BI-hyperdoctrines [2]. Given a classical hyperdoctrine $\mathbf{P} : \mathcal{C}^{\text{op}} \longrightarrow \mathbf{Pos}$, we define a BI-hyperdoctrine $\mathcal{H}(\mathbf{P})$ on the same base category by composition with the functor \mathcal{L} :

$$\mathcal{H}(\mathbf{P}) = \mathcal{L} \circ \mathbf{P} : \mathcal{C}^{\text{op}} \longrightarrow \mathbf{Pos}.$$

Note that \mathbf{Pos} is an order-enriched category, and \mathcal{L} is an order-enriched functor, so it preserves adjoints, and hence in particular preserves the interpretations of the quantifiers. This observation is spelled out in more detail in Proposition 4.

This exactly generalizes the concrete Hodges construction, which is obtained by applying \mathcal{H} to the standard Tarski hyperdoctrine.

A full account will be given elsewhere.

4. Quantifiers are adjoints in the Hodges construction

We recall the team semantics for the quantifiers.

$$\begin{aligned} T \models_X \forall v. \phi &\equiv T[v \mapsto A] \models_{X \cup \{v\}} \phi \\ T \models_X \exists v. \phi &\equiv \exists f : T \longrightarrow A. T[v \mapsto f] \models_{X \cup \{v\}} \phi. \end{aligned}$$

We may wonder what underlying principles dictate these definitions.

To answer this question, we firstly recall the fundamental insight due to Lawvere [16] that **quantifiers are adjoints to substitution**.⁹

4.1. QUANTIFIERS AS ADJOINTS

Consider a function $f : X \rightarrow Y$. This induces a function

$$f^{-1} : \mathcal{P}(Y) \longrightarrow \mathcal{P}(X) :: T \mapsto \{x \in X \mid f(x) \in T\}.$$

This function f^{-1} has both a left adjoint $\exists(f) : \mathcal{P}(X) \longrightarrow \mathcal{P}(Y)$, and a right adjoint $\forall(f) : \mathcal{P}(X) \longrightarrow \mathcal{P}(Y)$. These adjoints are uniquely specified by the following conditions. For all $S \subseteq X$, $T \subseteq Y$:

$$\exists(f)(S) \subseteq T \iff S \subseteq f^{-1}(T), \quad f^{-1}(T) \subseteq S \iff T \subseteq \forall(f)(S).$$

The unique functions satisfying these conditions can be defined explicitly as follows:

$$\begin{aligned} \exists(f)(S) &:= \{y \in Y \mid \exists x \in X. f(x) = y \wedge x \in S\}, \\ \forall(f)(S) &:= \{y \in Y \mid \forall x \in X. f(x) = y \Rightarrow x \in S\}. \end{aligned}$$

Given a formula ϕ with free variables in $\{v_1, \dots, v_{n+1}\}$, it will receive its Tarskian denotation $\llbracket \phi \rrbracket$ in $\mathcal{P}(A^{n+1})$ as the set of satisfying assignments:

$$\llbracket \phi \rrbracket = \{s \in A^{n+1} \mid s \models_X \phi\}.$$

We have a projection function

$$\pi : A^{n+1} \longrightarrow A^n \ :: (a_1, \dots, a_{n+1}) \mapsto (a_1, \dots, a_n).$$

Note that this projection is the Tarskian denotation of the tuple of terms (v_1, \dots, v_n) . We can characterize the standard quantifiers as **adjoints to this projection**:

$$\llbracket \forall v_{n+1}. \phi \rrbracket = \forall(\pi)(\llbracket \phi \rrbracket), \quad \llbracket \exists v_{n+1}. \phi \rrbracket = \exists(\pi)(\llbracket \phi \rrbracket).$$

If we unpack the adjunction conditions for the universal quantifier, they yield the following bidirectional inference rule:

$$\frac{\Gamma \vdash_X \phi}{\Gamma \vdash_X \forall v_{n+1}. \phi} \quad X = \{v_1, \dots, v_n\}.$$

Here the set X keeps track of the free variables in the assumptions Γ . Note that the usual “eigenvariable condition” is automatically taken care of in this way.

Since adjoints are uniquely determined, this characterization completely captures the meaning of the quantifiers.

4.2. QUANTIFIERS IN THE HODGES SEMANTICS

We shall now verify that the definitions of the quantifiers in the Hodges semantics **are exactly the images under \mathcal{L} of their standard interpretations in the Tarski semantics**, and hence in particular that they are adjoints to substitution. Thus these definitions are **forced**.

It will be convenient to work with the semantic view of quantifiers, as operators on subsets. Consider formulas with free variables in $\{v_1, \dots, v_{n+1}\}$. The Tarski semantics over a structure $\mathcal{M} = (A, \dots)$ assigns such formulas values in $\mathcal{P}(A^{n+1})$. We can regard the quantifiers $\exists v_{n+1}, \forall v_{n+1}$ as functions

$$\exists(\pi), \forall(\pi) : \mathcal{P}(A^{n+1}) \longrightarrow \mathcal{P}(A^n)$$

$$\exists(\pi)(S) = \{s \in A^n \mid \exists a \in A. s[v_{n+1} \mapsto a] \in S\}$$

$$\forall(\pi)(S) = \{s \in A^n \mid \forall a \in A. s[v_{n+1} \mapsto a] \in S\}$$

For any m , we define $\mathcal{H}(A^m) = \mathcal{L}(\mathcal{P}(A^m))$. Thus $\mathcal{H}(A^m)$ is the set of downwards closed sets of teams on the variables $\{v_1, \dots, v_m\}$. This provides the corresponding “space” of semantic values for formulas in the

Hodges semantics. The interpretation of quantifiers in that semantics is given by the following set operators:

$$\exists_H, \forall_H : \mathcal{H}(A^{n+1}) \longrightarrow \mathcal{H}(A^n)$$

$$\exists_H(\mathbf{U}) = \{T \in \mathcal{P}(A^n) \mid \exists f : T \rightarrow A. T[v_{n+1} \mapsto f] \in \mathbf{U}\}$$

$$\forall_H(\mathbf{U}) = \{T \in \mathcal{P}(A^n) \mid T[v_{n+1} \mapsto A] \in \mathbf{U}\}$$

We extend the definition of \mathcal{L} to act on functions¹⁰ $h : \mathcal{P}(Y) \longrightarrow \mathcal{P}(X)$:

$$\mathcal{L}(h) : \mathcal{H}(Y) \longrightarrow \mathcal{H}(X) :: \mathbf{U} \mapsto \downarrow\{h(T) \mid T \in \mathbf{U}\}.$$

In the case that $h = f^{-1}$, where $f : X \longrightarrow Y$, we write $\mathcal{L}(h) = \mathcal{H}(f)$.

Proposition 3 *The Hodges quantifiers are the image under \mathcal{L} of the Tarski quantifiers:*

$$\exists_H = \mathcal{L}(\exists(\pi)), \quad \forall_H = \mathcal{L}(\forall(\pi)).$$

Proof Firstly, we show that $\mathcal{L}(\exists(\pi))(\mathbf{U}) \subseteq \exists_H(\mathbf{U})$ for all $\mathbf{U} \in \mathcal{H}(A^{n+1})$. Suppose that $T \in \mathbf{U}$. Let $T' = \exists(\pi)(T)$. This means that

$$\forall t \in T'. \exists a \in A. t[v_{n+1} \mapsto a] \in T.$$

Using the axiom of choice, there exists a function $f : T' \longrightarrow A$ such that

$$T'[v_{n+1} \mapsto f] \subseteq T \in \mathbf{U}.$$

Since \mathbf{U} is downwards closed, this implies that $T' \in \exists_H(\mathbf{U})$, as required.

The converse follows immediately from the fact that $\exists(\pi)(T[v_{n+1} \mapsto f]) = T$.

Next we show that $\mathcal{L}(\forall(\pi))(\mathbf{U}) \subseteq \forall_H(\mathbf{U})$. Since $(\forall(\pi)(T))[v_{n+1} \mapsto A] \subseteq T$, if $T \in \mathbf{U}$, then $\forall(\pi)(T) \in \forall_H(\mathbf{U})$ by downwards closure. The converse follows similarly from $T \subseteq \forall(\pi)(T[v_{n+1} \mapsto A])$. \square

Proposition 4 *The Hodges quantifiers are adjoints to substitution:*

1. \exists_H is left adjoint to $\mathcal{H}(\pi)$:

$$\exists_H(\mathbf{U}) \subseteq \mathbf{V} \iff \mathbf{U} \subseteq \mathcal{H}(\pi)(\mathbf{V}).$$

2. \forall_H is right adjoint to $\mathcal{H}(\pi)$:

$$\mathcal{H}(\pi)(\mathbf{V}) \subseteq \mathbf{U} \iff \mathbf{V} \subseteq \forall_H(\mathbf{U}).$$

Proof It is straightforward to verify the adjunction conditions directly. We give a more conceptual argument. There is a natural point-wise ordering on monotone functions between partially ordered sets, $h, k : P \rightarrow Q$:

$$h \leq k \equiv \forall x \in P. h(x) \leq k(x).$$

\mathcal{L} is an **order-enriched functor** with respect to this ordering. Functoriality means that

$$\mathcal{L}(h \circ g) = \mathcal{L}(h) \circ \mathcal{L}(g), \quad \mathcal{L}(\text{id}_M) = \text{id}_{\mathcal{L}(M)},$$

while order-enrichment means that

$$h \leq k \Rightarrow \mathcal{L}(h) \leq \mathcal{L}(k).$$

These properties imply that \mathcal{L} automatically preserves adjointness. That is, if we are given monotone maps

$$f : P \rightarrow Q, \quad g : Q \rightarrow P$$

such that $\text{id}_P \leq g \circ f$ and $f \circ g \leq \text{id}_Q$, *i.e.* so that f is left adjoint to g , then

$$\text{id}_{\mathcal{L}(P)} = \mathcal{L}(\text{id}_P) \leq \mathcal{L}(g \circ f) = \mathcal{L}(g) \circ \mathcal{L}(f),$$

and similarly $\mathcal{L}(f) \circ \mathcal{L}(g) \leq \text{id}_{\mathcal{L}(Q)}$, so $\mathcal{L}(f)$ is left adjoint to $\mathcal{L}(g)$ (and of course $\mathcal{L}(g)$ is right adjoint to $\mathcal{L}(f)$). Combining this with Proposition 3 yields the required result. \square

4.3. THE DEPENDENCE-FRIENDLY QUANTIFIERS

We shall also give characterizations of the dependence-guarded quantifiers as certain adjoints: this will be our first use of the intuitionistic implication.

We recall the definition of the dependence-friendly existential quantifier:

$$(\exists x \setminus x_1, \dots, x_n). \phi \equiv \exists x. (D(x_1, \dots, x_n, x) \wedge \phi).$$

There has not been a comparably natural notion of dependence-friendly universal quantification. According to our analysis, this is because the appropriate connective needed to express the right notion, namely intuitionistic implication, has not been available. Using it, we can define such a quantifier:

$$(\forall x \setminus x_1, \dots, x_n). \phi \equiv \forall x. (D(x_1, \dots, x_n, x) \rightarrow \phi).$$

As evidence for the naturalness of these quantifiers, we shall express them both as adjoints.

Firstly, we recall that intuitionistic conjunction and implication are related by another fundamental adjointness [16]:

$$\mathbf{U} \cap \mathbf{V} \subseteq \mathbf{W} \iff \mathbf{U} \subseteq \mathbf{V} \rightarrow \mathbf{W}. \quad (1)$$

This can be expressed as a bidirectional inference rule:

$$\frac{\phi \wedge \psi \vdash \theta}{\phi \vdash \psi \rightarrow \theta}.$$

Next, we extend our semantic notation to the dependence-friendly quantifiers. Given $W \subseteq \{v_1, \dots, v_n\}$, we define $D_W \in \mathcal{H}(A^{n+1})$:

$$D_W = \{T \mid \forall s, t \in T. s \simeq_W t \Rightarrow s(v_{n+1}) = t(v_{n+1})\}.$$

Now we can define the semantic operators corresponding to the dependence-friendly quantifiers:

$$\begin{aligned} \exists_W, \forall_W &: \mathcal{H}(A^{n+1}) \longrightarrow \mathcal{H}(A^n) \\ \exists_W(\mathbf{U}) &= \exists_H(D_W \cap \mathbf{U}) \\ \forall_W(\mathbf{U}) &= \forall_H(D_W \rightarrow \mathbf{U}) \end{aligned}$$

Proposition 5 *The dependence-friendly existential \exists_W is left adjoint to the following operation:*

$$\mathbf{V} \mapsto (D_W \rightarrow \mathcal{H}(\pi)(\mathbf{V})).$$

The dependence-friendly universal \forall_W is right adjoint to the following operation:

$$\mathbf{V} \mapsto (D_W \cap \mathcal{H}(\pi)(\mathbf{V})).$$

Proof A direct verification is straightforward, but it suffices to observe that adjoints compose, and then to use Proposition 4 and the adjointness (1). \square

Of course, the analysis we have given in this sub-section applies to any guarded quantifiers; the dependence predicates play no special rôle here. The point is to show how the intuitionistic connectives round out the logic in a natural fashion. We shall apply them to a finer analysis of dependence itself in section 6.

5. Full Abstraction

We shall now prove a full abstraction result in the sense of Hodges [13].¹¹ The point of this is to show that, even if we take sentences and their truth-values as primary, the information contained in the semantics of formulas in general is not redundant, since whenever two formulas receive different denotations, they make different contributions overall to the truth-values assigned to sentences.

The fact that such a result holds for BID^- -logic is notable, in that the logic is highly non-classical, while the semantics of sentences is bivalent. For BID -logic, the set of possible truth values for open formulas is huge even in finite models [3], while the semantics of sentences is trivalent.

While our argument follows that of Hodges [13], we find a natural rôle for the intuitionistic implication, and can give a very simple proof, while Hodges' argument goes through the correspondence with the game-theoretical semantics.

To formalize full abstraction, we introduce the notion of a **sentential context** with respect to a set of variables X . This is a formula with an occurrence of a “hole” $[\cdot]$ such that inserting a formula with free variables in X into the hole yields a sentence. Now consider two formulas ϕ and ψ of BID -logic, with free variables in X . We say that the formulas are **semantically equivalent** if they have the same denotations, *i.e.* the same sets of satisfying teams, in all interpretations with respect to all structures. We say that ϕ and ψ are **observationally equivalent** if for all sentential contexts $C[\cdot]$ for X , $C[\phi]$ and $C[\psi]$ are assigned the same truth values in all interpretations. The fact that semantic equivalence implies observational equivalence follows immediately from the compositional form of the semantics. The converse is **full abstraction**.¹²

Proposition 6 *The team semantics is fully abstract for any sublanguage of BID -logic containing universal quantification and intuitionistic implication.*

Proof Suppose that $\llbracket \phi \rrbracket \setminus \llbracket \psi \rrbracket$ in some interpretation contains a team T . Extend the language with a relation symbol R , and the interpretation by assigning $\downarrow(T)$ to R . Then use the context

$$C[\cdot] \equiv \forall v_1, \dots, \forall v_n. (R(v_1, \dots, v_n) \rightarrow [\cdot]),$$

where the free variables in ϕ and ψ are contained in $\{v_1, \dots, v_n\}$. Then $C[\phi]$ is true (satisfied by the empty tuple), since for every team T' satisfying $R(v_1, \dots, v_n)$, $T' \subseteq T$, and hence by assumption and downwards closure, T' satisfies ϕ . This means that all teams over $\{v_1, \dots, v_n\}$

satisfy the implication $R(v_1, \dots, v_n) \rightarrow \phi$, and hence $\langle \rangle$ satisfies $C[\phi]$. On the other hand, $C[\psi]$ is not satisfied by the empty tuple, since T satisfies $R(v_1, \dots, v_n)$, while T does not satisfy ψ by assumption. \square Note that the use of the intuitionistic implication in relativizing to those teams satisfying the precondition $R(v_1, \dots, v_n)$ is exactly what is needed.

6. Analyzing Dependence

We now turn to the dependence predicate itself. Since it encapsulates the “jump” from first-order to second-order semantics, we cannot be too hopeful about taming it axiomatically¹³. But it turns out that we can give a finer analysis in BID-logic.

Consider the following “trivial” case of dependence:

$$C(v) \equiv D(\emptyset, v).$$

This expresses that v depends on nothing at all, and hence has a fixed value — functional dependency for the constant function. Semantically, this is the following simple special case of the semantics of dependence:

$$T \models_X C(v) \equiv \forall t_1, t_2 \in T. t_1(v) = t_2(v).$$

Using the intuitionistic implication, we can **define** the general dependence predicate from this special case:

$$D(W, v) := \bigwedge_{w \in W} C(w) \rightarrow C(v)$$

Proposition 7 *The definition of D from C is semantically equivalent to the definition given previously:*

$$T \models_X D(W, v) \equiv \forall s, t \in T. s \simeq_W t \Rightarrow s(v) = t(v).$$

Proof This is just an exercise in unwinding the definitions. Note that the intuitionistic implication lets us range over all subsets of the team which are in a single equivalence class under \simeq_W , and require that v is constant on those subsets. \square

6.1. ARMSTRONG AXIOMS

The current stock of plausible axioms for the dependence predicates is limited to the **Armstrong axioms** from database theory [1]. These are a standard complete set of axioms for functional dependence. They can be given as follows.

- (1) Always $D(x, x)$.
- (2) If $D(x, y, z)$, then $D(y, x, z)$.
- (3) If $D(x, x, y)$, then $D(x, y)$.
- (4) If $D(x, z)$, then $D(x, y, z)$.
- (5) If $D(x, y)$ and $D(y, z)$, then $D(x, z)$.

However, in the light of our analysis, the Armstrong axioms simply fall out as standard properties of implication and conjunction. In fact, they correspond to the well-known axioms **I**, **C**, **W**, **K**, **B** respectively [4] — which form a complete axiomatization of **intuitionistic** (but not classical!) implication.¹⁴ For example, (2) corresponds to

$$(p \rightarrow q \rightarrow r) \rightarrow (q \rightarrow p \rightarrow r).$$

where we set $p = C(x)$, $q = C(y)$, $r = C(z)$.

Thus we have reduced the understanding of the dependence predicate to understanding of the, *prima facie* simpler, constancy predicate C .

7. Further Directions

In this final section, we shall sketch a number of further directions. Detailed accounts are under development, and will appear elsewhere.

7.1. COMPLETENESS

Predicate BI-logic is a well developed formalism, with a proof theory which is sound and complete relative to an algebraic semantics [24]. Since BID-logic is a special case, we have a sound ambient inference system. Of course this is not complete for the intended semantics for BID-logic — and cannot be. We may hope to obtain completeness for some smaller class of models, possibly on the lines of the Henkin completeness theorem for higher-order logic [7].

7.2. DIAGRAMS

Now fix a particular interpretation in a structure \mathcal{M} with universe A . Consider the following construction. We introduce constants for each

$a \in A$, the usual first order diagram (all true atomic sentences), and the following infinitary axiom:

$$\forall v. \bigotimes_{a \in A} (v = a).$$

We can **define** the predicate C (and hence dependence D) by the following infinitary formula:

$$C(v) := \bigvee_{a \in A} (v = a).$$

Note how the two different connectives (one additive, the other multiplicative) feature naturally.

This gives a logical (albeit infinitary) characterization of dependence.

7.3. REPRESENTATION

We can also consider representation theory for the structures $\mathcal{H}(X) = \mathcal{L}(\mathcal{P}(X))$. We seek lattice-theoretic properties of these structures which suffice to characterize them.

Firstly, we note that the down-closures of single teams are exactly the **complete join-primes** of the lattice:

$$a \leq \bigvee_i b_i \Rightarrow \exists i. a \leq b_i.$$

Moreover, these join-primes order generate, *i.e.* every element is the join of the join-primes below it. All of this structure is in terms of the intuitionistic disjunction.

Next, we note that \otimes is closed and idempotent on the join-primes, endowing them with the structure of a semilattice. This is very different to the semilattice structure given by intuitionistic disjunction: e.g.

$$\downarrow(T_1) \vee \downarrow(T_2) = \downarrow(\{T_1, T_2\}) \neq \downarrow(T_1 \cup T_2) = \downarrow(T_1) \otimes \downarrow(T_2).$$

The **double singletons** are exactly the complete atoms in this semilattice, which is complete atomic in the usual sense.

Syntactically, assuming names for elements, we can describe these atomic join-primes in the lattice of propositions over variables v_1, \dots, v_n as

$$(v_1 = a_1) \wedge \dots \wedge (v_n = a_n).$$

These are of course the tuples. (Downclosures of) arbitrary teams are then described by expressions $\bigotimes_i A_i$, where A_i ranges over such atoms.

Arbitrary elements are joins (intuitionistic disjunctions) of such elements. So there is a normal form for general elements:

$$\bigvee_i \bigotimes_{ij} A_{ij}.$$

Moreover, from the lattice-theoretic properties it is easily shown that the ordering between such normal forms agrees with the set inclusion ordering.

7.4. EXPRESSIVENESS

One of the defining characteristics of Dependence Logic as well as IF-logic is that they can be expressed in Existential Second Order Logic, Σ_1^1 , and conversely, every Σ_1^1 definable property of structures can be expressed with a sentence of Dependence Logic. Both are true even on finite structures. To see what this connection with Σ_1^1 means let us adopt the notation that if T is a team on a set X of variables, then $\text{rel}(T)$ is the corresponding relation. Hodges [14] associates with every formula ϕ of IF-logic (equivalently, of Dependence Logic) with free variables in the set $X = \{x_1, \dots, x_n\}$ an Existential Second Order sentence $\tau_\phi(R)$, with R an n -ary predicate symbol, such that in any model \mathcal{M} and for any team T on X the following holds:

$$\mathcal{M}, T \models_X \phi \iff (\mathcal{M}, \text{rel}(T)) \models \tau_\phi(R). \quad (2)$$

Conversely, if Φ is any Existential Second Order sentence, then there is a sentence ϕ of Dependence Logic such that the following holds for all models \mathcal{M} :

$$\mathcal{M} \models \Phi \iff \mathcal{M}, \{\langle \rangle\} \models \phi.$$

Virtually all model theoretic properties of Dependence Logic follow from this relationship with Σ_1^1 , for example, the Compactness Theorem, the downward and upward Löwenheim-Skolem Theorems, the Interpolation Theorem, and the fact that every sentence ϕ in Dependence Logic for which there exists a “negation” ψ such that for all \mathcal{M}

$$\mathcal{M} \models \phi \iff \mathcal{M} \not\models \psi,$$

is actually first order definable¹⁵. Also the interesting fact that the class of properties of finite structures expressible in Dependence Logic is exactly NP follows from this. Because of these connections it is quite interesting to ask whether the extensions BID^- and BID can likewise be embedded in Σ_1^1 , the existential fragment of Second Order Logic.

Now the question arises which semantics one should use. To be able to compare results with Dependence Logic and IF-logic, we use the full semantics familiar from [13] and [33].

Proposition 8 *There is no translation of BID^- , and hence none of BID , into existential second order Σ_1^1 . The same is true on finite models, assuming $NP \neq co-NP$.*

Proof Let $\phi(x_1)$ be a formula of Dependence Logic in the empty vocabulary such that for any team $T: \mathcal{M}, T \models \phi(x_1)$ if and only if A is infinite¹⁶. Let \perp denote a sentence in the empty vocabulary, only satisfied by the empty team, e.g. $\forall x.x = x \wedge \neg x = x$. Suppose there were an Existential Second Order sentence $\tau(R)$ such that a model \mathcal{M} and a team T on $\{x_1\}$ satisfy $\phi(x_1) \rightarrow \perp$ if and only if $(\mathcal{M}, \text{rel}(T))$ satisfies $\tau(R)$. If \mathcal{M} is any finite model and $T = \{s\}$, where $s(x_1) \in M$, then $\mathcal{M}, T \models \phi(x_1) \rightarrow \perp$, whence $(\mathcal{M}, \{s(x_1)\}) \models \tau(R) \wedge \exists x_1 R(x_1)$. By the Compactness Theorem of Existential Second Order Logic, $\tau(R) \wedge \exists x_1 R(x_1)$ has an infinite model $(\mathcal{M}', \text{rel}(T'))$. Thus \mathcal{M}' and the team T' satisfy $\phi(x_1) \rightarrow \perp$. Moreover, $T' \neq \emptyset$. By the definition of the semantics of \rightarrow , since T' satisfies $\phi(x_1)$ in \mathcal{M}' , T' must satisfy \perp , a contradiction.

Let us then consider finite models. It is easy to write down a formula $\phi(x_1)$ of Dependence Logic in the vocabulary of graphs such that for any team $T \neq \emptyset: \mathcal{M}, T \models \phi(x_1)$ if and only if \mathcal{M} is 3-colorable. Let \perp be as above. If \mathcal{M} is any graph that is not 3-colorable and $T = \{s\}$, where $s(x_1) \in M$, then $\mathcal{M}, T \models \phi(x_1) \rightarrow \perp$. On the other hand, suppose \mathcal{M} is 3-colorable, but \mathcal{M} and some team $\{s\}$ satisfy $\phi(x_1) \rightarrow \perp$. By the definition of the semantics of \rightarrow , since $\{s\}$ satisfies $\phi(x_1)$ in \mathcal{M} , $\{s\}$ must satisfy \perp , a contradiction. Thus a graph \mathcal{M} and a team $\{s\}$ satisfy $\phi(x_1) \rightarrow \perp$ if and only if \mathcal{M} is not 3-colorable. Suppose now there were an Existential Second Order sentence $\tau(R)$ such that a graph \mathcal{M} and a team T satisfy $\phi(x_1) \rightarrow \perp$ if and only if $(\mathcal{M}, \text{rel}(T))$ satisfies $\tau(R)$. Then we could check if a graph \mathcal{M} is not 3-colorable by checking if $\tau(R)$ is satisfied by \mathcal{M} and a team $\{s\}$, where s can be any assignment. The latter is NP, so we get $NP = co-NP$. \square

The same argument can be used to show that \multimap leads outside of Σ_1^1 : Suppose $\phi(x_1)$ is as above and there is an Existential Second Order sentence $\tau(R)$ such that a model \mathcal{M} and a team T satisfy $\perp \wedge \phi(x_1) \multimap \perp$ if and only if $(\mathcal{M}, \text{rel}(T))$ satisfies $\tau(R)$. If \mathcal{M} is any finite model and $T = \emptyset$, then $\mathcal{M}, T \models \perp \wedge \phi \multimap \perp$, whence $(\mathcal{M}, \emptyset) \models \tau(R)$. By the Compactness Theorem of Existential Second Order Logic, $\tau(R)$ has an infinite model $(\mathcal{M}', \text{rel}(T'))$. Thus \mathcal{M}' and the team T' satisfy $\perp \wedge \phi \multimap \perp$. In particular, $T' = \emptyset$ and \emptyset satisfies $\phi \multimap \perp$. Since in this model any $\{s\}$ satisfies ϕ , by the definition of the semantics of \multimap , $\{s\}$ satisfies \perp , a contradiction.

The proof actually shows that BID^- fails to satisfy the Compactness Theorem. A similar argument shows that BID^- fails to satisfy the Downward Löwenheim Skolem Theorem.

Proposition 9 *There is a translation of BID into Full Second Order Logic.*

Proof We follow [14] (see also [33]) and present only the additions needed over and above Dependence Logic and IF-logic:

$$\begin{aligned}\tau_{\phi \dashv\vdash \psi}(R) &= \forall S(\tau_{\phi}(S) \rightarrow \forall U(\forall \vec{x}(U(\vec{x}) \leftrightarrow (S(\vec{x}) \vee R(\vec{x}))) \rightarrow \tau_{\psi}(U))) \\ \tau_{\phi \rightarrow \psi}(R) &= \forall S(\forall \vec{x}(S(\vec{x}) \rightarrow R(\vec{x})) \rightarrow (\tau_{\phi}(S) \rightarrow \tau_{\psi}(S)))\end{aligned}$$

□

In conclusion we may say that BID^- and BID seem to have a more robust and uniform algebraic structure than Dependence Logic and IF-logic. We anticipate that this is reflected also in an effective proof theory, still to be developed. On the other hand the price of this seems to be that “nice” model theoretic properties are lost at least in the full semantics. Perhaps there are some underlying, hitherto unidentified, reasons why logics developed for dependence cannot have at the same time “nice” model theory and effective proof theory. After all, we know from Lindström’s Theorem ([17]) that there are intrinsic blocks in having model theoretically defined extensions of first order logic with both nice proof theory and nice model theory. The difference here is that we have a trivalent logic, something different from the approach of Lindström. So it is too early to say whether there are general reasons why BID does not satisfy Compactness and other model theoretic properties familiar from Dependence Logic, or whether we have just not hit the right concepts for such yet.

Notes

¹ See e.g. online resources such as <http://library.wolfram.com/infocenter/MathSource/4734/>.

² This can be described as the fragment comprising formulas $\exists f_1 \dots \exists f_n. \phi$, where the f_i are function variables, and ϕ is a first-order formula over a signature extended by these function variables.

³ In [33] the notation $\models(x_1, \dots, x_n, x)$ for $D(x_1, \dots, x_n, x)$ is used.

⁴ Hintikka has apparently not conceded the point [10], although there is no argument as to the mathematical content of Hodges’ results. As far as we are concerned, Hodges’ semantics meets all the criteria for a compositional semantics, and is moreover fully abstract. Our concern here is to understand it better, as an interesting construction in its own right.

⁵ The classic reference is [29], but in fact the modern model-theoretic definition first appeared in [30], as pointed out in Wilfrid Hodges’ article on “Tarski’s Truth Definitions” in the Stanford Encyclopedia of Philosophy, available online at <http://plato.stanford.edu/entries/tarski-truth/>, which gives an informative overview.

⁶ Explicitly, an assignment is simply a function $s : X \rightarrow A$. We write A^X for the set of all such assignments. Older tradition was to define satisfaction relative to assignments to **all** variables, which were typically arrayed in infinite sequences. More recently, it has been understood, under the influence of categorical logic, that to reveal the salient structure one should give the definition relative to a finite environment that grows as quantifiers are stripped off in the recursive definition.

⁷ More precisely, it is the left adjoint to the evident forgetful functor.

⁸ It is also an instance of Urquhart’s semilattice semantics for relevance logic [31]. Mitchell and Simmons observe in [20] that in the case (such as ours) where the monoid is a boolean algebra, $\mathcal{L}(M)$ is actually a model of **classical linear logic**. This does not seem apposite to our purposes here.

⁹ See [5] for an introduction to adjunctions on posets.

¹⁰ More precisely, homomorphisms of the appropriate kind. The reader familiar with category theory will see that we are really specifying the functorial action of \mathcal{L} in a particular case.

¹¹ As Hodges notes, he himself takes the term, and the concept, from Computer Science [18, 23].

¹² While this notion is perfectly consistent with usage in Computer Science, one very important tensioning ingredient in the programming language context is missing, namely correspondence with an independently defined operational semantics [18, 23].

¹³ See [32] for details on this.

¹⁴ Under the (Curry part of the) Curry-Howard correspondence, they correspond to a well-known functionally complete set of **combinators** [4].

¹⁵ See e.g. [33] for details.

¹⁶ The free variable x_1 plays no role in this.

References

1. Armstrong, W. W.: 1974, ‘Dependency structures of data base relationships’. In: *Information Processing 74. Proc. IFIP Congress*. pp. 580–583.
2. B. Biering, L. Birkedal, N. T.-S.: 2007, ‘BI-hyperdoctrines, higher-order separation logic, and abstraction’. *ACM Transactions on Programming Languages and Systems* **29**(5).
3. Cameron, P. and W. Hodges: 2001, ‘Some combinatorics of imperfect information’. *J. Symbolic Logic* **66**(2), 673–684.
4. Curry, H. B. and R. Feys: 1958, *Combinatory Logic Volume 1*, Studies in Logic and the Foundations of Mathematics. North Holland.
5. Davey, B. A. and H. A. Priestley: 2002, *Introduction to Lattices and Order*. Cambridge University Press, second edition.
6. Girard, J.-Y.: 1987, ‘Linear Logic’. *Theoretical Computer Science*.
7. Henkin, L.: 1950, ‘Completeness in the Theory of Types’. *J. Symbolic Logic* **15**, 81–91.
8. Henkin, L.: 1961, ‘Some remarks on infinitely long formulas’. In: *Infinitistic Methods. Proc. Symposium on Foundations of Mathematics*. pp. 167–183.
9. Hintikka, J.: 1998, *The Principles of Mathematics Revisited*. Cambridge University Press.
10. Hintikka, J.: 2002, ‘Hyperclassical logic (a.k.a. IF logic) and its implications for logical theory’. *Bulletin of Symbolic Logic* **8**(3), 404–423.

11. Hintikka, J. and G. Sandu: 1989, 'Informational independence as a semantical phenomenon'. In: J. E. F. et al. (ed.): *Logic, Methodology and Philosophy of Science VIII*. pp. 571–589.
12. Hintikka, J. and G. Sandu: 1996, 'Game-theoretical Semantics'. In: J. van Benthem and A. ter Meulen (eds.): *Handbook of Logic and Language*. Elsevier.
13. Hodges, W.: 1997a, 'Compositional Semantics for a Language of Imperfect Information'. *Logic Journal of the IGPL* **5**(4), 539–563.
14. Hodges, W.: 1997b, 'Some strange quantifiers'. In: G. R. J. Mycielski and A. Salomaa (eds.): *Structures in Logic and Computer Science*, Vol. 1261 of *Lecture Notes in Computer Science*. Springer, pp. 51–65.
15. Lang, S.: 1964, *Algebraic numbers*. Addison-Wesley.
16. Lawvere, F. W.: 1969, 'Adjointness in foundations'. *Dialectica* **23**, 281–296.
17. Lindström, P.: 1969, 'On extensions of elementary logic'. *Theoria* **35**, 1–11.
18. Milner, R.: 1977, 'Fully Abstract Models of Typed Lambda-Calculi'. *Theoretical Computer Science* **4**, 1–22.
19. Mitchell, J. C.: 1996, *Foundations for Programming Languages*. MIT Press.
20. Mitchell, W. P. R. and H. Simmons: 2001, 'Monoid Based Semantics for Linear Formulas'. *J. Symbolic Logic* **66**(4), 1597–1619.
21. O'Hearn, P. W. and D. J. Pym: 1999, 'The Logic of Bunched Implications'. *Bulletin of Symbolic Science* **5**(2), 215–244.
22. Pitts, A.: 2000, 'Categorical logic'. In: S. Abramsky, D. Gabbay, and T. Maibaum (eds.): *Handbook of Logic in Computer Science*, Vol. 5. Oxford University Press, pp. 39–128.
23. Plotkin, G. D.: 1977, 'LCF considered as a Programming Language'. *Theoretical Computer Science* **5**, 223–255.
24. Pym, D. J.: 2002, *The Semantics and Proof Theory of the Logic of Bunched Implications*, Vol. 26 of *Applied Logic Series*. Kluwer.
25. Pym, D. J., P. W. O'Hearn, and H. Yang: 2004, 'Possible worlds and resources: the semantics of BI'. *Theoretical Computer Science* **315**, 257–305.
26. Reynolds, J.: 2002, 'Separation logic: a logic for shared mutable data structures'. In: *Proc. LiCS 2002*.
27. Rosenthal, K. I.: 1990, *Quantales and Their Applications*, No. 234 in Pitman Research Notes in Mathematics. Longman Scientific and Technical.
28. Scott, D. S.: 1969, 'Outline of a Mathematical Theory of Computation'. Technical Monograph PRG-2, Oxford University Computing Laboratory.
29. Tarski, A.: 1936, 'Der Wahrheitsbegriff in den formalisierten Sprachen'. *Studia Philosophica* **1**, 261–405.
30. Tarski, A. and R. Vaught: 1956, 'Arithmetical extensions of relational systems'. *Compositio Mathematica* pp. 81–102.
31. Urquhart, A.: 1972, 'Semantics for Relevant Logics'. *J. Symbolic Logic* **37**(1), 159–169.
32. Väänänen, J.: 2001, 'Second-order logic and foundations of mathematics'. *Bull. Symbolic Logic* **7**(4), 504–520.
33. Väänänen, J.: 2007, *Dependence Logic*, Vol. 70 of *London Mathematical Society Student Texts*. Cambridge University Press.
34. Winskel, G.: 1993, *The Formal Semantics of Programming Languages*. MIT Press.
35. Yetter, D. N.: 1990, 'Quantales and (Noncommutative) Linear Logic'. *J. Symbolic Logic* **55**(1), 41–64.

