The dynamics of imperfect information

Galliani, P.

Publication date
2012

Chapter 2
Logics of Imperfect Information

This chapter is a brief, and neither comprehensive nor impartial, introduction to the field of logics of imperfect information. In Section 2.1, we will recall the history of the development of these logics, from Branching Quantifier Logic to Dependence Logic; and then, in Section 2.2, we will discuss in some detail the definition and the known properties of Dependence Logic and of its variants.

Our treatment of Game Theoretic Semantics in Subsection 2.2.3 is somewhat different from the usual one in that we only consider *memory-free* strategies for our agents; but apart from this, the first two sections of this chapter can be seen as little more than a very condensed and somewhat updated exposition of [65].

Section 2.3, instead, contains novel results. Subsection 2.3.1 is a summary of the main theorem of [8], in which Cameron and Hodges proved, through a combinatorial argument, that no Tarski-like semantics exists for a logic of imperfect information; and Subsection 2.3.2 contains a generalization of this result developed by the author and published in [28].

Finally, this chapter ends with Section 2.4, in which we briefly introduce some of the most important variants and extensions of Dependence Logic.

2.1 From Branching Quantifiers to Dependence Logic

2.1.1 Branching Quantifiers

One aspect of First Order Logic which accounts for much of its expressive power is the fact that this formalism permits *nested quantification*, and, in particular, *alternation* between existential and universal quantifiers. Through this device, it

is possible to specify complex patterns of *dependence and independence* between variables: for example, in the sentence

$$\forall x \exists y \forall z \exists w R(x, y, z, w), \tag{2.1}$$

the existential variable w is a function of both x and z, whereas the existential variable y is a function of x alone.

As Skolem's normal form for (2.1) illustrates, nested quantification can be understood as a restricted form of second-order existential quantification: indeed, the above sentence can be seen to be equivalent to

$$\exists f \exists g \forall x \forall z R(x, f(x), z, g(x, z)). \tag{2.2}$$

In First Order Logic, the notion of quantifier dependence or independence is intrinsically tied to the notion of *scope*: an existential quantifier $\exists y$ *depends* on an universal quantifier $\forall x$ if and only if the former is in the scope of the latter. As observed by Henkin in [36], these patterns can be made more general. In particular, one may consider *branching quantifier expressions* of the form

$$\begin{pmatrix} Q_{11}x_{11} & \dots & Q_{1m}x_{1m} \\ & \dots & \\ Q_{n1}x_{n1} & \dots & Q_{nm}x_{nm} \end{pmatrix}, \tag{2.3}$$

where each Q_{ij} is $\exists$ or $\forall$ and all x_{ij} are distinct. The intended interpretation of such an expression is that each x_{ij} may depend on all $x_{ij'}$ for $j' < j$, but *not* on any $x_{i'j'}$ for $i' \neq i$: for example, in the sentence

$$\begin{pmatrix} \forall x & \exists y \\ \forall z & \exists w \end{pmatrix} R(x, y, z, w) \tag{2.4}$$

the variable y depends on x but not on z, and the variable w depends on z but not on x, and hence the corresponding Skolem expression is

$$\exists f \exists g \forall x \forall z R(x, f(x), z, g(z)) \tag{2.5}$$

If, as we said, quantifier alternation in First Order Logic can be understood as a restricted form of second order existential quantification, then, as a comparison between (2.2) and (2.5) makes clear, allowing branching quantifiers can be understood as a weakening of these restrictions.

How restricted is second order existential quantification in Branching Quantifier Logic, that is, in First Order Logic extended with branching quantifiers

As proved by Enderton and Walkoe in [18] and [73], the answer is *not restricted at all*! Branching Quantifier Logic is precisely as expressive as Existential Second Order Logic (Σ^1_1). Hence, Branching Quantifier Logic can be understood as an alternative approach to the study of Σ^1_1, of its fragments and of its extensions; and indeed, much of the research done on the subject (as well as on the formalisms which we will describe in the next sections) can be seen as an attempt to study Σ^1_1 through the lens of these variants of first-order logic.

2.1.2 Independence Friendly Logic

One striking aspect of the history of logics of imperfect information is how, in many cases, apparently minor modifications to the syntax of a formalism can bring forward profound consequences and insights.

The development of Independence Friendly Logic [39, 37, 54], also called IF Logic, is a clear example of this phenomenon. On a superficial level, the language of IF Logic is a straightforward linearization of the one of Branching Quantifier Logic: rather than dealing the unwieldy quantifier matrices of (2.3), Hintikka and Sandu introduced *slashed quantifiers* $\exists v/V$ with the intended interpretation of "there exists a v, chosen *independently* from the variables in V". For example, the sentence (2.4) can be translated in IF Logic as

$$\forall x \exists y \forall z (\exists w / \{x, y\}) R(x, y, z, w) \tag{2.6}$$

This – at first sight entirely unproblematic – modification led to a number of important innovations on the semantical side.

Game-theoretical explanations for the semantics of branching quantifiers predate the development of IF Logic; but it is with IF Logic that the Game Theoretic Semantics [40] for First Order Logic was extended and adapted to a logic of imperfect information in a formal way. In Subsection 2.2.3, we will present in detail a successor of the Game Theoretic Semantics for IF Logic; but for now, we will limit ourselves to saying that, in the Game Theoretic Semantics for IF Logic, slashed quantifiers correspond to *imperfect information moves* in which the corresponding player has to select a value for the quantified variable *without* having access to the values of the slashed variables.

One interesting phenomenon that IF Logic brings in evidence is *signalling*. Even if a quantified variable is specified to be independent from a previous variable, it is possible to use other quantifiers occurring between the two in order to encode the value of the supposedly "invisible" variable. For example,

it is easy to see that the sentence

$$\forall x(\exists y/\{x\})(x = y), \tag{2.7}$$

corresponding to the Branching Quantifier expression

$$\begin{pmatrix} \forall x \\ \exists y \end{pmatrix} (x = y), \tag{2.8}$$

is not true in any model with at least two elements: indeed, it is not possible for y to be chosen independently from x and still be equal to x in all possible cases.

However, the variant of (2.7) given by

$$\forall x \exists z(\exists y/\{x\})(x = y), \tag{2.9}$$

which may be represented in Branching Quantifier Logic as

$$\begin{pmatrix} \forall x & \exists z \\ \forall z' & \exists y \end{pmatrix} (z = z' \to x = y), \tag{2.10}$$

is instead valid: even if the value of y is to be chosen independently of the value of x, it is possible to let $z = x$ and then choose $y = z$ (or $y = z'$, in the case of the Branching Quantifier formulation).

Signalling, at first, was considered a problematic phenomenon: for example, the variant of IF Logic presented in [38] attempts to prevent it by requiring existential variables to be always independent on previous existential variables. However, such attempts are not without drawbacks (Janssen's paper [47] contains an in-depth discussion of this topic), and most of the modern work on IF Logic tends instead to treat signalling as a useful, if subtle, property of IF Logic ([54]).

Although the Game Theoretic Semantics for IF Logic is a relatively straightforward generalization of the Game Theoretic Semantics for First Order Logic, there is no obvious way of extending Tarski's compositional semantics to the case of IF Logic. In [42], however, Hodges succeeded in finding such a generalization, the *Team Semantics* which we will describe in Subsection 2.2.1.[1] In Team Semantics, satisfaction is predicated over sets of assignments (which, following [65], we will call *teams*), and not over single assignments; and the notion of informational independence contained in the game-theoretical interpretation

[1] The name "Team Semantics" originates from Väänänen's work on Dependence Logic [65], and is now the most common name for this semantical framework.

of slashed quantifiers is now represented as

TS-∃-slash: $M \models_X (\exists x/V)\phi$ if and only if there exists a function $F : X \to \texttt{Dom}(M)$ such that

1. If $s, s' \in X$ assign the same values to all variables other than those in V then $F(s) = F(s')$;
2. For $X[F/v] = \{s[F(s)/v] : s \in X\}$, it holds that $M \models_{X[F/v]} \psi$.

As we will see in Subsection 2.3.1, Cameron and Hodges proved in [8] that it is not possible to create a compositional semantics for IF Logic in which, over finite models, the satisfaction is predicated in terms of single assignments; and, as we will see in Section 2.3.2, this result can be extended to the infinite case if we add a further, natural requirement to our semantics. So, in a sense, Team Semantics is the optimal compositional semantics for logics of imperfect information.

2.1.3 Dependence Logic

In Branching Quantifier Logic and IF Logic both, independence and independence are predicated about *quantifiers*. We can say that a given quantifier $\exists y$ is dependent, or independent, on another quantifier $\forall x$: but neither of these languages offers any instrument to assert that a *variable* y is dependent, or independent, on another variable x.

Väänänen's Dependence Logic (which we will abbreviate as $\mathcal{D}$) arises from the observation that this need not be the case: in the framework of Team Semantics, one may certainly ask whether, with respect to a team X, y is *functionally dependent* on x, in the sense that

$$\forall s, s' \in X, s(x) = s'(x) \Rightarrow s(y) = s'(y). \tag{2.11}$$

This notion of functional dependence is one of the central concepts of Database theory, and has been studied extensively in this context[58, 13]. On the level of sentences, Dependence Logic is equivalent to IF Logic or Branching Quantifier Logic; and the same can be said even about open formulas, as long as the set $\texttt{Var}$ of all relevant variables is known and finite (see [65] for the details). However, the possibility of enquiring directly about dependencies or independencies between free variables is no small advantage.

In Dependence Logic, the assertion that y is functionally dependent on x is written as $=(x, y)$, and, analogously, the assertion that y is functionally dependent on an empty sequence of variables (that is, that y is constant) is

written $=(y)$. It may be instructive to attempt to reproduce the example of signalling of the previous section in this language: whereas (2.7) is translated as

$$\forall x \exists y (=(y) \wedge x = y), \tag{2.12}$$

(2.9) is translated as

$$\forall x \exists z \exists y (=(z, y) \wedge x = y) \tag{2.13}$$

Differently from the case of IF Logic, the functional dependency of y from z is now explicitly declared; and that cannot be avoided, because the Locality Theorem (Proposition 2.2.8 here) states that only the variables which occur free in a Dependence Logic subformula are relevant for its interpretation. Hence, Dependence Logic makes the phenomenon of signalling far less mysterious than it is in IF Logic: instead of a "spooky action at a distance" of a variable z over a subformula $(\exists y/\{x\})(x = y)$ in which such variable does not occur, we now have the perfectly plain fact that if y is a function of z and z can be a function of x then y can be a function of x.[2]

In Section 2.2, we will discuss the language and the semantics of Dependence Logic in more detail. For now, it will suffice to point out out that the development of Dependence Logic has led to a wealth of model-theoretic results, some of which we will recall in Subsection 2.2.2, which advanced significantly our understanding of this class of logics; and that, furthermore, this formalism proved itself highly amenable to the development of variants and extensions, some of which we will describe in Section 2.4.

2.2 Dependence Logic and its Extensions

2.2.1 Team Semantics

Hodges' Team Semantics [42, 65] is the fundamental semantical framework for Dependence Logic, and its interpretation in terms of doxastic states lies at the root of much of its work. In this subsection, we will recall its definition; then in Subsection 2.2.2 we will point out some useful properties of Dependence Logic, and in Subsection 2.2.3 we will define an equivalent Game Theoretic Semantics.

As is common in the study of Dependence Logic, we will assume that all formulas are in Negation Normal Form[3]. Hence, the language of Dependence

[2] This is, in essence, nothing more than William Ward Armstrong's *axiom of transitivity* for functional dependence ([4]).

[3] The reason for this choice, in brief, is that dual negation in Dependence Logic is not a semantic operation, in the sense that not much can be inferred about the falsity conditions of a sentence from its truth conditions. See [7] for the formal statement and proof, and [51]

Logic will be defined as follows:

Definition 2.2.1. Let Σ be a first order signature. The *Dependence Logic formulas* over this signature are given by

$$\phi ::= R\vec{t} \mid \neg R\vec{t} \mid \ =(t_1 \dots t_n) \mid \phi \vee \phi \mid \phi \wedge \phi \mid \exists v\phi \mid \forall v\phi$$

where R ranges over all relation symbols of our signature, $\vec{t}$ ranges over all tuples of terms of the required lengths, n ranges over $\mathbb{N}$, $t_1 \dots t_n$ range over the terms of our signature, and v ranges over the set $\mathtt{Var}$ of all variables of our language.[4]

As can be seen from this definition, the language of Dependence Logic extends the one of First Order Logic with *dependence atoms* $=(t_1 \dots t_n)$, whose intended interpretation is "the value of t_n is a function of the values of $t_1 \dots t_{n-1}$".

The set $\mathtt{Free}(\phi)$ of all *free variables* of a formula ϕ is defined precisely as in First Order Logic, with the additional condition that all variables occurring in a dependence atom $=(t_1 \dots t_n)$ are free in it; and as usual, a formula with no free variables will be called a *sentence*.

As we said, a team is a set of assignments:

Definition 2.2.2. Let M be a first order model and let $\vec{v}$ be a tuple of variables. A *team* X over M with *domain* $\mathtt{Dom}(X) = \vec{v}$ is a set of variable assignments from $\vec{v}$ to $\mathtt{Dom}(M)$.[5]

Given a team X and a tuple $\vec{w} \subseteq \mathtt{Dom}(X)$, we define $\mathtt{Rel}_{\vec{w}}(X)$ as the relation $\{s(\vec{w}) : s \in X\}$. The *relation corresponding to a team* X will be $\mathtt{Rel}(X) = \mathtt{Rel}_{\mathtt{Dom}(X)}(X) = \{s(\mathtt{Dom}(X)) : s \in X\}$.

We now have all the ingredients to give the formal definition of the Team Semantics of Dependence Logic.

Definition 2.2.3. Let M be a first-order model. Then, for all teams X over M and all Dependence Logic formulas ϕ over the same signature of M and with $\mathtt{Free}(\phi) \subseteq \mathtt{Dom}(X)$, we write $M \models_X \phi$ if and only if the team X *satisfies* ϕ in M. This satisfaction relation respect the following rules:

TS-lit: For all first-order literals α, $M \models_X \alpha$ if and only if for all $s \in X$, $M \models_s \alpha$ in the usual first order sense;

for the extension of this result to open formulas.

[4] Expressions of the form $=(t_1 \dots t_n)$ are usually written $=(t_1, \dots, t_n)$, with commas separating the terms. In this work, we will be quite free in omitting or using commas depending on which choice is more readable.

[5] Hence, the domain of a team is only defined up to permutations and repeated elements. This is entirely unproblematic; but if one wishes to avoid this, there is no harm assuming that the set of all variables is linearly ordered. Also, we will be quite free in using set-theoretical terminology when referring to tuples of variables.

TS-dep: For all $n \in \mathbb{N}$ and all terms $t_1 \dots t_n$, $M \models_X =(t_1 \dots t_n)$ if and only if any two $s, s' \in X$ which assign the same values to $t_1 \dots t_{n-1}$ also assign the same value to t_n;

TS-$\vee$: For all ψ_1 and ψ_2, $M \models_X \psi_1 \vee \psi_2$ if and only if $X = X_1 \cup X_2$ for two subteams X_1 and X_2 such that $M \models_{X_1} \psi_1$ and $M \models_{X_2} \psi_2$;

TS-$\wedge$: For all ψ_1 and ψ_2, $M \models_X \psi_1 \wedge \psi_2$ if and only if $M \models_X \psi_1$ and $M \models_X \psi_2$;

TS-$\exists$-strict: For all variables v and formulas ψ, $M \models_X \exists v\psi$ if and only if there exists a function $F : X \to \mathtt{Dom}(M)$ such that $M \models_{X[F/v]} \psi$, where

$$X[F/v] = \{s[F(s)/v] : s \in X\};$$

TS-$\forall$: For all variables v and formulas ψ, $M \models_X \forall v\psi$ if and only if $M \models_{X[M/v]} \psi$, where

$$X[M/v] = \{s[m/v] : s \in X, m \in \mathtt{Dom}(M)\}.$$

If ϕ is a sentence, we say that a model M satisfies ϕ, and we write $M \models \phi$, if and only if $M \models_{\{\emptyset\}} \phi$.[6]

There exists an alternative semantics for existential quantification, which arises naturally from Engström's treatment of generalized quantifiers in Dependence Logic ([19]) and which allows one to select more than one new variable value for assignment. We will call it the *lax* semantics for existential quantification, in comparison to the *strict* semantics **TS-$\exists$-strict** which we described above; and we will define it formally as

TS-$\exists$-lax: For all variables v and formulas ψ, $M \models_X \exists v\psi$ if and only if[7] there exists a function $H : X \to \mathtt{Parts}(\mathtt{Dom}(M))\backslash\{\emptyset\}$ such that $M \models_{X[H/v]} \psi$, where

$$X[H/v] = \{s[m/v] : s \in X, m \in H(s)\}.$$

For logics satisfying the Downwards Closure Property (Proposition 2.2.7 here), the two formulations are equivalent modulo the Axiom of Choice, and **TS-$\exists$-strict** has the advantage of being terser; but as we will see in Chapter 4, for such formalisms as Independence Logic (Subsection 2.4.1) or Inclusion/Exclusion

[6] The choice of $\{\emptyset\}$ as the initial team is entirely arbitrary. Because of Propositions 2.2.6 and 2.2.8, a Dependence Logic sentence is either satisfied by all assignments or only by the empty one.

[7] Here, and through all of this work, we will write $\mathtt{Parts}(A)$ for the set of all subsets of A.

Logic (Chapter 4) only **TS-∃-lax** respects the property of *locality* (Proposition 2.2.8 here).

We end this section by introducing a family of derived connectives which will be useful for some parts of the rest of this work.

Definition 2.2.4. Let ψ_1 and ψ_2 be two Dependence Logic formulas, let $\vec{t}$ be a tuple of terms, and let u_1 and u_2 be two variables not occurring in $\vec{t}$, in ψ_1 or in ψ_2. Then we write $\psi_1 \sqcup_{\vec{t}} \psi_2$ as a shorthand for

$$\exists u_1 \exists u_2 (=(\vec{t}, u_1) \wedge =(\vec{t}, u_2) \wedge ((u_1 = u_2 \wedge \psi_1) \vee (u_1 \neq u_2 \wedge \psi_2))).$$

Proposition 2.2.5. *For all formulas ψ_1 and ψ_2, all tuples $\vec{t}$ of terms, all models M with at least two elements*[8] *whose signature contains that of ψ_1 and ψ_2 and all teams X whose domain contains the free variables of ψ_1 and ψ_2, $M \models_X \psi_1 \sqcup_{\vec{t}} \psi_2$ if and only if $X = X_1 \cup X_2$ for two X_1 and X_2 such that $M \models_{X_1} \psi_1$, $M \models_{X_2} \psi_2$, and furthermore*

$$s \in X_i, \vec{t}\langle s \rangle = \vec{t}\langle s' \rangle \Rightarrow s' \in X_i$$

for all $s, s' \in X$ and all $i \in \{1, 2\}$.

As a special case of "dependent disjunction", we have the *classical disjunction* $\psi_1 \sqcup \psi_2 := \psi_1 \sqcup_{\emptyset} \psi_2$: and by the above proposition, it is easy to see that

$$M \models_X \psi_1 \sqcup \psi_2 \Leftrightarrow M \models_X \psi_1 \text{ or } M \models_X \psi_2$$

as expected.

2.2.2 Some Known Results

In this section, we will recall some properties Dependence Logic. All results are from Väänänen's book [65] unless specified otherwise.

The following four propositions hold for all first-order models M with at least two elements, all formulas ϕ over the signature of M and all teams X, and can be proved by structural induction on ϕ:

Proposition 2.2.6 ([65], §3.9)**.** *$M \models_{\emptyset} \phi$.*

Proposition 2.2.7 (Downwards Closure: [65], §3.10)**.** *If $M \models_X \phi$ and $Y \subseteq X$ then $M \models_Y \phi$.*

[8] In general, we will assume through this whole work that all first-order models which we are considering have at least two elements. As one-element models are trivial, this is not a very onerous restriction.

Proposition 2.2.8 (Locality: [65], §3.27)**.** *If Y is the restriction of X to the free variables of ϕ then*

$$M \models_X \phi \Leftrightarrow M \models_Y \phi.$$

Proposition 2.2.9 ([65], §3.30 and §3.31)**.** *If ϕ is a first-order formula, $M \models_X \phi$ if and only if for all $s \in X$, $M \models_s \phi$ in the usual first order sense.*

The next theorem relates Dependence Logic to Σ^1_1 on the level of sentences:

Theorem 2.2.10 ([65], §6.3 and §6.15)**.** *For any Dependence Logic sentence ϕ there exists a Σ^1_1 sentence Φ such that $M \models \phi$ if and only if $M \models \Phi$. Conversely, for any Σ^1_1 sentence Φ there exists a Dependence Logic sentence ϕ which is satisfied if and only if Φ is satisfied.*

Exploiting the equivalence between Dependence Logic and Σ^1_1, Väänänen then proved a number of model-theoretic properties of Dependence Logic. Here we report the Compactness Theorem and the Löwenheim-Skolem Theorem for Dependence Logic:

Theorem 2.2.11 ([65], §6.4)**.** *If T is a set of Dependence Logic sentences over a finite vocabulary and all finite $T' \subseteq T$ are satisfiable then T itself is satisfiable.*

Theorem 2.2.12 ([65], §6.5)**.** *If ϕ is a Dependence Logic sentence that has an infinite model or arbitrarily large finite models then it has models of all infinite cardinalities.*

What about open formulas? Given a Dependence Logic formula, it is possible to consider the family of all teams which satisfy it; but which families of teams correspond to the satisfaction condition of some Dependence Logic formula?

Because of Proposition 2.2.7, it is clear that not all families of teams which correspond to Σ^1_1-definable relations are expressible in terms of the satisfaction conditions of Dependence Logic formulas. However, [65] has the following result:[9]

Theorem 2.2.13 ([65], §6.2)**.** *Let Σ be a first order signature, let $\phi(\vec{v})$ be a Dependence Logic formula with free variables in ϕ, and let R be a relation symbol not in Σ with arity $|\vec{v}|$. Then there exists a Σ^1_1 sentence $\Phi(R)$, over the signature of $\Sigma \cup \{R\}$, such that*

$$M \models_X \phi \Leftrightarrow M \models \Phi(\mathtt{Rel}(X))^{10}$$

[9] An analogous result was found in [43] with respect to IF Logic.

[10] Here we write $M \models \Phi(\mathtt{Rel}(X))$ to say that if M' is the unique extension of M to $\Sigma \cup \{R\}$ such that $R^{M'} = \mathtt{Rel}(X)$ then $M' \models \Phi$.

for all models M with signature Σ and all team X with domain $\vec{v}$.

Furthermore, R occurs only negatively in Φ.

In [50], Kontinen and Väänänen proved a converse of this result:

Theorem 2.2.14. *Let Σ be a first order signature, let R be a relation symbol not in Σ, let $\vec{v}$ be a tuple of distinct variables with $|\vec{v}|$ equal to the arity of R, and let $\Phi(R)$ be a Σ^1_1 sentence over $\Sigma \cup \{R\}$ in which R occurs only negatively. Then there exists a Dependence Logic formula $\phi(\vec{v})$, with free variables in $\vec{v}$, such that*

$$M \models_X \phi \Leftrightarrow M \models \Phi(\mathtt{Rel}(X))$$

for all models M with signature Σ and all nonempty *teams X with domain $\vec{v}$.*

We finish this subsection by mentioning an easy corollary of this result which will be of some use in Chapter 6:

Corollary 2.2.15. *Let P be any predicate symbol and let $\phi(\vec{v}, P)$ be any Dependence Logic formula with $\mathtt{Free}(\phi) = \vec{v}$. Then there exists a Dependence Logic formula $\phi'(\vec{v})$ such that*

$$M \models_X \phi'(\vec{v}) \Leftrightarrow \exists P \text{ s.t. } M \models_X \phi(\vec{v}, P)$$

for all suitable models M and for all teams X whose domain contains $\vec{v}$.

Proof. By Theorem 2.2.13, there exists a Σ^1_1 sentence $\Phi(R, P)$, in which R occurs only negatively, such that

$$M \models_X \phi(\vec{v}, P) \Leftrightarrow M \models \Phi(X(\vec{v}), P)$$

for all M and all nonempty X with domain $\mathtt{Free}(\phi) = \vec{v}$.

Now consider $\Phi'(R) := \exists P \Phi(R, P)$: by Theorem 2.2.14, there exists a formula $\phi'(\vec{v})$ such that, for all M and all nonempty X with domain $\vec{v}$, $M \models_X \phi'(\vec{v})$ if and only if $M \models \Phi'(X(\vec{v}))$.

By Proposition 2.2.8, the same holds for teams whose domains contain properly $\vec{v}$; and if X is empty then by Proposition 2.2.6 we have that $M \models_X \phi(\vec{v}, P)$ and $M \models_X \phi'(\vec{v})$. Therefore, $\phi'(\vec{v})$ is the formula which we were looking for. □

Definition 2.2.16. Let $\phi(\vec{v}, P)$ be any Dependence Logic formula. Then we write $\exists P \phi(\vec{v}, P)$ for the Dependence Logic formula, whose existence follows from Corollary 2.2.15, such that

$$M \models_X \exists P \phi(\vec{v}, P) \Leftrightarrow \exists P \text{ s.t. } M \models_X \phi(\vec{v}, P)$$

for all suitable models M and all (empty or nonempty) teams X.

Also worth recalling in this subsection is Jarmo Kontinen's PhD thesis [48], which contains a number of results about the finite model theory of Dependence Logic and of fragments thereof. We will not summarize such results here; but we will mention that in that work Jarmo Kontinen proved that the model checking problems for even very simple fragments of Dependence Logic are already NP-complete, and that even relatively small fragments of it do not admit a 0-1 law. We will not make use of these results in the remainder of this work; but the techniques that have been employed in that thesis, and in particular the notion of *k-coherence* that was defined in it, appear to hold no small promise for further clarifying the finite model theory of Dependence Logic and of its extensions.

2.2.3 Game Theoretic Semantics

As we mentioned, the Game Theoretic Semantics for logics of imperfect information predates the Team Semantics which we discussed in the previous section. Dependence Logic and the other formalisms which we will examine here take Team Semantics as their starting point[11]: however, the role of the interplay between Team Semantics and Game Theoretic Semantics in the study of logics of imperfect information is not to be underestimated.

The Game Theoretic Semantics which we describe here differs in some details from the one defined in [65]: most importantly, here we will only admit *memory-free* strategies, which do not look at the past history of the play in order to select the next position. For the purpose of Dependence Logic, or of any other logic of imperfect information satisfying the principle of locality, this will not be problematic: but this should be taken in consideration if one wished to adapt the results of this work of such a logic such as IF Logic, in which locality fails.[12]

Definition 2.2.17. Let ϕ be any Dependence Logic formula. Then $\mathtt{Player}(\phi) \in \{\mathtt{E}, \mathtt{A}\}$ is defined as follows:

1. If ϕ is a first-order literal or a dependence atom, $\mathtt{Player}(\phi) = \mathtt{E}$;
2. If ϕ is of the form $\psi_1 \vee \psi_2$ or $\exists v \psi$ then $\mathtt{Player}(\phi) = \mathtt{E}$;

[11] This differs from the case of IF Logic, in whose study Game Theoretic Semantics is instead generally taken as the fundamental semantical formalism and Team Semantics is treated as a sometimes useful technical device.

[12] The failure of locality in IF Logic is easily seen. Consider any model M with two elements 0 and 1, consider the two teams $X = \{(x := 0, z := 0), (x := 1, z := 1)\}$ and let ϕ be $(\exists y/\{x\})(x = y)$. Then clearly $M \models_X \phi$, but for the restriction $X' = \{(x := 0), (x := 1)\}$ of X to $\mathtt{Free}(\phi)$ it holds that $M \not\models_{X'} \phi$.

3. If ϕ is of the form $\psi_1 \wedge \psi_2$ or $\forall v\psi$ then $\texttt{Player}(\phi) = \texttt{A}$.

The positions of our game are pairs (ψ, s), where ψ is a formula and s is an assignment. The *successors* of a given position are defined as follows:

Definition 2.2.18. Let M be a first order model, let ψ be a formula and let s be an assignment over M. Then the set $\texttt{Succ}_M(\psi, s)$ of the *successors* of the position (ψ, s) is defined as follows:

1. If ψ is a first order literal α then

$$\texttt{Succ}_M(\psi, s) = \begin{cases} \{(\lambda, s)\} & \text{if } M \models_s \alpha \text{ in First Order Logic;} \\ \emptyset & \text{otherwise,} \end{cases}$$

   where λ stands for the empty string;

2. If ψ is a dependence atom then $\texttt{Succ}_M(\psi, s) = \{(\lambda, s)\}$;

3. If ψ is of the form $\exists v\theta$ or $\forall v\theta$ then $\texttt{Succ}_M(\psi, s) = \{(\theta, s[m/v]) : m \in \texttt{Dom}(M)\}$;

4. If ψ is of the form $\theta_1 \vee \theta_2$ or $\theta_1 \wedge \theta_2$ then $\texttt{Succ}_M(\psi, s) = \{(\theta_1, s), (\theta_2, s)\}$.

We can now define formally the semantic games associated to Dependence Logic formulas:

Definition 2.2.19. Let M be a first-order model, let ϕ be a Dynamic Dependence Logic formula, and let X be a team. Then the game $G^M_X(\phi)$ is defined as follows:

- The set $\mathbf{I}$ of the *initial positions* of the game is $\{(\phi, s) : s \in X\}$;
- The set $\mathbf{W}$ of the *winning positions* of the game is $\{(\lambda, s') : s' \text{ is an assignment}\}$;
- For any position (ψ, s'), the *active player* is $\texttt{Player}(\psi)$ and the *set of successors* is $\texttt{Succ}_M(\psi, s')$.

Definition 2.2.20. Let $G^M_X(\phi)$ be as in the above definition. Then a *play* of this game is a finite sequence $\vec{p} = p_1 \dots p_n$ of positions of the game such that

1. $p_1 \in \mathbf{I}$ is an *initial position* of the game;
2. For every $i \in 1 \dots n-1$, $p_{i+1} \in \texttt{Succ}_M(p_1)$.

If furthermore $\texttt{Succ}_M(p_n) = \emptyset$, we say that $\vec{p}$ is *complete*; and if $p_n \in \mathbf{W}$ is a *winning position*, we say that $\vec{p}$ is *winning*.

So far, we did not deal with the satisfaction conditions of dependence atoms at all. Such conditions are made to correspond as *uniformity conditions* over sets of plays:

Definition 2.2.21. Let $G_X^M(\phi)$ be a game, and let P be a set of plays in it. Then P is *uniform* if and only if for all $\vec{p}, \vec{q} \in P$ and for all $i, j \in \mathbb{N}$ such that $p_i = (=(t_1 \dots t_n), s)$ and $q_j = (=(t_1 \dots t_n), s')$ for the same instance of the dependence atom $=(t_1 \dots t_n)$,

$$(t_1 \dots t_{n-1})\langle s \rangle = (t_1 \dots t_{n-1})\langle s' \rangle \Rightarrow t_n\langle s \rangle = t_n\langle s' \rangle.$$

It is not difficult to see that, due to the structure of Dependence Logic formulas, the above condition only needs to be verified for $|\vec{p}| = |\vec{q}|$ and $i = j$.

We will only consider *positional strategies*, that is, strategies that depend only on the current position.

Definition 2.2.22. Let $G_X^M(\phi)$ be as above, and let ψ be any expression such that (ψ, s') is a possible position of the game for some s'. Then a *local strategy* for ψ is a function f_ψ sending each s' into a $(\theta, s'') \in \mathtt{Succ}_M(\psi, s')$.

Definition 2.2.23. Let $G_X^M(\phi)$ be as above, let $\vec{p} = p_1 \dots p_n$ be a play in it, and let f_ψ be a local strategy for some ψ. Then $\vec{p}$ is said to *follow* f_ψ if and only if for all $i \in 1 \dots n-1$ and all s',

$$p_i = (\psi, s') \Rightarrow p_{i+1} = f_\psi(s').$$

Definition 2.2.24. Let $G_X^M(\phi)$ be as above. Then a *global strategy* (for $\mathtt{E}$) in this game is a function f associating to each expression ψ occurring in some nonterminal position of the game and such that $\mathtt{Player}(\psi) = \mathtt{E}$ with some local strategy f_ψ for ψ.

Definition 2.2.25. A play $\vec{p}$ of a game $G_X^M(\phi)$ is said to *follow* a global strategy f if and only if it follows f_ψ for all subformulas ψ of ϕ with $\mathtt{Player}(\psi) = \mathtt{E}$.

Definition 2.2.26. A global strategy f for a game $G_X^M(\phi)$ is said to be *winning* if and only if all complete plays which follow f are winning.

Definition 2.2.27. A global strategy f for a game $G_X^M(\phi)$ is said to be *uniform* if and only if the set of all complete plays which follow f respects the uniformity condition of Definition 2.2.21.

The following result then connects the Game Theoretic Semantics and the Team Semantics for Dependence Logic:

Theorem 2.2.28. *Let M be a first-order model, let X be a team, and let ϕ be any Dependence Logic formula. Then $M \models_X \phi$ if and only if the existential player $\mathtt{E}$ has a uniform winning strategy for $G^M_X(\phi)$.*

The proof of this result is essentially identical to the corresponding proof of [65]. However, since the Game Theoretic Semantics which we just defined is slightly different from the one of that book and since this proof will be the model for a number of similar results of later chapters, it will be useful to report it in full.

Proof. The proof is by structural induction on ϕ.

1. If ϕ is a first-order literal and $M \models_X \phi$ then $M \models_s \phi$ for all $s \in X$. But then the only strategy available to $\mathtt{E}$ in $G^M_X(\phi)$ is winning for this game, and it is trivially uniform.

   Conversely, suppose that $M \not\models_s \phi$ for some $s \in X$. Then the initial position (ϕ, s) is not winning and has no successors, and hence $\mathtt{E}$ does not have a winning strategy for this game.

2. If ϕ is a dependence atom $=(t_1 \dots t_n)$ then the only strategy available to $\mathtt{E}$ for this game sends each initial position $(=(t_1 \dots t_n), s)$ (for $s \in X$) into the winning terminal position (λ, s). This strategy is uniform if and only if any two assignments $s, s' \in X$ which coincide over $t_1 \dots t_{n-1}$ also coincide over t_n, that is, if and only if $M \models_X =(t_1 \dots t_n)$.

3. If ϕ is a disjunction $\psi_1 \vee \psi_2$ and $M \models_X \phi$ then $X = X_1 \cup X_2$ for two teams X_1 and X_2 such that $M \models_{X_1} \psi_1$ and $M \models_{X_2} \psi_2$. Then, by induction hypothesis, there exist two winning uniform strategies f_1 and f_2 for $\mathtt{E}$ in $G^M_{X_1}(\psi_1)$ and $G^M_{X_2}(\psi_2)$ respectively. Then define the strategy f for $\mathtt{E}$ in $G^M_X(\psi_1 \vee \psi_2)$ as follows:

   - If θ is part of ψ_1 then $f_\theta = (f_1)_\theta$;
   - If θ is part of ψ_2 then $f_\theta = (f_2)_\theta$;
   - If θ is the initial formula $\psi_1 \vee \psi_2$ then $f_\theta(s) = \begin{cases} (\psi_1, s) & \text{if } s \in X_1; \\ (\psi_2, s) & \text{if } s \in X_2 \backslash X_1. \end{cases}$

   This strategy is clearly uniform, as any violation of the uniformity condition would be a violation for f_1 or f_2 too. Furthermore, it is winning: indeed, any play of $G^M_X(\psi_1 \vee \psi_2)$ in which $\mathtt{E}$ follows f strictly contains a play of $G^M_{X_1}(\psi_1)$ in which $\mathtt{E}$ follows f_1 or a play of $G^M_{X_2}(\psi_2)$ in which $\mathtt{E}$ follows f_2, and in either case the game ends in a winning position.

Conversely, suppose that f is a uniform winning strategy for $\mathtt{E}$ in $G^M_X(\phi)$. Now let $X_1 = \{s \in X : f_\phi(s) = (\psi_1, s)\}$, let $X_2 = \{s \in X : f_\phi(s) = (\psi_2, s)\}$, and let f_1 and f_2 be the restrictions of f to the subgames corresponding to ψ_1 and ψ_2 respectively. Then f_1 and f_2 are uniform and winning for $G^M_{X_1}(\psi_1)$ and $G^M_{X_2}(\psi_2)$ respectively, and hence by induction hypothesis $M \models_{X_1} \psi_1$ and $M \models_{X_2} \psi_2$. But $X = X_1 \cup X_2$, and hence this implies that $M \models_X \phi$.

4. If ϕ is $\psi_1 \wedge \psi_2$ for some ψ_1 and ψ_2 and $M \models_X \psi_1 \wedge \psi_2$, then $M \models_X \psi_1$ and $M \models_X \psi_2$. By induction hypothesis, this implies that $\mathtt{E}$ has two uniform winning strategies f_1 and f_2 for $G^M_X(\psi_1)$ and $G^M_X(\psi_2)$ respectively. Now let f be the strategy for $G^M_X(\psi_1 \wedge \psi_2)$ which behaves like f_1 over the subgame corresponding to ψ_1 and like f_2 over the subgame corresponding to ψ_2 (it is not up to $\mathtt{E}$ to choose the successors of the initial positions $(\psi_1 \wedge \psi_2, s)$, so she needs not specify a strategy for those). This strategy is winning and uniform, as required, because ψ_1 and ψ_2 are so.

   Conversely, suppose that $\mathtt{E}$ has a uniform winning strategy f for $G^M_X(\psi_1 \wedge \psi_2)$. Since the opponent $\mathtt{A}$ chooses the successor of the initial positions $\{(\psi_1 \wedge \psi_2, s) : s \in X\}$, any element of $\{(\psi_1, s) : s \in X\}$ and of $\{(\psi_2, s) : s \in X\}$ can occur as part of a play in which $\mathtt{E}$ follows f. Now, let f_1 and f_2 be the restrictions of f to the subgames corresponding to ψ_1 and ψ_2 respectively: then f_1 and f_2 are uniform, because f is so, and they are winning for $G^M_X(\psi_1)$ and $G^M_X(\psi_2)$ respectively, because every play of these games in which $\mathtt{E}$ follows f_1 (resp. f_2) starting from a position (ψ_1, s) (resp. (ψ_2, s)) for $s \in X$ can be transformed into a play of $G^M_X(\psi_1 \wedge \psi_2)$ in which $\mathtt{E}$ follows f simply by appending the initial position $(\psi_1 \wedge \psi_2, s)$ at the beginning.

5. If ϕ is $\exists v\psi$ for some ψ and variable $v \in \mathtt{Var}$ and $M \models_X \phi$ then there exists a $F : X \to \mathtt{Dom}(M)$ such that $M \models_{X[F/v]} \psi$. By induction hypothesis, this implies that $\mathtt{E}$ has a uniform winning strategy g for $G^M_{X[F/v]}(\psi)$. Now define the strategy f for $\mathtt{E}$ in $G^M_X(\exists v\psi)$ as

   - If θ is part of ψ then $f_\theta = g_\theta$;
   - $f_\phi(\exists v\psi, s) = (\psi, s[F(s)/v])$.

   Then any play of $G^M_X(\phi)$ in which $\mathtt{E}$ follows f contains a play of $G^M_{X[F/v]}(\psi)$ in which $\mathtt{E}$ follows g, and hence f is uniform and winning.

   Conversely, suppose that $\mathtt{E}$ has a uniform winning strategy f for $G^M_X(\exists v\psi)$. Then define the function $F : X \to \mathtt{Dom}(M)$ so that for all $s \in X$,

$f_\phi(\exists v\psi, s) = (\psi, s[F(s)/v])$, and let g be the restriction of f to ψ. Then g is winning and uniform for $G^M_{X[F/v]}(\psi)$, and hence by induction hypothesis $M \models_{X[F/v]} \psi$, and finally $M \models_X \exists v\psi$.

6. If ϕ is $\forall v\psi$ for some ψ and variable $v \in \mathtt{Var}$ and $M \models_X \phi$ then $M \models_{X[M/v]} \psi$. By induction hypothesis, this implies that $\mathtt{E}$ has a uniform winning strategy f for $G^M_{X[M/v]}(\psi)$. But then the same f is a uniform winning strategy for $\mathtt{E}$ for $G^M_X(\phi)$, since $\mathtt{Player}(\phi) = \mathtt{A}$ and any play of $G^M_X(\phi)$ in which $\mathtt{E}$ follows f contains a play of $G^M_{X[M/v]}(\psi)$ in which $\mathtt{E}$ follows f. Conversely, suppose that $\mathtt{E}$ has a uniform winning strategy f for $G^M_X(\phi)$. Then the same f is a uniform strategy for $\mathtt{E}$ in $G^M_{X[M/v]}(\psi)$, and hence by induction hypothesis $M \models_{X[M/v]} \psi$, and therefore $M \models_X \phi$.

□

As this theorem illustrates, a team X satisfies a formula ϕ in a model M if and only if $\mathtt{E}$ has a strategy which is winning and uniform for the corresponding semantic game and for *any* initial assignment in X. This can be seen a first hint of the doxastic interpretation of Team Semantics: indeed, $M \models_X \phi$ if and only if a hypothetical agent, who believes that the initial assignment (state of things) s belongs in X, can be confident that they will win the semantic game $G^M(\phi)$.

2.3 Sensible Semantics

This section contains Cameron and Hodges' result about the combinatorics of imperfect information ([8]) and their generalization to the infinite case developed by the author in [28].

The significance of these two results for the purpose of this work is the following: by observing that there exists no natural semantics for Dependence Logic in which the satisfaction relation is predicated over single assignments, we obtain some justification for our choice of Team Semantics as the natural framework for the study of logics of imperfect information.

2.3.1 The Combinatorics of Imperfect Information

As we recalled in Subsections 2.2.1 and 2.2.2, Team Semantics is a compositional semantics for logics of imperfect information in which Dependence Logic or

IF Logic formulas are interpreted as downwards-closed[13] sets of teams, which, following Hodges, we will call *suits*.[14]

As Hodges showed in [45], the choice of these kinds of objects comes, in a very natural way, from a careful analysis of the Game Theoretic Semantics for IF-Logic; but is it possible to find an equivalent semantics whose meaning-carrying entities are simpler? In particular, is it possible to find such a semantics in which meanings are *sets of assignments*, as in the case of Tarski's semantics for First Order Logic?

In [8], a negative answer to this question was found, and the corresponding argument will now be briefly reported. In that paper, Cameron and Hodges introduced the concept of "adequate semantics" for IF-Logic, which can be easily adapted to Dependence Logic:

Definition 2.3.1. An *adequate semantics* for Dependence Logic is a function μ that associates to each pair (ϕ, M), where ϕ is a formula and M is a model whose signature includes that of ϕ, a value $\mu_M(\phi)$, and that furthermore satisfies the following two properties:

1. There exists a value TRUE such that, for all sentences ϕ and all models M, $\mu_M(\phi) = \text{TRUE}$ if and only if $M \models \phi$ (according to the Game Theoretic Semantics);

2. For any two formulas ϕ, ψ and for any sentence χ and any model M such that $\mu_M(\phi) = \mu_M(\psi)$, if χ' is obtained from χ by substituting an occurrence of ϕ in χ with one occurrence of ψ then

$$\mu_M(\chi) = \text{TRUE} \Leftrightarrow \mu_M(\chi') = \text{TRUE}.$$

The first condition states that the semantics μ coincides with the Game Theoretic Semantics on sentences, and the second one is a very weak notion of compositionality (which is easily verified to be implied by compositionality in the frameworks of both [44] and [46], the latter of whom can be seen as a descendant of that of [56]).

They also proved the following result:

[13] Because of Proposition 2.2.7, which is easily seen to hold for IF Logic too.

[14] More precisely, in Cameron and Hodges' paper formulas are interpreted as *double suits*, that is, pairs of downward-closed sets of sets of assignments which intersect only in the empty set of assignment. This is because their logic admits a "dual negation" $\neg\phi$ as a primitive operator, and hence their semantics has to keep track explicitly of the truth and the *falsity* conditions of formulas. For our purposes, this difference is not significant: indeed, as Cameron and Hodges proved in Proposition 5.2 of their paper, the number of double suits has the same asymptotic behaviour of the number of suits modulo a factor of two.

Definition 2.3.2. Let M be a first order model, and let $k \in \mathbb{N}$. The a k-suit over M is a set $\mathcal{R}$ of k-ary relations over $\mathtt{Dom}(M)$ which is *downwards closed*, in the sense that

$$R \in \mathcal{R}, S \subseteq R \Rightarrow S \in \mathcal{R}$$

for all $R, S \in \mathtt{Dom}(M)^k$.

Proposition 2.3.3. *Let $f(n)$ be the number of 1-suits over a model M with n elements. Then*

$$f(n) \in \Omega\left(2^{2^n/(\sqrt{\pi\lfloor n/2 \rfloor})}\right)$$

Cameron and Hodges then verified that there exist finite models in which every 1-suit corresponds to the interpretation of a formula with one free variable, and hence that[15]

Proposition 2.3.4. *Let μ be an adequate semantics for Dependence Logic, let x be any variable, and let $n \in \mathbb{N}$. Then there exists a model A_n with n elements, such that*

$$|\{\mu_{A_n}(\phi(x)) : FV(\phi) = \{x\}\}| \geq f(n).$$

Furthermore, the signature of A_n contains only relations.

From this and from the previous proposition, they were able to conclude at once that, for any $k \in \mathbb{N}$, there exists no adequate semantics (and, as a consequence, no compositional semantics) μ such that $\mu_M(\phi)$ is a set of k-tuples whenever $FV(\phi) = \{x\}$: indeed, the number of sets of k-tuples of assignments in a model with n elements is $2^{(n^k)}$, and there exists a $n_0 \in \mathbb{N}$ such that $f(n_0) > 2^{(n_0^k)}$. Then, since μ is adequate we must have that $|\{\mu_{A_{n_0}}(\phi(x)) : FV(\phi) = \{x\}\}| \geq f(n_0) > 2^{n_0^k}$, and this contradicts the hypothesis that μ interprets formulas with one free variables as k-tuples.

However, as Cameron and Hodges observe, this argument does not carry over if we let M range only over infinite structures: indeed, in Dependence Logic (or in IF-Logic) there only exist countably many classes of formulas modulo choice of predicate symbols[16], and therefore for every model A of cardinality $\kappa \geq \aleph_0$ there exist at most $\omega \cdot 2^\kappa = 2^\kappa$ distinct interpretations of IF-Logic formulas in A. Hence, there exists an injective function from the equivalence classes of formulas in A to 1-tuples of elements of A, and in conclusion there exists a semantics which encodes each such congruence class as a 1-tuple.

[15] Again, Cameron and Hodges' results refer to double suits and to IF-Logic rather than to Dependence Logic, but it is easy to see that their arguments are still valid in the Dependence Logic case.

[16] This is not the same of *countably many formulas*, of course, since the signature might contain uncountably many relation symbols.

Cameron and Hodges then conjectured that there exists no reasonable way to turn this mapping into a semantics for IF-Logic:

> *Common sense suggests that there is no sensible semantics for [IF-Logic] on infinite structures A, using subsets of the domain of A as interpretations for formulas with one free variable. But we don't know a sensible theorem along these lines.*

What I will attempt to do in the rest of this section is to give a precise, natural definition of "sensible semantics" according to which Cameron and Hodges' conjecture may be turned into a formal proof: even though, by the cardinality argument described above, it is possible to find a compositional semantics for IF-Logic assigning sets of elements to formulas with one free variable, it will be proved that it is not possible for such a semantics to be also "sensible" according to this definition.

Furthermore, we will also verify that this property is satisfied by Team Semantics, by Tarski's semantics for First Order Logic and by Kripke's semantics for Modal Logic: this, in addition to the naturalness (at least, according to the author's intuitions) of this condition, will go some way in suggesting that this is a property that we may wish to require any formal semantics to satisfy.

2.3.2 Sensible Semantics of Imperfect Information

Two striking features of Definition 2.3.1. are that

1. The class $\mathcal{M}$ of all first order models is not used in any way other than as an *index class* for the semantic relation: no matter what relation exists between two models M and N, no relation is imposed between the functions μ_M and μ_N. Even if M and N were isomorphic, nothing could be said in principle about the relationship between $\mu_M(\phi)$ and $\mu_N(\phi)$!

2. The second part of the definition of adequate semantics does not describe a property of the semantics μ itself, but rather a property of the *synonymy modulo models* relation that it induces. This also holds for the notion of compositionality of [44], albeit not for that of [46]; in any case, in neither of these two formalisms morphisms between models are required to induce morphisms between the corresponding "meaning sets", and in particular isomorphic models may well correspond to non-isomorphic meaning sets.

These observations justify the following definition:

Definition 2.3.5. Let L be a *partial algebra* representing the syntax of our logic[17] for some fixed signature[18] and let $\mathcal{M}$ be the category of the *models* of L for the same signature[19]. Then a *sensible semantics* for it is a triple $(\mathcal{S}, \mathrm{Me}, \mu)$, where

- $\mathcal{S}$ is a subcategory of the category *Set* of all sets;
- Me is a functor from $\mathcal{M}$ to $\mathcal{S}$;
- For every $M \in \mathcal{M}$, μ_M is a function from L to $S_M = \mathrm{Me}(M) \in \mathcal{S}$, called the *meaning set* for L in $\mathcal{S}$

and such that

1. For all $\phi, \psi, \chi \in L$ and for all $M \in \mathcal{M}$, if $\mu_M(\phi) = \mu_M(\psi)$ and χ' is obtained from χ by substituting an occurrence of ϕ as a subterm of χ with an occurrence of ψ, then $\chi' \in L$ and $\mu_M(\chi) = \mu_M(\chi')$;
2. If $f : M \to N$ is an isomorphism between two models $M, N \in \mathcal{M}$, then $\mu_N = \mu_M \circ \mathrm{Me}(f)$ for all formulas $\phi \in L$.

The first condition is, again, a weak variant of compositionality, plus a version of the *Husserl Property* of [44]: if two formulas have the same interpretation in a model M then the operation of substituting one for the other sends grammatical expressions into grammatical expressions with the same interpretation in M. One could strengthen this notion of compositionality after the fashion of [46], by imposing an algebraic structure over each set S_M with respect to the same signature of L and by requiring each μ_M to be an homomorphism between L and M, but as this is not necessary for the purpose of this work we will content ourselves with this simpler statement.

The second condition, instead, tells us something about the way in which isomorphisms between models induce isomorphisms between formula meanings, that is, that the diagram of Figure 2.3.2. commutes whenever f is an isomorphism: if M and N are isomorphic through f then the interpretation $\mu_N(\phi)$ of any formula ϕ in the model N can be obtained by taking the interpretation $\mu_M(\phi) \in S_M$ of ϕ in M and applying the "lifted isomorphism" $\mathrm{Me}(f) : S_M \to S_N$.

[17] That is, the objects of L are the well-formed formulas of our logic and the operations of L are its formation rules.

[18] If the notion of signature is applicable to the logic we are studying; otherwise, we implicitly assume that all models and formulas have the same empty signature.

[19] The choice of morphisms in $\mathcal{M}$ is supposed to be given, and to be part of our notion of model for the semantics which is being considered.

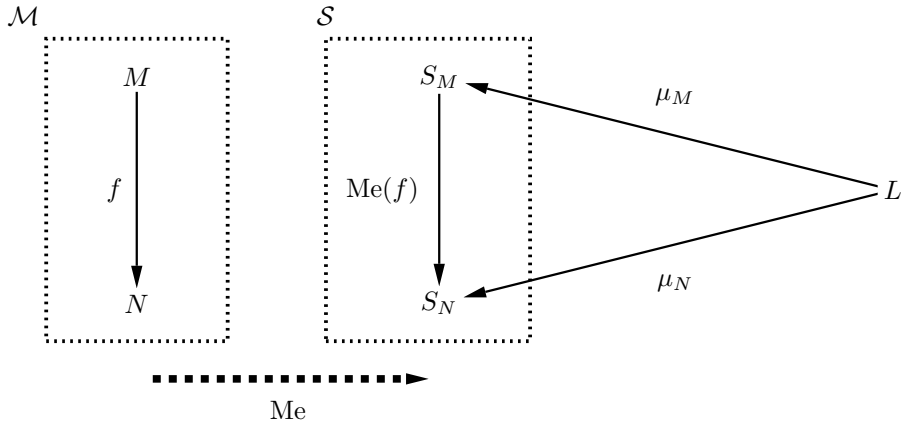

Figure 2.1: Diagram representation of Condition 2 of Definition 2.3.5 (sensible semantics): if $f : M \to N$ is an isomorphism then $\mu_N(\phi) = \mathrm{Me}(f)(\mu_N(\phi))$ for all formulas $\phi \in L$.

Before applying this definition to the case of Dependence Logic, let us verify its naturality by checking that it applies to a couple of very well-known logics with their usual semantics, as well as to Dependence Logic with team semantics:

Proposition 2.3.6. *Let FO be the language of First Order Logic (for some signature Σ which we presume fixed), and let $\mathcal{M}$ be the category of all first order models for the same signature.*

Furthermore, for every $M \in \mathcal{M}$ let S_M be the disjoint union, for k ranging over $\mathbb{N}$, of all sets of k-tuples of elements of M[20] and let Me *be such that* $\mathrm{Me}(M) = S_M$ *for all $M \in \mathcal{M}$ and*

$$\mathrm{Me}(f)(H) = f_{\uparrow}(H) = \{(f(m_1) \dots f(m_k)) : (m_1 \dots m_k) \in H\} \tag{2.14}$$

for all $f : M \to N$ and all $H \in S_M$.

Now, let μ be the usual Tarski semantics, that is, for every model M and formula $\phi(x_1 \dots x_k)$ with $FV(\phi) = \{x_1 \dots x_k\}$ let

$$\mu_M(\phi(x_1 \dots x_k)) = \{(m_1 \dots m_k) \in M^K : M \models_{(x_1:m_1 \dots x_k:m_k)} \phi(x_1 \dots x_k)\}.$$

Then $(\mathcal{S}, \mathrm{Me}, \mu)$ is a sensible semantics for the logic $(FO, \mathcal{M})$.

[20] In particular, this definition implies that S_M contains distinct "empty sets of k-tuples" for all $k \in \mathbb{N}$.

Proof. The first condition is an obvious consequence of the compositionality of Tarski's semantics: if $\Phi[\phi]$ is a well-formed formula, ϕ is equivalent to ψ in the model M and $FV(\phi) = FV(\psi)$ then $\Phi[\psi]$ is also a well-formed formula and it is equivalent to $\Phi[\phi]$ in M.

For the second one, it suffices to observe that if $f : M \to N$ is an isomorphism then

$$M \models_s \phi \Leftrightarrow N \models_{f \circ s} \phi \tag{2.15}$$

for all assignments s and all First Order formulas ϕ. □

Mutatis mutandis, the same holds for Kripke's semantics for Modal Logic:

Proposition 2.3.7. *Let ML be the language of modal logic and let $\mathcal{M}$ be the category of all Kripke models $M = (W, R, V)$, where W is the set of possible worlds, R is a binary relation over W and V is a* valutation function *from atomic propositions to subsets of W. Furthermore, for any $M = (W, R, V) \in \mathcal{M}$ let S_M be the powerset $\mathcal{P}(W)$ of W, and, for every $f : M \to N$, let $\mathrm{Me}(f) : S_M \to S_N$ be such that*

$$\mathrm{Me}(f)(X) = \{f(w) : w \in X\}$$

for all $X \subseteq W$.

Finally, let μ be Kripke's semantics choosing, for each model $M = (W, R, V)$ and each modal formula ϕ, the set $\mu_M(\phi) = \{w \in W : M \models_w \phi\}$: then $(\mathcal{S}, \mathrm{Me}, \mu)$ is a sensible semantics for $(ML, \mathcal{M})$.

Proof. Again, the first part of the definition is an easy consequence of the compositionality of μ. For the second part, it suffices to observe that, if $f : M \to N$ is an isomorphism between Kripke models,

$$M \models_w \phi \Leftrightarrow N \models_{f(w)} \phi$$

for all w in the domain of M, as required. □

Finally, Hodges' Team Semantics for Dependence Logic, whose meaning sets are the disjoint unions over $k \in \mathbb{N}$ of the sets of all k-suits, is also sensible: indeed, for all isomorphisms $f : M \to N$, all sets of k-tuples X and all formulas $\phi(x_1 \dots x_k)$, $M \models_X \phi(x_1 \dots x_k)$ if and only if $N \models_{f_\uparrow(X)} \phi(x_1 \dots x_k)$, where $f_\uparrow$ is defined as in Equation 2.14.

Let us now get to the main result of this work. First, we need a simple lemma:

Lemma 2.3.8. *Let $(\mathcal{S}, \mathrm{Me}, \mu)$ be a sensible semantics for $(\mathcal{D}, \mathcal{M})$, where $\mathcal{D}$ is the language of Dependence Logic (seen as a partial algebra) and $\mathcal{M}$ is the category of all First Order models. Suppose, furthermore, that TRUE is a distinguished value such that $\mu_M(\phi) = TRUE$ if and only if $M \models \phi$ for all models M and sentences ϕ. Then μ is an adequate semantics for Dependence Logic.*

Proof. Obvious from Definition 2.3.1. and Definition 2.3.5. □

Theorem 2.3.9. *Let $\mathcal{M}$ be the class of all infinite models for a fixed signature, let S_M be the set of all sets of k-tuples of elements of M (for all k), and for every $f : M \to N$ let $\mathrm{Me}(f)$ be defined as*

$$\mathrm{Me}(f)(X) = \{f_\uparrow(s) : s \in X\}.$$

for all sets of tuples $X \in S_M$.

Then, for every $k \in \mathbb{N}$, there exists no function μ such that

1. *For all models M and formulas $\phi(x)$ with only one free variable, $\mu_M(S)$ is a set of k-tuples;*
2. *$M \models \phi \Leftrightarrow \mu_M(\phi) = TRUE$ for all $M \in \mathcal{M}$, for all $\phi \in D$ and for some fixed value TRUE;*
3. *$(\mathcal{S}, \mathrm{Me}, \mu)$ is a sensible semantics for Dependence Logic with respect to $\mathcal{M}$.*

Proof. Suppose that such a μ exists for some $k \in \mathbb{N}$: then, by Lemma 2.3.8, μ is an adequate semantics for Dependence Logic.

Let $f(n)$ be the number of suits in a finite model M with n elements, let $h(n) = 2^{2(nk)^k}$, and let n_0 be the least number (whose existence follows from Proposition 2.3.3.) such that $f(n_0) > h(n_0)$. Furthermore, let A_{n_0} be the relational model with n_0 elements, defined as in Proposition 2.3.4., for which Cameron and Hodges proved that any compositional semantics for Dependence Logic must assign at least $f(n_0)$ distinct interpretations to formulas with exactly one free variable x.

Now, let the infinite model B_{n_0} be obtained by adding countably many new elements $\{b_i : i \in \mathbb{N}\}$ to A_{n_0}, by letting $R^{B_{n_0}} = R^{A_{n_0}}$ for all relations R in the signature of A_{n_0} and by introducing a new unary relation P with $P^{B_{n_0}} = \mathtt{Dom}(A_{n_0})$.

It is then easy to see that, with respect to B_{n_0}, our semantics must assign at least $f(n_0)$ different meanings to formulas ϕ with $FV(\phi) = \{x\}$: indeed, if

$\phi^{(P)}$ is the relativization of ϕ with respect to the predicate P we have that

$$\mu_{A_{n_0}}(\phi) = \mu_{A_{n_0}}(\psi) \Leftrightarrow \mu_{B_{n_0}}(\phi^{(P)}) = \mu_{B_{n_0}}(\psi^{(P)}),$$

and we already know that $|\{\mu_{A_{n_0}}(\phi) : FV(\phi) = \{x\}\}| = f(n_0)$.

Now, suppose that μ is sensible and $\mu_{B_{n_0}}(\phi)$ is a set of k-tuples for every formula $\phi(x)$: then, since every permutation $\pi : B_{n_0} \to B_{n_0}$ that pointwise fixes the element of A_{n_0} is an automorphism of B_{n_0}, we have that

$$\mathrm{Me}(\pi)(\mu_{B_{n_0}}(\phi)) = \mu_{B_{n_0}}(\phi)$$

for all such π.

But then $|\mu_{B_{n_0}}(\phi) : FV(\phi) = \{x\}| \leq h(n_0)$, since there exist at most $2^{2(n_0k)^k}$ equivalence classes of tuples with respect to the relation

$$\overline{b} \equiv \overline{b}' \Leftrightarrow \exists f : B_{n_0} \to B_{n_0}, f \text{ automorphism, s.t. } f_{\uparrow} b = c.$$

Indeed, one may represent such an equivalence class by first specifying whether it contains any element of A_{n_0}, then listing without repetition all elements of A_{n_0} occurring in $\overline{b}$, padding this into a list $\overline{m}$ to a length of k by repeating the last element, and finally encoding each item b_i of $\overline{b}$ as an integer t_i in $1 \dots k$ in such a way that

- If $b_i \in A_{n_0}$, $m_{t_i} = \overline{b}_i$ and $\overline{m}_{t_i - 1} \neq \overline{m}_{t_i}$ whenever $t_i > 0$;
- If $b_i \notin A_{n_0}$, $\overline{m}_{t_i} = \overline{m}_{t_i - 1}$ whenever $t_i > 0$;
- $t_i = t_j$ if and only if $b_i = b_j$.

In total, this requires $1 + k\log(n_0) + k\log(k)$ bits, and therefore there exist at most $2(n_0k)^k$ such equivalence classes; and since each $\mu_{B_{n_0}}(\phi)$ is an union of these equivalence classes, there are at most $2^{2(n_0k)^k}$ possible interpretations of formulas with one free variable.

But this contradicts the fact that $f(n_0) > h(n_0)$, and hence no such semantics exists. □

2.4 Extensions of Dependence Logic

In this last section of the chapter, we will briefly describe some variants of Dependence Logic. We make no pretence of completeness: in particular, we will not discuss *Modal Dependence Logic* [67] and its variants here, nor in any other part of this thesis.

2.4.1 Independence Logic

Independence Logic ($\mathcal{I}$) is a formalism, developed by Grädel and Väänänen ([33]), which replaces the dependence atoms $=(t_1 \dots t_n)$ of Dependence Logic with[21] *independence atoms* $\vec{t}_2 \perp_{\vec{t}_1} \vec{t}_3$, where the $\vec{t}_i$ are tuples of terms not necessarily of the same lengths and where

TS-indep: $M \models_X \vec{t}_2 \perp_{\vec{t}_1} \vec{t}_3$ if and only if any for any two $s_1, s_2 \in X$ with $\vec{t}_1\langle s_1\rangle = \vec{t}_1\langle s_2\rangle$ there exists a $s_3 \in X$ with $(\vec{t}_1\vec{t}_2)\langle s_1\rangle = (\vec{t}_1\vec{t}_2)\langle s_3\rangle$ and $(\vec{t}_1\vec{t}_3)\langle s_2\rangle = (\vec{t}_1\vec{t}_3)\langle s_3\rangle$.

This condition is best understood in terms of *informational independence*: in brief, $M \models_X \vec{t}_2 \perp_{\vec{t}_1} \vec{t}_3$ if and only if, in X, all the information about the value of $\vec{t}_3$ which can be inferred by the values of $\vec{t}_1$ and $\vec{t}_2$ can already be inferred by the value of $\vec{t}_1$ alone.

The downwards closure property of Proposition 2.2.7 does not transfer to the case of Independence Logic: for example, the team

$$X = \begin{array}{c|cc} & x & y \\ \hline s_1 & 0 & 0 \\ s_2 & 0 & 1 \\ s_3 & 1 & 0 \\ s_4 & 1 & 1 \end{array}$$

satisfies the independence statement $x \perp_\emptyset y$,[22], but the same cannot be said of its subset

$$Y = \begin{array}{c|cc} & x & y \\ \hline s_1 & 0 & 0 \\ s_4 & 1 & 1 \end{array}$$

in which, as it is easy to see, x and y are not informationally independent.

As pointed out in [33], a dependence atom $=(t_1 \dots t_n)$ can be expressed in Independence Logic as $t_n \perp_{t_1 \dots t_{n-1}} t_n$; and moreover, Theorems 2.2.10 and 2.2.13 can be adapted to the case of Independence Logic, although in the case of the second one we lose the condition that R occurs only negatively, and hence we have that

Theorem 2.4.1. *Any Dependence Logic sentence is equivalent to some Independence Logic sentence, and any Independence Logic sentence is equivalent to*

[21] In this, Independence Logic can be thought of as a variant of Dependence Logic through *generalized dependence atoms*, in the sense suggested by Jarmo Kontinen in the conclusion of [48] and made explicit by Kuusisto in [53].

[22] As a shorthand, we will occasionally write $\vec{t}_1 \perp \vec{t}_2$ instead of $\vec{t}_1 \perp_\emptyset \vec{t}_2$.

some Dependence Logic sentence.

However, the problem of finding the equivalent of Theorem 2.2.14 for the case of Independence Logic in order to characterize the definable classes of teams of this logic was mentioned in [33] as an open problem:

> *The main open question raised by the above discussion is the following, formulated for finite structures:*
>
> **Open Problem:** *Characterize the NP properties of teams that correspond to formulas of independence logic.*

In Chapter 4, we will answer this question by proving that *all* Σ^1_1 properties (and hence, by Fagin's Theorem, all NP properties) of teams corresponds to formulas of Independence Logic.[23]

2.4.2 Intuitionistic and Linear Dependence Logic

In [3], Abramsky and Väänänen examined the *adjoints* of Dependence Logic conjunction and disjunction. In other words, they introduced two downwards closed connectives $\psi_1 \to \psi_2$ and $\psi_1 \multimap \psi_2$ such that

$$\phi \wedge \psi \models \theta \Leftrightarrow \phi \models \psi \to \theta$$

and

$$\phi \vee \psi \models \theta \Leftrightarrow \phi \models \psi \multimap \theta$$

respectively.

The satisfaction conditions corresponding to these two requirements are:

TS-$\to$: $M \models_X \psi \to \theta$ if and only if for all $Y \subseteq X$, $M \models_Y \psi \Rightarrow M \models_Y \theta$;

TS-$\multimap$: $M \models_X \psi \multimap \theta$ if and only if for all Y such that $M \models_Y \psi$, $M \models_{X \cup Y} \theta$.

Because of the similarity between these two rules and the semantics for implication in intuitionistic and linear logic respectively, the $\to$ operator has been dubbed the "intuitionistic implication" and the $\multimap$ operator has been dubbed the "linear implication", and the languages $\mathcal{D}(\to)$ and $\mathcal{D}(\multimap)$ obtained by adding them to Dependence Logic have been dubbed *Intuitionistic Dependence Logic* and *Linear Dependence Logic* respectively.

[23] This result, as well as all the content of that chapter except Section 4.6, has been published by the author in [30].

One interesting aspect of the linear implication operator, mentioned in [3], is that it can be used to *decompose* a dependence atom in terms of constancy atoms: indeed, for all models M, teams X, integers $n \in \mathbb{N}$ and terms $t_1 \dots t_n$, one can verify that

$$M \models_X =(t_1 \dots t_n) \Leftrightarrow M \models_X =(t_1) \to (\dots \to (=(t_{n-1}) \to =(t_n)) \dots).$$

Intuitionistic and Linear Dependence Logic are strictly more expressive than Dependence Logic: in particular, the set of sentences of these languages is closed by contradictory negation, since for any model M and sentence ϕ

$$M \models_{\{\emptyset\}} \phi \to \bot \Leftrightarrow M \models_{\{\emptyset\}} \phi \multimap \bot \Leftrightarrow M \not\models_{\{\emptyset\}} \phi,$$

and therefore Intuitionistic Dependence Logic and Linear Dependence Logic both contain $\Sigma^1_1 \cup \Pi^1_1$. In [74], Yang proved that Intuitionistic Dependence Logic is, in fact, equivalent to full Second Order Logic.

2.4.3 Team Logic

Team Logic $\mathcal{T}$ [66, 65] extends Dependence Logic with a *contradictory negation* operator $\sim \phi$ whose semantics is given by

TS-$\sim$: $M \models_X \sim \phi$ if and only if $M \not\models_X \phi$.

It is a very expressive formalism, which is equivalent to full Second Order Logic over sentences; and furthermore, as Kontinen and Nurmi proved in [49], *all* second-order relations which are definable in Second Order Logic correspond to classes of teams which are definable in Team Logic.

The language of Team Logic is somewhat different from that of Dependence Logic or of most other logics of imperfect information. The disjunction $\phi \vee \psi$ of Dependence Logic, with the corresponding rule **TS-$\vee$**, is written in Team Logic as $\phi \otimes \psi$; instead, $\phi \vee \psi$ in Team Logic represents $\sim ((\sim \phi) \vee (\sim \psi))$, which is easily seen to be equivalent to the "classical" disjunction which we defined as $\phi \sqcup \psi$ in Subsection 2.2.1. Similarly, the universal quantifier $\forall x \phi$ of Dependence Logic is written in Team Logic as $!x\phi$, and $\forall x \phi$ is instead taken as a shorthand for $\sim (\exists x (\sim \phi))$. One surprising aspect of Team Logic is that a sentence $\phi \in \mathcal{T}$ can have *four* possible truth values:

$\bot$: No team satisfies ϕ;

$\top$: Both $\emptyset$ and $\{\emptyset\}$ satisfy ϕ;

1: $\{\emptyset\}$ satisfies ϕ, but $\emptyset$ does not;

0: $\emptyset$ satisfies ϕ, but $\{\emptyset\}$ does not.

Team Logic is the most expressive logic of imperfect information which we will discuss in this work. It is a remarkably powerful formalism, about which much is not known yet; and while this work is mostly concerned with more treatable sublogics, it is the hope of the author that the ideas discussed here (and, in particular, the doxastic interpretation of Chapter 7) may provide some incentive for the study of this intriguing and powerful logic.