The Dynamics of Imperfect Information

Pietro Galliani

The Dynamics of Imperfect Information

ILLC Dissertation Series DS-200X-NN

INSTITUTE FOR LOGIC, LANGUAGE AND COMPUTATION

For further information about ILLC-publications, please contact

Institute for Logic, Language and Computation
Universiteit van Amsterdam
Science Park 904
1098 XH Amsterdam
phone: +31-20-525 6051
fax: +31-20-525 5206
e-mail: illc@uva.nl
homepage: http://www.illc.uva.nl/

Printed and bound by GVO Drukkers & Vormgevers B.V.

ISBN: 978–90–9026932–0

The Dynamics of Imperfect Information

Academisch Proefschrift

ter verkrijging van de graad van doctor aan de
Universiteit van Amsterdam
op gezag van de Rector Magnificus
prof.dr. D.C. van den Boom
ten overstaan van een door het college voor
promoties ingestelde commissie, in het openbaar
te verdedigen in de Aula der Universiteit
op vrijdag 21 september 2012, te 13.00 uur

door

Pietro Galliani

geboren te Bologna, Italië.

Promotor:
Prof. dr. J. Väänänen
Overige leden:
Prof. dr. S. Abramsky
Prof. dr. J.F.A.K. van Benthem
Dr. A. Baltag
Dr. U. Endriss
Prof. dr. E. Grädel
Prof. dr. D.H.J. de Jongh
Prof. dr. B. Löwe
Faculteit der Natuurwetenschappen, Wiskunde en Informatica

to Dino and Franca Davoli

thanks for everything

Contents

Acknowledgments

First of all, I would like to thank my supervisor Jouko Väänänen: his advice and encouragement during all these years has been nothing short of invaluable. Furthermore, I would also like to thank all the members of the LINT (Logic for Interaction) LogICCC subproject for many stimulating discussions, and the European Science Foundation as a whole for this fascinating and highly successful project.

The Dependence Logic research community as a whole has been extremely welcoming, and an endless source of inspiration and suggestions. In particular, I wish to thank Samson Abramsky, Dietmar Berwanger, Denis Bonnay, Fredrik Engström, Erich Grädel, Lauri Hella, Jarmo and Juha Kontinen, Antti Kuusisto, Allen Mann, Jonni Virtema, and Fan Yang for many, many interesting and fruitful discussions. I also thank Alexandru Baltag, Johan van Benthem, Dietmar Berwanger and Dag Westerståhl for many useful suggestions and comments on my work.

Furthermore, I thank Karin Gigengack, Tanja Kassenaar, Peter van Ormondt and Marco Vervoort for helping me in many practical matters, and Stefanie Kooistra for translating the abstract of this thesis into Dutch.

Moreover, I thank Fausto Barbero, Vincenzo Ciancia, Cédric Dégremont, Sujata Ghosh, Nina Gierasimvzuk, Umberto Grandi, Lauri Keskinen, Lena Kurzen, Ştefan Minică, Daniele Porello, Federico Sangati, Jakub Szymanik, Paolo Turrini, Fernardo Velázquez-Quesada and Joel and Sara Uckelman; and finally, I thank my mother Chiara and my father Pierluigi for sending me many packages of excellent Italian cheeses and sausages, a fundamental source of inspiration and motivation for me during all of my studies.

Amsterdam   Pietro Galliani
April, 2012

Chapter 1

Introduction

Logics of imperfect information are extensions of First Order Logic[1] in which very general patterns of dependence and independence between logical operations and/or between variables are allowed. Among these logics, Dependence Logic is particularly suitable for the study of the very notion of dependence, because it represents dependence of variables directly by means of special atomic formulas. After the introduction of Dependence Logic in 2007, a considerable amount of results (some of which we will summarize in Chapter 2) have been obtained about it and its extensions. In this thesis we solve several open problems of the area and suggest new ways to think about this family of logics.

Logics of imperfect information admit a Game Theoretic Semantics, an imperfect information generalization of the Game Theoretic Semantics for First Order Logic; and furthermore, they also admit an equivalent Team Semantics (also referred to in the literature as Hodges Semantics or Trump Semantics), which instead generalizes Tarski's semantics for First Order Logic. Team Semantics extends Tarski's semantics by defining the satisfaction relation not in terms of single assignments but in terms of sets of assignments, called *teams*.

This thesis is a Team Semantics-centered exploration of the properties of variants and extensions of Dependence Logic. Our two principal claims, for which we will build gradually support through this whole work and which will find their most general formulations in Chapters 6 and 7, are the following:

1. Teams represent information states;
2. Formulas in Dependence Logic and its variants can be interpreted in terms of *transitions* between information states.

[1] Or, more rarely, of other logics: see for example the *Modal Dependence Logic* of [67], or the *Independence-Friendly Modal Logic*(s) of [64, 6].

The first claim is not new, and in a way it is already implicit in Hodges' proof of the equivalence between Team Semantics and Game Theoretic Semantics. However, what is (to the knowledge of the author) new is the idea that the teams-as-information-states interpretation of Team Semantics can (and, in the opinion of the author, *should*) be used as the main driving impulse towards the further development of this fascinating area of research, as we try to do in this work.

Chapter 2 The second chapter is a brief introduction to the study of logics of imperfect information. First, in Section 2.1, we recall the history of the development of such logics, from the early days of Branching Quantifiers Logic until the creation of Dependence Logic. This account is neither complete nor impartial: more could certainly be said about the development of Independence Friendly Logic, for example, during which many of the salient peculiarities of logics of imperfect information were first isolated. Furthermore, we will say nothing about *modal logics of imperfect information* such as IF Modal Logic or Modal Dependence Logic: indeed, even though such formalisms are certainly of no small interest, the present work will be exclusively concerned with *first order* logics of imperfect information.

Then, in Section 2.2, we introduce formally Dependence Logic, its Team Semantics, its Game Theoretic Semantics, and some of its main properties. Our presentation here is essentially a summarized and updated version of the introduction to Dependence Logic contained in [65]. The principal differences between our approach and the one of Väänänen's book (to which we encourage the reader to refer for a more in-depth introduction to the field) are the following:

1. We assume that all formulas are in negation normal form, and hence we do not take the "dual negation" $\neg\phi$ as a primitive of our language. It is easy to recover it inductively, of course, by defining $\neg(\phi \ [\vee \mid \wedge] \ \psi) := (\neg\phi) \ [\wedge \mid \vee] \ (\neg\psi)$, $\neg([\exists v \mid \forall v] \ \psi) := [\forall v \mid \exists v] \ (\neg\psi)$, and so on; but as [7, 51] show, in Dependence Logic not much can be inferred about the satisfaction conditions of a formula from the satisfaction conditions of its negation. Furthermore, having the dual negation as one of our primitives would have forced us to add the (in the opinion of the author, rather counterintuitive) rule stating that the negation of a dependence atom is true only in the empty team.

2. When introducing the Team Semantics in Subsection 2.2.1, we give the rule for existential quantification in both the *strict* version **TS-∃-strict** and in the *lax* version **TS-∃-lax**. For the case of Dependence Logic,

these two variants are easily seen to be equivalent, and the strict version (which is adopted in Väänänen's work, and more in general in the study of Dependence Logic and of other downwards closed[2] logics of imperfect information) has the advantage of being terser; however, as we will argue in Subsection 4.3.2, in the cases of some interesting non downwards-closed logics of imperfect information only the lax existential quantifiers satisfy the (very useful and natural) property of *locality* in the sense of Proposition 2.2.8.

3. In Subsection 2.2.2, we mention the characterization of definability in Dependence Logic of Kontinen and Väänänen (Theorem 2.2.14 here), which will be of fundamental importance in a number of parts of our work and which is certainly among the most important model-theoretic results in the field of Dependence Logic.

4. Our treatment of Game Theoretic Semantics in Subsection 2.2.3 differs in some details from the standard one: in particular, we only take in consideration *positional*, or memoryless, strategies. This technically simpler choice will allow us to extend and adapt more easily this game theoretic semantics to extensions or variants of Dependence Logic.

5. In Section 2.3, we present a very general result by Cameron and Hodges about the combinatorial properties of semantics for logics of imperfect information, as well as the author's generalization of this discovery to the case of infinite models. Cameron and Hodges' theorem, which we report in Subsection 2.3.1, is a highly abstract result about the question of which semantics may capture the behaviour of a logic of imperfect information: and, in particular, one of its consequences is that no compositional semantics for IF Logic or Dependence Logic may send formulas with one free variable into sets of tuples of elements (as Tarski's semantics does for First Order Logic). Cameron and Hodges' theorem, however, fails over infinite models, as an easy counting argument demonstrates; and in their paper [8], they suggest that even in this case no "sensible" semantics for a logic of imperfect information may have that property. In Subsection 2.3.2, which corresponds to the publication [28], we introduce a precise definition of "sensible semantics", argue that it is a natural one, and prove Cameron and Hodges' conjecture with respect to it.

[2] A logic of imperfect information is said to be *downwards closed* if it satisfies an analogue of Proposition 2.2.7.

Finally, in Section 2.4, we introduce some of the most important known extensions and variants of Dependence Logic:

1. Independence Logic[3] adds to the language of First Order Logic *independence atoms* $\vec{t}_2 \perp_{\vec{t}_1} \vec{t}_3$, with the intended meaning of "$\vec{t}_2$ and $\vec{t}_3$ are informationally independent given $\vec{t}_1$." As Grädel and Väänänen show in [33], this logic is equivalent to Dependence Logic with respect to sentences but is more expressive than it when it comes to open formulas and definability of classes of teams. In Chapter 4, we will answer a question asked by Grädel and Väänänen in their paper, by characterizing definability in Independence Logic and hence finding the analogue of Theorem 2.2.14 for Independence Logic.

2. Linear and Intuitionistic Dependence Logics add to the language of Dependence Logic a *linear implication* $\phi \multimap \psi$ or an *intuitionistic implication* $\phi \rightarrow \psi$, respectively, as the downwards-closed *adjoints* of the disjunction or conjunction of Dependence Logic. These formalisms are known to be more powerful than Dependence Logic proper; and in Section 7.4, we will show that intuitionistic and linear implication have very natural interpretations in terms of *predictions* about the outcomes of belief updates.

3. Team Logic adds to Dependence Logic a *contradictory negation* $\sim\phi$. It is a remarkably powerful and expressive formalism, which is roughly equivalent – as Kontinen and Nurmi's result in [49] shows – to Second Order Logic; and in Chapter 7, we will develop a notational variant of it and examine its doxastic significance.

Chapter 3 The third chapter is an adaptation of the author's publications [26, 27]. We add *announcement operators* to the language of Dependence Logic and examine their properties, showing how they may be employed to decompose dependence atoms (and also the $\forall^1$ quantifier introduced by Kontinen and Väänänen in [50]). Furthermore, we show that as the name suggests, these operators can be interpreted in Game Theoretic Semantics in terms of public announcements, and we illustrate how the Ehrenfeucht-Fraïssé game for Dependence Logic may be adapted to the $\forall^1$ quantifier and its variants. Finally, we solve an open problem of [50] by proving that the $\forall^1$ quantifier is not *uniformly* definable in Dependence Logic, in the sense that there exists no context $\Phi[\cdot]$ such that $\Phi[\psi]$ is equivalent to $\forall^1 v\psi$ for all Dependence Logic formulas ψ.

[3] Independence Logic is not to be confused with Independence *Friendly* (IF) Logic, which is historically antecedent to Dependence or Independence Logic and which we will briefly discuss in Subsection 2.1.2.

Chapter 4 The fourth chapter is a study of some variants of Dependence Logic obtained by considering non-functional, Database Theory-inspired notions of dependence, and it is a revised and expanded version of the publication [30]. We show that Constancy Logic, that is, the fragment of Dependence Logic containing only constancy atoms[4], is equivalent to First Order Logic with respect to sentences; and furthermore, we prove that Engström's *Multivalued Dependence Logic* is, in fact, equivalent to Independence Logic. We then define *Inclusion Logic*, *Equiextension Logic*, *Exclusion Logic* and *Inclusion/Exclusion Logic* by examining various notions of non-functional dependence, and we prove that

1. Inclusion Logic and Equiextension Logic are equivalent, are contained in Independence Logic, and are neither contained in Dependence Logic nor contain it;
2. Exclusion Logic is equivalent to Dependence Logic;
3. Inclusion/Exclusion Logic is equivalent to Independence Logic.

We then adapt the Game Theoretic Semantics of Dependence Logic to the case of Inclusion/Exclusion Logic. This is unproblematic; however, an interesting peculiarity of our treatment is that, in order to obtain the lax interpretations of the existential quantifiers, we need to consider *nondeterministic* strategies.

We then use the results found so far to prove that all NP properties of teams are expressible in Independence Logic (or, equivalently, in Inclusion/Exclusion Logic), thus solving an open problem of [33]; and finally, we show that, just as dependence atoms can be decomposed in terms of constancy logic and announcement operators, independence atoms can be decomposed in terms of constancy atoms, *inconstancy* atoms, announcement operators, and other connectives. This highlights the value of announcement operators as tools for reducing complex dependence or independence notions to simpler ones.

Chapter 5 The fifth chapter, which corresponds to the publication [29], is concerned with proof-theoretic issues. With respect to their standard Team Semantics, Dependence Logic and its extensions are non-axiomatizable; however, we can introduce *general models* and a *General Semantics* over them which permits only a limited form of quantification over teams, much in the same sense in which Henkin's semantics for Second Order Logic permits only a limited form of quantification over sets. We will see that for Inclusion/Exclusion Logic or

[4] A constancy atom is simply a dependence atoms which states that a given variable term or variable depends on *nothing*.

Independence Logic, it suffices to examine some very special kinds of general models; and furthermore, that in these models it is possible to represent teams syntactically as first order formulas with parameters. We will then introduce a proof system and verify its soundness and completeness with respect to our semantics. The intellectual debt of the author to Väänänen is, for this chapter, even greater than for the rest of the work: the idea of studying general semantics for logics of imperfect information comes from a direct suggestion of him, and our treatment is very much inspired by the course on Second Order Logic which Väänänen taught at the University of Helsinki in 2011 and which the author followed.

Chapter 6 The sixth chapter is an examination of the dynamics of information change which lies underneath the Team Semantics of Dependence Logic. After recalling van Benthem's mutual embedding result between First Order Logic and Dynamic Game Logic, we develop an imperfect-information, player-versus-Nature variant of Dynamic Game Logic and prove the existence of a mutual embedding between it and Dependence Logic. We use the insights arising from this construction to create two variants of Dependence Logic: Transition Dependence Logic, in which formulas are interpreted as assertions *about* games against Nature, and Dynamic Dependence Logic, a Dynamic Semantics-inspired variant in which formulas are interpreted *as* games against Nature. From a technical point of view, the study of these variants requires the development of *Team Transition Semantics*, a variant of Team Semantics in which formulas are interpreted as transition systems between teams: and Hodges' equivalence proof between Team Logic and Game Theoretic Semantics, adapted to this new formalism, shows that satisfaction conditions in Team Transition Semantics correspond precisely to *reachability* conditions in Game Theoretic Semantics.

Chapter 7 In the last chapter, we gradually develop a notational variant of Team Logic which contains many of the operators and concepts which we discussed in the rest of the work. The resulting system has no pretence of being of independent interest; rather, we use it as a means for highlighting and emphasizing the doxastic interpretation of Team Semantics. In particular, we show that quantifiers, the "tensorial" disjunction $\phi \otimes \psi$ and the implications all have natural interpretations in terms of belief updates, and that first order expressions and dependence atoms have natural interpretations in terms of belief descriptions. Together with the other results of this work, this lends strong support to our claim that the doxastic interpretation of Team Semantics is

1. A solid, comprehensive point of view under which to understand Team

Semantics;

2. A useful testing ground for the development of further extensions and variants of our logics;

3. A highly promising area of application for logics of dependence and independence.

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

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

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

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

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 `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 $\texttt{Var}$ 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 $\texttt{Free}(\phi)$ 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* $\texttt{Dom}(X) = \vec{v}$ is a set of variable assignments from $\vec{v}$ to $\texttt{Dom}(M)$.[5]

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

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 $\texttt{Free}(\phi) \subseteq \texttt{Dom}(X)$, 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*

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

[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 $\texttt{Player}(\phi) \in \{\texttt{E}, \texttt{A}\}$ is defined as follows:

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

[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 $\texttt{Free}(\phi)$ 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 $\vec{p} = p_1 \ldots p_n$ of positions of the game such that

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

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 $G^M_X(\phi)$ 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 $p_i = (=(t_1 \ldots t_n), s)$ and $q_j = (=(t_1 \ldots t_n), s')$ for the same instance of the dependence atom $=(t_1 \ldots t_n)$,

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

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 $G^M_X(\phi)$ 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 $G^M_X(\phi)$ be as above, let $\vec{p} = p_1 \ldots p_n$ 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 $i \in 1 \ldots n-1$ and all s',

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

Definition 2.2.24. Let $G^M_X(\phi)$ 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 $G^M_X(\phi)$ 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 $G^M_X(\phi)$ 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 $G^M_X(\phi)$ 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* `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 `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 `E` does not have a winning strategy for this game.

2. If ϕ is a dependence atom $=\!(t_1 \ldots t_n)$ then the only strategy available to `E` for this game sends each initial position $(=\!(t_1 \ldots t_n), s)$ (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 $t_1 \ldots t_{n-1}$ also coincide over t_n, that is, if and only if $M \models_X =\!(t_1 \ldots t_n)$.

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 `E` in $G^M_{X_1}(\psi_1)$ and $G^M_{X_2}(\psi_2)$ respectively. Then define the strategy f for `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 `E` follows f strictly contains a play of $G^M_{X_1}(\psi_1)$ in which `E` follows f_1 or a play of $G^M_{X_2}(\psi_2)$ in which `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 E in $G_X^M(\phi)$. 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 $G_{X_1}^M(\psi_1)$ and $G_{X_2}^M(\psi_2)$ 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 E has two uniform winning strategies f_1 and f_2 for $G_X^M(\psi_1)$ and $G_X^M(\psi_2)$ respectively. Now let f be the strategy for $G_X^M(\psi_1 \wedge \psi_2)$ 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 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 E has a uniform winning strategy f for $G_X^M(\psi_1 \wedge \psi_2)$. Since the opponent 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 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 $G_X^M(\psi_1)$ and $G_X^M(\psi_2)$ respectively, because every play of these games in which 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 $G_X^M(\psi_1 \wedge \psi_2)$ in which 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 $v \in \texttt{Var}$ and $M \models_X \phi$ then there exists a $F : X \to \texttt{Dom}(M)$ such that $M \models_{X[F/v]} \psi$. By induction hypothesis, this implies that E has a uniform winning strategy g for $G_{X[F/v]}^M(\psi)$. Now define the strategy f for E in $G_X^M(\exists v \psi)$ 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 $G_X^M(\phi)$ in which E follows f contains a play of $G_{X[F/v]}^M(\psi)$ in which E follows g, and hence f is uniform and winning.

   Conversely, suppose that E has a uniform winning strategy f for $G_X^M(\exists v \psi)$. Then define the function $F : X \to \texttt{Dom}(M)$ 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 $v \in \texttt{Var}$ and $M \models_X \phi$ then $M \models_{X[M/v]} \psi$. By induction hypothesis, this implies that $\texttt{E}$ has a uniform winning strategy f for $G^M_{X[M/v]}(\psi)$. But then the same f is a uniform winning strategy for $\texttt{E}$ for $G^M_X(\phi)$, since $\texttt{Player}(\phi) = \texttt{A}$ and any play of $G^M_X(\phi)$ in which $\texttt{E}$ follows f contains a play of $G^M_{X[M/v]}(\psi)$ in which $\texttt{E}$ follows f. Conversely, suppose that $\texttt{E}$ has a uniform winning strategy f for $G^M_X(\phi)$. Then the same f is a uniform strategy for $\texttt{E}$ 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 $\texttt{E}$ 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 $\texttt{Dom}(M)$ which is *downwards closed*, in the sense that

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

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

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.

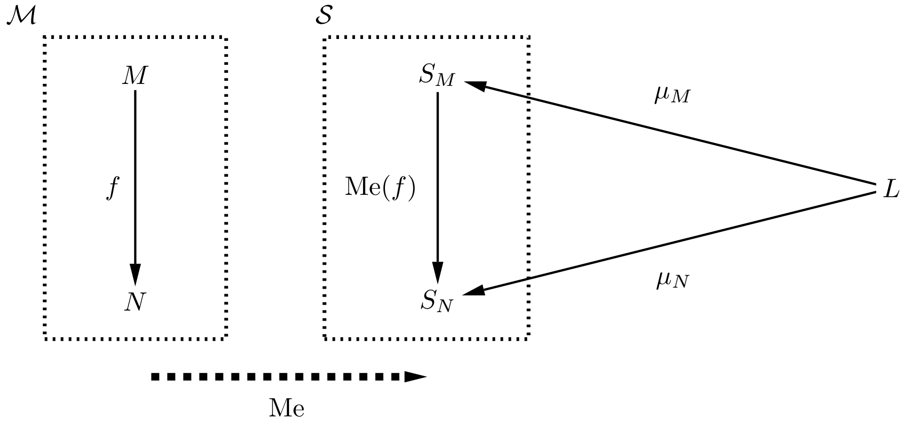

Figure 2.1: Diagram representation of Condition 2 of Definition 2.3.5 (sensible semantics): if $f : M \to N$ is an isomorphism then $\mu_N(\phi) = \text{Me}(f)(\mu_N(\phi))$ for all formulas $\phi \in L$.

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* $\text{Me}(M) = S_M$ *for all $M \in \mathcal{M}$ and*

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

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 $\phi(x_1 \ldots x_k)$ with $FV(\phi) = \{x_1 \ldots x_k\}$ let

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

Then $(\mathcal{S}, \text{Me}, \mu)$ 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* 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 $\phi(x_1 \ldots x_k)$, $M \models_X \phi(x_1 \ldots x_k)$ if and only if $N \models_{f_\uparrow(X)} \phi(x_1 \ldots x_k)$, 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

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

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 $1 \ldots k$ 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 $=\!(t_1 \ldots t_n)$ 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 $=\!(t_1 \ldots t_n)$ can be expressed in Independence Logic as $t_n \perp_{t_1 \ldots t_{n-1}} t_n$; 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 $\psi_1 \rightarrow \psi_2$ and $\psi_1 \multimap \psi_2$ such that

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

and

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

respectively.

The satisfaction conditions corresponding to these two requirements are:

TS-$\rightarrow$: $M \models_X \psi \rightarrow \theta$ 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 $\rightarrow$ operator has been dubbed the "intuitionistic implication" and the $\multimap$ operator has been dubbed the "linear implication", and the languages $\mathcal{D}(\rightarrow)$ 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 $t_1 \ldots t_n$, one can verify that

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

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

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

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.

Chapter 3

Announcement Operators

In this chapter, we will examine and generalize an operator defined by Kontinen and Väänänen in [50]. As we will see, this operator has a natural game-theoretic interpretation in terms of *announcements*. This will be our first hint towards the doxastic interpretation of Team Semantics which will be one of the main themes of this thesis.

3.1 Some Strange Operators

3.1.1 $\exists^1$, $\forall^1$ and δ^1

In [50], the $\exists^1$ and $\forall^1$ quantifiers were defined as

TS-$\exists^1$: $M \models_X \exists^1 x\phi$ if and only if there exists a $m \in \mathtt{Dom}(M)$ such that $M \models_{X[m/x]} \phi$;

TS-$\forall^1$: $M \models_X \forall^1 x\phi$ if and only if for all $m \in \mathtt{Dom}(M)$, $M \models_{X[m/x]} \phi$

where $X[m/x]$ is the team $\{s[m/x] : s \in X\}$.

Kontinen and Väänänen then observed that, for any variable x and formula ϕ, $\exists^1 x\phi$ is equivalent to $\exists x(=(x) \wedge \phi)$; and, furthermore, that – because of Theorem 2.2.14 – adding the $\forall^1$ quantifiers to the language of Dependence Logic does not increase its expressive power, even with respect to open formulas.

However, this left open the problem of whether it is possible to define the $\forall^1$ quantifier in terms of the connectives of Dependence Logic: and at the end of that paper, Kontinen and Väänänen mentioned that

> *It remains open whether the quantifier $\forall^1$ is "uniformly" definable in the logic $\mathcal{D}$.*

We will answer this question later in this chapter, in Subection 3.3.2.

In this subsection, instead, we will begin our study of the $\forall^1 x$ quantifier by introducing a new operator, which will be shown to be strictly related to it:[1]

Definition 3.1.1. For any formula ϕ and variable x, let $\delta^1 x\phi$ be a formula with $\texttt{Free}(\delta^1 x\phi) = \texttt{Free}(\phi) \cup \{x\}$ and such that

TS-δ^1: $M \models_X \delta^1 x\phi$ if and only if for all $m \in \texttt{Dom}(M)$, $M \models_{X(x=m)} \phi$,

where $X(x = m)$ is the team $\{s \in X : s(x) = m\}$.

We write $\mathcal{D}(\delta^1)$ for the logic obtained by adding the δ^1 operator to Dependence Logic.

The following result shows that δ^1 and $\forall^1$ are reciprocally definable:

Proposition 3.1.2. *For any formula ϕ of Dependence Logic, $\forall^1 x\phi \equiv \forall x\delta^1 x\phi$ and $\delta^1 x\phi \equiv \forall^1 y(x \neq y \vee \phi)$, where y is a variable which does not occur in $\texttt{Free}(\phi)$.*

Proof. Let M be any first-order model, let X be any team, and let ϕ be any formula with $y \notin \texttt{Free}(\phi)$. Then, for all X with $\texttt{Free}(\phi) \subseteq \texttt{Dom}(X) \cup \{x\}$,

$$M \models_X \forall x\delta^1 x\phi \Leftrightarrow M \models_{X[M/x]} \delta^1 x\phi \Leftrightarrow \text{ for all } m \in \texttt{Dom}(M), M \models_{X[M/x](x=m)} \phi \Leftrightarrow$$
$$\Leftrightarrow \text{ for all } m \in \texttt{Dom}(M), M \models_{X[m/x]} \phi \Leftrightarrow M \models_X \forall^1 x\phi.$$

and for all X with $\texttt{Free}(\phi) \subseteq \texttt{Dom}(X)$,

$$\begin{aligned} &M \models_X \forall^1 y(x \neq y \vee \phi) \Leftrightarrow \text{ for all } m \in \texttt{Dom}(M), M \models_{X[m/y]} x \neq y \vee \phi \Leftrightarrow \\ &\Leftrightarrow \text{for all } m \in \texttt{Dom}(M)\ \exists\, Y_m, Z_m \text{ such that } X[m/y] = Y_m \cup Z_m, \\ &\qquad \text{if } s \in Y_m \text{ then } s(x) \neq m \text{ and } M \models_{Z_m} \phi \Leftrightarrow \\ &\Leftrightarrow \text{for all } m \in \texttt{Dom}(M), M \models_{X[m/y](x=m)} \phi \Leftrightarrow \\ &\Leftrightarrow \text{for all } m \in \texttt{Dom}(M), M \models_{X(x=m)} \phi \Leftrightarrow M \models_X \delta^1 x\phi \end{aligned}$$

where we used the fact that $y \notin \texttt{Free}(\phi)$. ☐

Since Kontinen and Väänänen proved that the $\forall^1 x$ quantifier does not increase the expressive power of Dependence Logic, from the previous proposition it follows immediately that the $\delta^1 x$ operator does not increase it either.

[1] An interesting application of these δ^1 operators can be found in Dotlačil's paper [16], in which they are used to formalize the distributive interpretation of nominal predicates in the framework of Team Logic.

However, as the next result shows, it is possible to give a simpler, and *constructive*, proof that each formula of $\mathcal{D}(\delta^1)$ can be translated into Dependence Logic:

Proposition 3.1.3. *Let ϕ be any Dependence Logic formula. Then there exists a Dependence Logic formula ϕ^* such that $\phi^* \equiv \delta^1 x\phi$.*

Proof. The proof is a simple structural induction on ϕ:

1. If ϕ is a first order literal then let $\phi^* = \phi$. Indeed,

   $$M \models_X \delta^1 x\phi \Leftrightarrow \forall m \in \mathtt{Dom}(M), M \models_{X(x=m)} \phi \Leftrightarrow$$
   $$\Leftrightarrow \forall m \in \mathtt{Dom}(M), \forall s \in X(x=m), M \models_{\{s\}} \phi \Leftrightarrow \forall s \in X, M \models_{\{s\}} \phi \Leftrightarrow X \models \phi.$$

2. If ϕ is an atomic dependence atom $=\!(t_1 \dots t_n)$, let ϕ^* be $=\!(x, t_1 \dots t_n)$. Indeed,

   $$M \models_X \delta^1 x =\!(t_1 \dots t_n) \Leftrightarrow \forall m \in \mathtt{Dom}(M), M \models_{X(x=m)} =\!(t_1 \dots t_n) \Leftrightarrow$$
   $$\Leftrightarrow \forall m \in \mathtt{Dom}(M), \forall s, s' \in X(x=m), \text{ if } t_i\langle s\rangle = t_i\langle s'\rangle \text{ for } i = 1 \dots n-1 \text{ then } t_n\langle s\rangle = t_n\langle s'\rangle \Leftrightarrow$$
   $$\Leftrightarrow \forall s, s' \in X, \text{ if } s(x) = s'(x), t_i\langle s\rangle = t_i\langle s'\rangle \text{ for } i = 1 \dots n-1 \text{ then } t_n\langle s\rangle = t_n\langle s'\rangle \Leftrightarrow M \models_X =\!(x, t_1 \dots t_n).$$

3. If $\phi = \psi \vee \theta$, let $\phi^* = \psi^* \vee \theta^*$: indeed,

   $$M \models_X \delta^1 x(\psi \vee \theta) \Leftrightarrow \forall m \in \mathtt{Dom}(M), M \models_{X(x=m)} \psi \vee \theta \Leftrightarrow$$
   $$\Leftrightarrow \forall m \in \mathtt{Dom}(M)\ \exists Y_m, Z_m \text{ such that } X(x=m) = Y_m \cup Z_m, M \models_{Y_m} \psi \text{ and } M \models_{Z_m} \theta \Leftrightarrow$$
   $$\Leftrightarrow \exists Y, Z \text{ such that } X = Y \cup Z \text{ and } \forall m \in \mathtt{Dom}(M), M \models_{Y_{x=m}} \psi \text{ and } M \models_{Z_{x=m}} \theta \Leftrightarrow$$
   $$\Leftrightarrow \exists Y, Z \text{ such that } X = Y \cup Z, M \models_Y \delta^1 x\psi \text{ and } M \models_Z \delta^1 x\theta \Leftrightarrow$$
   $$\Leftrightarrow M \models_X \psi^* \vee \theta^*$$

   where for the passage from the second line to the third one we take $Y = \bigcup_{m \in \mathtt{Dom}(M)} Y_m$ and $Z = \bigcup_{m \in \mathtt{Dom}(M)} Z_m$, and for the passage from the third line to the second one we take $Y_m = Y_{x=m}$ and $Z_m = Z_{x=m}$.

4. If $\phi = \psi \wedge \theta$, let $\phi^* = \psi^* \wedge \theta^*$: indeed,

$$M \models_X \delta^1 x(\psi \wedge \theta) \Leftrightarrow \forall m \in \texttt{Dom}(M), M \models_{X(x=m)} \psi \wedge \theta \Leftrightarrow$$
$$\Leftrightarrow \forall m \in \texttt{Dom}(M), M \models_{X(x=m)} \psi \text{ and } M \models_{X(x=m)} \theta \Leftrightarrow$$
$$\Leftrightarrow M \models_X \delta^1 x\psi \text{ and } M \models_X \delta^1 x\theta \Leftrightarrow$$
$$\Leftrightarrow M \models_X \psi^* \wedge \theta^*.$$

5. If $\phi = \exists y\psi$ for some variable $y \neq x$,[2] we let $\phi^* = \exists y\psi^*$: indeed,

$$M \models_X \delta^1 x\exists y\psi \Leftrightarrow \forall m \in \texttt{Dom}(M), M \models_{X(x=m)} \exists y\psi \Leftrightarrow$$
$$\Leftrightarrow \forall m \in \texttt{Dom}(M)\ \exists F_m : X(x=m) \to M \text{ s.t. } M \models_{X(x=m)[F_m/y]} \psi \Leftrightarrow$$
$$\Leftrightarrow \exists F : X \to M \text{ s.t. } \forall m \in \texttt{Dom}(M), M \models_{X[F/y](x=m)} \psi \Leftrightarrow$$
$$\Leftrightarrow \exists F : X \to M \text{ s.t. } M \models_{X[F/y]} \delta^1 x\psi \Leftrightarrow M \models_X \exists y\psi^*$$

where, for the passage from the second line to the third one, we take the function F defined as

$$\forall s \in X, F(s) = F_{s(x)}(s)$$

and, for the passage from the third line to the second one, we take for each F_m the restriction of F to $X(x = m)$.

6. If $\phi = \forall y\psi$ for some variable $y \neq x$, we let $\phi^* = \forall y\psi^*$. Indeed,

$$M \models_X \delta^1 x\forall y\psi \Leftrightarrow$$
$$\Leftrightarrow \forall m \in \texttt{Dom}(M), M \models_{X(x=m)} \forall y\psi \Leftrightarrow \forall m \in \texttt{Dom}(M), M \models_{X(x=m)[M/y]} \psi \Leftrightarrow$$
$$\Leftrightarrow \forall m \in \texttt{Dom}(M), M \models_{X[M/y](x=m)} \psi \Leftrightarrow M \models_{X[M/y]} \delta^1 x\psi \Leftrightarrow M \models_X \forall y\psi^*.$$

□

Therefore, the logics $\mathcal{D}$, $\mathcal{D}(\forall^1)$ and $\mathcal{D}(\delta^1)$ define exactly the same classes of teams over all models and all signatures.

The δ^1 operators – and, hence, the $\forall^1$ quantifiers – are uniformly definable in Intuitionistic Dependence Logic: indeed, for all formulas ϕ, models M and

[2] If $y = x$, we define $(\exists x\psi)^* := (\exists z\psi[z/x])^*$ and $(\forall x\psi)^* := (\forall z\psi[z/x])^*$ for some new variable z.

all teams X it holds that

$$M \models_X =\!(x) \to \phi \Leftrightarrow \text{ for all } Y \subseteq X, \text{ if } M \models_X =\!(x) \text{ then } M \models_X \phi \Leftrightarrow$$
$$\Leftrightarrow \text{ for all } m \in \texttt{Dom}(M), M \models_{X(x=m)} \phi \Leftrightarrow M \models_X \delta^1_x \phi.$$

This suggests that, as for the case of intuitionistic implication, the δ^1 operators may be used to reduce dependency atoms $=\!(t_1 \ldots t_n)$ to constancy atoms $=\!(t_i)$. This is indeed the case:

Proposition 3.1.4. *Let $x_1 \ldots x_n$ be variables. Then*

$$=\!(x_1 \ldots x_n) \;\equiv\; \delta^1 x_1 \ldots \delta^1 x_{n-1} =\!(x_n).$$

Proof.

$$M \models_X \delta^1 x_1 \ldots \delta^1 x_{n-1} =\!(x_n) \Leftrightarrow$$
$$\Leftrightarrow \forall m_1 \ldots m_{n-1} \in \texttt{Dom}(M), M \models_{X(x_1=m_1 \ldots x_{n-1}=m_{n-1})} =\!(x_n) \Leftrightarrow$$
$$\Leftrightarrow \forall m_1 \ldots m_{n-1} \in \texttt{Dom}(M), s, s' \in X, \text{ if } s(x_1) = s'(x_1) = m_1, \ldots$$
$$\ldots, s(x_{n-1}) = s'(x_{n-1}) = m_{n-1} \text{ then } s(x_n) = s'(x_n) \Leftrightarrow$$
$$\Leftrightarrow M \models_X =\!(x_1 \ldots x_n).$$

□

In the same way, one may decompose dependency atoms of the form $=\!(t_1 \ldots t_n)$ as

$$\exists x_1 \ldots x_{n-1} \left(\left(\bigwedge_{i=1}^{n-1} x_i = t_i \right) \wedge \delta^1 x_1 \ldots \delta^1 x_{n-1} =\!(t_n) \right)$$

or introduce operators $\delta^1(t)$ with the obvious semantics; hence, by removing non-constant dependency atoms from Dependence Logic and adding the δ^1 operators, one may obtain a formalism with the same expressive power of Dependence Logic, but in which *constancy* takes the place of *functional dependency.*

3.1.2 $\forall^\kappa$ and δ^κ

$\forall^1$ and δ^1 can be seen as representatives of a proper class of operators $\{\exists^\kappa, \forall^\kappa, \delta^\kappa : \kappa \in \texttt{Card}\}$:

Definition 3.1.5. For any (finite or infinite) cardinal κ, for every formula ϕ and for every variable x, let $\delta^\kappa x \phi$ and $\forall^\kappa x \phi$ be formulas with $\texttt{Free}(\delta^\kappa x \phi) =$

$\texttt{Free}(\phi) \cup \{x\}$, $\texttt{Free}(\forall^\kappa x\phi) = \texttt{Free}(\phi)\backslash\{x\}$, and satisfaction conditions

TS-$\forall^\kappa$: $M \models_X \forall^\kappa x\phi$ if and only if for all $A \subseteq \texttt{Dom}(M)$ such that $|A| \leq \kappa$, $M \models_{X[A/x]} \phi$.

TS-δ^κ: $M \models_X \delta^\kappa x\phi$ if and only if for all $A \subseteq \texttt{Dom}(M)$ such that $|A| \leq \kappa$, $M \models_{X(x\in A)} \phi$;

where $X(x \in A) = \{s \in X : s(x) \in A\}$ and $X[A/x] = \{s[m/x] : s \in X, m \in A\}$.

Again, we can define uniformly δ^κ by means of $\forall^\kappa$ and vice versa:

Proposition 3.1.6. *For all cardinals κ, formulas $\phi \in \mathcal{D}$, variables x and teams X,*

$$\forall^\kappa x\phi \equiv \forall x\delta^\kappa x\phi.$$

Furthermore, if $y \notin \texttt{Free}(\phi)$ then

$$\delta^\kappa x\phi \equiv \forall^\kappa y(y \neq x \vee \phi).$$

Proof.

$$M \models_X \forall x\delta^\kappa x\phi \Leftrightarrow M \models_{X[M/x]} \delta^\kappa x \Leftrightarrow \forall A \subseteq^\kappa \texttt{Dom}(M), M \models_{X[M/x](x\in A)} \phi \Leftrightarrow$$
$$\Leftrightarrow \forall A \subseteq^\kappa \texttt{Dom}(M), M \models_{X[A/x]} \Leftrightarrow M \models_X \forall^\kappa x\phi$$

and

$$M \models_X \forall^\kappa y(y \neq x \vee \phi) \Leftrightarrow \forall A \subseteq^\kappa \texttt{Dom}(M), M \models_{X[A/y]} (y \neq x \vee \phi) \Leftrightarrow$$
$$\Leftrightarrow \forall A \subseteq^\kappa \texttt{Dom}(M), M \models_{X[A/y](y=x)} \phi \Leftrightarrow \forall A \subseteq^\kappa \texttt{Dom}(M), M \models_{X(x\in A)} \phi \Leftrightarrow$$
$$\Leftrightarrow M \models_X \delta^\kappa x\phi,$$

where we used the fact that $y \notin \texttt{Free}(\phi)$ and we used $A \subseteq^\kappa \texttt{Dom}(M)$ as a shorthand for "$A \subseteq \texttt{Dom}(M)$ and $|A| \leq \kappa$". □

For every $n \in \mathbb{N}_0$, the quantifier $\forall^n$ can be uniformly defined in terms of $\forall^1$ and the operator δ^1 can be uniformly defined in terms of δ^n:

Proposition 3.1.7. *For every $n \in \mathbb{N}_0$ and for all $\phi \in \mathcal{D}$ such that $\texttt{Free}(\phi) \cap \{x_1 \dots x_n\} = \emptyset$,*

$$\forall^n x\phi \equiv \forall^1 x_1 \dots \forall^1 x_n \forall x \left(\bigwedge_{i=1}^{n} (x \neq x_i) \vee \phi \right).$$

Proof.

$$M \models_X \forall^1 x_1 \dots \forall^1 x_n \forall x \left(\bigwedge_{i=1}^{n} (x \neq x_i) \vee \phi \right) \Leftrightarrow$$
$$\Leftrightarrow \forall \overline{m} \in \mathtt{Dom}(M)^n, M \models_{X[M/x][\overline{m}/\overline{x}]} \bigwedge_{i=1}^{n} (x \neq x_i) \vee \phi \Leftrightarrow$$
$$\Leftrightarrow \forall m_1 \dots m_n \in \mathtt{Dom}(M), M \models_{X[M/x](x \in \{m_1 \dots m_n\})} \phi \Leftrightarrow$$
$$\Leftrightarrow \forall A \subseteq^n B, M \models_{X[A/x]} \phi \Leftrightarrow M \models_X \forall^n x \phi.$$

□

Proposition 3.1.8. *For every $n \in \mathbb{N}_0$ and for every formula $\phi \in \mathcal{D}$,*

$$\delta^1 x \phi \equiv \delta^n x(\underbrace{(=(x) \wedge \phi) \vee \dots \vee (=(x) \wedge \phi)}_{n \text{ times}}).$$

Proof.

$$M \models_X \delta^n x(\underbrace{(=(x) \wedge \phi) \vee \dots \vee (=(x) \wedge \phi)}_{n \text{ times}}) \Leftrightarrow$$
$$\Leftrightarrow \forall A \subseteq^n \mathtt{Dom}(M), M \models_{X(x \in A)} (\underbrace{(=(x) \wedge \phi) \vee \dots \vee (=(x) \wedge \phi)}_{n \text{ times}}) \Leftrightarrow$$
$$\Leftrightarrow \forall A \subseteq^n \mathtt{Dom}(M), X(x \in A) = X_1 \cup \dots \cup X_n \text{ s.t. } M \models_{X_i} =(x) \wedge \phi \text{ for } i = 1 \dots n \Leftrightarrow$$
$$\Leftrightarrow \forall m_1 \dots m_n \in \mathtt{Dom}(M), M \models_{X(x=m_i)} \phi \text{ for all } i = 1 \dots n \Leftrightarrow M \models_X \delta^1 x \phi$$

□

However, this changes if we consider operators of the form $\forall^\kappa x \phi$, where κ is an infinite cardinal:

Proposition 3.1.9. *For any infinite κ, $\mathcal{D}(\forall^\kappa)$ is strictly more expressive than $\mathcal{D}$.*

Proof. For every model M, $M \models_{\{\emptyset\}} \forall^\kappa x \exists y (=(y) \wedge x \neq y)$ if and only if $|M| > \kappa$.

But $\mathcal{D}$ and all logics semantically equivalent to it satisfy the Löwenheim-Skolem Theorem (Theorem 2.2.12 in this work), and therefore the above formula is not expressible in Dependence Logic. □

3.2 A Game Theoretic Semantics for Announcement Operators

As we saw in Subsection 2.2.3, Dependence Logic has also a Game Theoretic Semantics, which is equivalent to its usual Team Semantics in the sense that a team X satisfies a formula ϕ if and only if the existential player $\mathtt{E}$ has a uniform winning strategy for the game $G^M_X(\phi)$.

In this section, we will adapt this Game Theoretic Semantics to the case of the δ^κ operators; and as we will see, this will allow us to find a natural interpretation of these operators in terms of *public announcements*.

3.2.1 Game Theoretic Semantics for $\mathcal{D}(\delta^1)$

In this subsection, we will extend the Game Theoretic Semantics for Dependence Logic to $\mathcal{D}(\delta^1)$.

Definition 3.2.1. Let ϕ be a formula in $\mathcal{D}(\delta^1)$. Then $\mathtt{Player}(\phi)$ and $\mathtt{Succ}_M(\phi)$ are defined precisely as in Definitions 2.2.17 and 2.2.18 respectively, with the additional conditions that if ϕ is of the form $\delta^1 x\psi$ for some variable x and some formula ψ then $\mathtt{Player}(\phi) = \mathtt{A}$ and $\mathtt{Succ}_M(\phi, s) = \{(\psi, s)\}$.

As we mentioned, the $\delta^1 x$ operator will be interpreted as a *public announcement* of the value of the variable x. Such an announcement corresponds to a *weakening* of the uniformity condition for our games: more precisely, as the proof of Proposition 3.1.3 illustrates, after such an announcement $\mathtt{E}$'s strategy may depend on the value of x even where dependence atoms forbid it.

More formally, we will weaken Definition 2.2.21 in the following way:

Definition 3.2.2. Let $G^M_X(\phi)$ be a game for $\phi \in \mathcal{D}(\delta^1)$, and let $\vec{p}$ and $\vec{q}$ be two plays of it. Then $\vec{p}$ and $\vec{q}$ are δ^1-*similar* if and only if for all $i, j \in \mathbb{N}$ such that $p_i = (\delta^1 x\psi, s)$ and $q_j = (\delta^1 x\psi, s')$ for the same instance of $\delta^1 x\psi$ it holds that $s(x) = s'(x)$.

Definition 3.2.3. Let $G^M_X(\phi)$ be a game for $\phi \in \mathcal{D}(\delta^1)$, and let P be a set of plays of it. Then P is *uniform* if and only if for all δ^1-similar plays $\vec{p}, \vec{q} \in P$ of the same length n and for all $i, j \in \mathbb{N}$ $p_i = (=(t_1 \ldots t_n), s)$ and $q_j = (=(t_1 \ldots t_n), s')$ for the same instance of $=(t_1 \ldots t_n)$ it holds that

$$(t_1 \ldots t_{n-1})\langle s'_1\rangle = (t_1 \ldots t_{n-1})\langle s'_2\rangle \Rightarrow t_n\langle s'_1\rangle = t_n\langle s'_2\rangle.$$

As before, a strategy f is uniform if and only if the set of all plays in which $\mathtt{E}$ follows f is uniform.

Apart from these minor modifications, our semantic games are defined precisely as in Subsection 2.2.3.

All that is left for us to do is to prove the equivalence between this Game Theoretic Semantics and the Team Semantics for $\mathcal{D}(\delta^1)$:

Theorem 3.2.4. *Let M be a first-order model, let X be a team, and let ϕ be any $\mathcal{D}(\delta^1)$ formula. Then $M \models_X \phi$ if and only if the existential player* E *has a uniform winning strategy for $G^M_X(\phi)$.*

Proof. The proof is by structural induction over ϕ. All cases except δ^1 are treated precisely as in Theorem 2.2.28, so we will only describe how to deal with this new one.

Suppose that $M \models_X \delta^1 x\psi$. Then, by definition, for all $m \in \mathtt{Dom}(M)$ we have that $M \models_{X(x=m)} \psi$. By induction hypothesis, this implies that for every $m \in M$ there exists an uniform winning strategy f^m for E in $G^M_{X(x=m)}(\psi)$; and furthermore, without loss of generality, we can assume that if ψ contains a subformula of the form $\forall x\theta$ or $\exists x\theta$ for the same θ then all f^m behave exactly in the same way over the subgame corresponding to this subformula.[3]

Then define the strategy f for Player II in $G^M_X(\delta^1 x\psi)$ as[4]

- For all subformulas θ of ψ and all assignments s, $f_\theta(s) = f^{s(x)}_\theta(s)$.

This strategy is winning, since any complete play $\vec{p} = p_1 \ldots p_n$ of $G^M_X(\delta^1 x\psi)$ in which E follows f contains, for some $m \in \mathtt{Dom}(M)$, a play of $G^M_{X(x=m)}(\psi)$ in which E follows f^m; and it is also uniform, since any two δ^1-similar complete plays assign the same value to the variable x, and hence are played according to the same f^m, and hence respect the uniformity condition.

Conversely, suppose that E has a uniform winning strategy f for $G^M_X(\delta^1 x\psi)$, let $m \in M$, and define the strategy f^m for $G^M_{X(x=m)}(\psi)$ as the restriction of f to the positions of this game.

Then each f^m is a winning strategy for $G^M_{X(x=m)}(\psi)$, because f itself is winning and each complete play of $G^M_{X(x=m)}(\psi)$ in which E follows f^m is included in a complete play for $G^M_X(\delta^1 x\psi)$ in which E follows f; and furthermore, it is uniform, because any two plays $\vec{p}_0$ and $\vec{q}_0$ of $G^M_{X(x=m)}(\psi)$ in which E follows f^m are included in two complete plays $\vec{p}$ and $\vec{q}$ of $G^M_X(\delta^1 x\psi)$ in which E follows f and in which the initial positions assign the value m to x. Hence, if $\vec{p}_0$ and $\vec{q}_0$ are δ^1-similar over $G^M_{X(x=m)}(\psi)$ then $\vec{p}$ and $\vec{q}$ are δ^1-similar over $G^M_X(\delta^1 x\psi)$, and in conclusion they satisfy the uniformity condition.

[3] In brief, this follows from the fact that Locality (Proposition 2.2.8) also holds for $\mathcal{D}(\delta^1)$, and hence the value of the variable x is entirely irrelevant as far as the winning conditions of $\forall x\theta$ or $\exists x\theta$ are concerned.

[4] Since $\mathtt{Player}(\delta^1 x\psi) = \mathtt{A}$, there is no need to specify a successor for the initial position.

Thus, by induction hypothesis, $M \models_{X(x=m)} \psi$ for all $m \in \texttt{Dom}(M)$, and therefore $M \models_X \delta^1 x\psi$. □

Because of this Game Theoretic Semantics, δ^1 may be called a "announcement operator": anthropomorphizing somewhat the two agents of the game, one can think of $\delta^1 x\phi$ as the subgame in which first the value of x is announced from A to E and then the game corresponding to ϕ is played.

3.2.2 Game Theoretic Semantics for δ^κ

In order to generalize the above approach to δ^κ operators for $\kappa > 1$, we need to make a few additional modifications to our games. As before, we will set $\texttt{Player}(\delta^\kappa x, s) = \texttt{A}$; but now, game positions will not be pairs (ψ, s) but *triples* $(\psi, s, \vec{\rho})$ where $\vec{\rho}$ is a *annotation sequence* representing the public announcements made by A:

Definition 3.2.5. Let M be a first-order model and let v be a variable. An *annotation* ρ for v is an expression $(v \in A)$ where $A \subseteq \texttt{Dom}(M)$. An *annotation sequence* $\vec{\rho}$ is simply a tuple of annotations $\rho_1 \ldots \rho_n$.

The initial positions of a game $G^M_X(\phi)$ will be of the form $(\phi, s, \emptyset)$ for $s \in X$ where $\emptyset$ represents the empty annotation sequence. The winning positions will be those of the form $(\lambda, s, \vec{\rho})$ for any assignment s and annotation sequence $\vec{\rho}$. Furthermore, the set of the *successors* of a given position will be given as follows:

Definition 3.2.6. Let M be a first order model, let ψ be a formula of $\mathcal{D}(\delta^\kappa)$, let s be an assignment over M and let $\vec{\rho}$ be an annotation. Then the set $\texttt{Succ}_M(\psi, s, \vec{\rho})$ of the *successors* of the position (ψ, s, A) is defined as follows:

1. If ψ is a first order literal κ then

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

   where λ stands for the empty string;

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

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

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

5. If ψ is of the form $\delta^\kappa x\theta$ then $\texttt{Succ}_M(\psi, s, \vec{\rho}) = \{(\theta, s, \vec{\rho}(x \in A) : s(x) \in A \subseteq \texttt{Dom}(M), |A| \leq \kappa\}$.

Again, the definition of uniform strategy requires some modifications:

Definition 3.2.7. Let $G^M_X(\phi)$ be a game for $\phi \in \mathcal{D}(\delta^\kappa)$, and let $\vec{p}$ and $\vec{q}$ be two plays of it. Then $\vec{p}$ and $\vec{q}$ are δ^κ-*similar* if and only if for all $i, j \in \mathbb{N}$ such that $p_i = (\psi, s, \vec{\rho})$ and $q_j = (\psi, s', \vec{\nu})$ for the same instance of ψ then $\vec{\rho} = \vec{\nu}$.

Definition 3.2.8. Let $G^M_X(\phi)$ be a game for $\phi \in \mathcal{D}(\delta^\kappa)$, and let P be a set of plays of it. Then P is *uniform* if and only if for all δ^κ-similar plays $\vec{p}, \vec{q} \in P$ of the same length n and for all $i, j \in \mathbb{N}$ $p_i = (=(t_1 \ldots t_n), s, \vec{\rho})$ and $q_j = (=(t_1 \ldots t_n), s', \vec{\nu})$ for the same instance of $=(t_1 \ldots t_n)$ it holds that

$$(t_1 \ldots t_{n-1})\langle s'_1\rangle = (t_1 \ldots t_{n-1})\langle s'_2\rangle \Rightarrow t_n\langle s'_1\rangle = t_n\langle s'_2\rangle.$$

As usual, a strategy f is uniform if and only if the set of all plays in which $\texttt{E}$ follows f is uniform.

Once again, we can prove that this Game Theoretic Semantics is equivalent to the Team Semantics for $\mathcal{D}(\delta^\kappa)$:

Theorem 3.2.9. *Let M be a first-order model, let X be a team, and let ϕ be any $\mathcal{D}(\delta^1)$ formula. Then $M \models_X \phi$ if and only if the existential player $\texttt{E}$ has a uniform winning strategy for $G^M_X(\phi)$.*

Proof. We only describe how to deal with the new case.

Suppose that $M \models_X \delta^\kappa x\psi$. Then, by definition, we have that for all subsets $A \subseteq^\kappa \texttt{Dom}(M)$ it holds that $M \models_{X(x\in A)} \psi$. Therefore, by induction hypothesis, for each such A there exists a uniform winning strategy f^A for $\texttt{E}$ in $G^M_{X(x\in A)}(\psi)$. Let us define the strategy f for $\texttt{E}$ in $G^M_X(\delta^\kappa x\phi)$ in such a way that

- $f_\theta(s, (x \in A)\vec{\rho}) = f^A_\theta(s, \vec{\rho})$

for all subformulas θ, all assignments s, all $A \subseteq^\kappa \texttt{Dom}(M)$ and all annotation sequences $\vec{\rho}$.

This strategy is winning, because each play of $G^M_X(\delta^\kappa x\psi)$ in which $\texttt{E}$ follows it contains, for some $A \subseteq^\kappa \texttt{Dom}(M)$, a play of $G^M_{X(x\in A)}(\psi)$ in which $\texttt{E}$ follows f^A; and it is uniform, as any two δ^κ-similar plays must have the same initial annotation for x and therefore must be played according to the same f^A, which we know by hypothesis to be uniform.

Conversely, suppose that E has a uniform winning strategy for $G^M_X(\delta^\kappa x\psi)$: then for each $A \subseteq^\kappa \texttt{Dom}(M)$ such that $\{s(x) : s \in X\} \cap A \neq \emptyset$ E has a uniform winning strategy for $G^M_{X(x\in A)}(\psi)$, and hence by induction hypothesis $M \models_{X(x\in A)} \psi$ for all such A. If instead $|A| \leq \kappa$ and $\{s(x) : s \in X\} \cap A = \emptyset$ then $M \models_{X(x\in A)} \psi$ trivially, and therefore $M \models_{X(x\in A)} \psi$ for all A with $|A| \leq \kappa$.

So, in conclusion, $M \models_X \delta^\kappa x\psi$, as required. □

Again, the intuition is that of an announcement, but this time it is a *partial* one: in the subgame corresponding to $\delta^\kappa x\psi$, A does not allow E full access to the value of x, but he chooses a set A of cardinality (at most) κ and gives her the (true) information that $x \in A$.

3.3 Some Properties of Public Announcements

In this section, we will examine a little more in-depth the properties of Dependence Logic augmented with the announcement operators. First, we will adapt to this framework the Ehrenfeucht-Fraïssé games for Dependence Logic developed by Jouko Väänänen ([65]); then we will prove that the announcement operator δ^1 and the $\forall^1$ quantifier are not uniformly definable in Dependence Logic, thus solving an open problem mentioned at the end of [50].

3.3.1 An Ehrenfeucht-Fraïssé game for $\mathcal{D}(\sqcup, \forall^\kappa)$

In ([65], §6.6), the following *semiequivalence* relation between models and teams was introduced:

Definition 3.3.1. Let M, N be two models, and let X, Y be teams over M and N respectively. Then $(M, X) \Rrightarrow (N, Y)$ if and only if

$$M \models_X \phi \Rightarrow N \models_Y \phi$$

for all Dependence Logic formulas ϕ.

In this section, I will adapt the Ehrenfeucht-Fraïssé game for Dependence Logic to $\mathcal{D}(\sqcup, \forall^\kappa)$, where $\sqcup$ is the *classical disjunction* of Definition 2.2.4 which, for the purposes of this section, will be taken as a basic connective of our language.

The following definitions are the obvious modifications of those of ([65], §6.6):

Definition 3.3.2. Let $\phi \in \mathcal{D}(\sqcup, \forall^\kappa)$. Then its rank $\mathrm{qr}(\phi)$ is defined inductively as follows:

- If ϕ is a literal, $\mathrm{qr}(\phi) = 0$;
- $\mathrm{qr}(\phi \vee \psi) = \max(\mathrm{qr}(\phi), \mathrm{qr}(\psi)) + 1$;
- $\mathrm{qr}(\phi \wedge \psi) = \max(\mathrm{qr}(\phi), \mathrm{qr}(\psi))$;
- $\mathrm{qr}(\phi \sqcup \psi) = \max(\mathrm{qr}(\phi), \mathrm{qr}(\psi))$;
- $\mathrm{qr}(\exists x \psi) = \mathrm{qr}(\psi) + 1$;
- $\mathrm{qr}(\forall x \psi) = \mathrm{qr}(\psi) + 1$;
- $\mathrm{qr}(\forall^\kappa x \psi) = \mathrm{qr}(\psi) + 1$.

Definition 3.3.3.

$$\mathcal{D}_n(\sqcup, \forall^\kappa) = \{\phi : \phi \text{ is a formula of } \mathcal{D}(\sqcup, \forall^\kappa) \text{ and } \mathrm{qr}(\phi) \leq n\}.$$

Definition 3.3.4. Let M, N be two models, and let X, Y be teams over M and N respectively. Then $(M, X) \Rrightarrow^\kappa (N, Y)$ if and only if

$$M \models_X \phi \Rightarrow N \models_Y \phi$$

for all formulas $\phi \in \mathcal{D}(\sqcup, \forall^\kappa)$.

Definition 3.3.5. Let M, N be two models, and let X, Y be teams over M and N respectively. Then $(M, X) \Rrightarrow^\kappa_n (N, Y)$ if and only if

$$M \models_X \phi \Rightarrow N \models_Y \phi$$

for all formulas $\phi \in \mathcal{D}_n(\forall^\kappa, \sqcup)$.

Lemma 3.3.6. *Let M, N, X and Y be as above. Then $(M, X) \Rrightarrow^\kappa (N, Y)$ if and only if $(M, X) \Rrightarrow^\kappa_n (N, Y)$ for all $n \in \mathbb{N}$.*

The following proposition is proved analogously to the corresponding result for Dependence Logic ([65], Proposition 6.48):

Proposition 3.3.7. *A class K of models (over the same, finite, signature Σ) with teams in a fixed domain V is definable in $\mathcal{D}_n(\sqcup, \delta^\kappa)$ if and only if it is closed under $\Rrightarrow^\kappa_n$.*

Proof. Suppose that K is $\{(M, X) : \mathtt{Dom}(X) = V, M \models_X \phi\}$ for some formula $\phi \in \mathcal{D}_n(\sqcup, \delta^\kappa)$. If $(M, X) \in K$ and $(M, X) \Rrightarrow^\kappa_n (N, Y)$ then $N \models_Y \phi$ too and hence $(N, Y) \in K$, and therefore K is closed under the $\Rrightarrow^\kappa_n$ relation.

Conversely, suppose that K is closed under the $\Rrightarrow^\kappa$ relation: then for every model $(M,X) \in K$ and for every $(N,Y) \not\in K$ there exists a formula $\phi_{MX,NY}$ of rank $\leq n$ such that $M \models_X \phi_{MX,NY}$ but $N \not\models_Y \phi_{MX,NY}$. Then consider the formula

$$\phi := \bigsqcup_{(M,X)\in K} \bigwedge_{(N,Y)\not\in K} \phi_{MX,NY}.$$

As there exist only finitely many logically different formulas

$$\psi \in \mathcal{D}_n(\sqcup, \delta^\kappa)$$

with $\texttt{Free}(\psi) \subseteq V$, the conjunction and the classical disjunction in ϕ are finite and $\phi \in \mathcal{D}_n(\sqcup, \delta^\kappa)$.

Furthermore, $K = \{(M,X) : M \models_X \phi\}$. Indeed, if $(M,X) \in K$ then for all $(N,Y) \not\in K$ it holds that $M \models_X \phi_{MX,NY}$, and if $(N,Y) \not\in K$ then $N \not\models_Y \phi_{MX,NY}$ for any $(M,X) \in K$. □

Then, for $n \in \mathbb{N}$, we can define the $EF_n^\kappa(M,X,N,Y)$ game as follows:

Definition 3.3.8. Let M, N be two models, let X, Y be teams with the same domain over M and N respectively, let κ be any (finite or infinite) cardinal and let $n \in \mathbb{N}$. Then the game $EF_n^\kappa(M,X,N,Y)$ is defined as follows:

- There are two players, which we will once again call `A` (Abelard) and `E` (Eloise);
- $x_1 \ldots x_n$ are variables which do not occur in $\texttt{Dom}(X) = \texttt{Dom}(Y)$.
- The set P of all positions of the game is

  $$\{(X^i, Y^i, i) : X^i \text{ is a team on } M, Y^i \text{ is a team on } N \text{ and } i \in 0 \ldots n\};$$

- The starting position is $(X, Y, 0)$;
- For each position (X^i, Y^i, i) with $i < n$, `A` decides which kind of move to play among the following possibilities:

  Splitting: `A` chooses teams X' and X'' with $X' \cup X'' = X^i$. Then `E` chooses teams Y' and Y'' with $Y' \cup Y'' = Y^i$, and `A` decides whether the next position is $(X', Y', i+1)$ or $(X'', Y'', i+1)$;

Supplementation: A chooses a function $F : X^i \to \texttt{Dom}(M)$. Then E chooses a function $G : Y^i \to \texttt{Dom}(N)$, and the next position is

$$(X^i[F/x_i], Y^i[G/x_i], i+1);$$

Duplication: The next position is $(X^i[M/x_i], Y^i[N/x_i], i+1)$;

Right-κ-duplication: A chooses a set of elements $B \subseteq^\kappa \texttt{Dom}(N)$. Then E chooses a set of elements $A \subseteq^\kappa \texttt{Dom}(M)$, and the next position is

$$(X^i[A/x_i], Y^i[B/x_i], i+1).$$

- The set W of all winning positions for E is

$$\{(X_n, Y_n, n) : (M, X_n) \Rrightarrow_0^\kappa (N, Y_n)\}.$$

The concepts of play, complete play, strategy and winning strategy are defined in the obvious way, and there is no uniformity condition for this game.

Theorem 3.3.9. *Let M, N, X and Y as above, and let $n \in \mathbb{N}$. Then $(M, X) \Rrightarrow_n^\kappa (N, Y)$ if and only if Player $\exists$ has a winning strategy for $EF_n^\kappa(M, X, N, Y)$.*

Proof. The left to right direction is proved by induction over n, and by considering all possible first moves of A. As the only new case compared to ([65], Theorem 6.44) is the right-κ-duplication, we will only take care of this one:

- Suppose that $(M, X) \Rrightarrow_n^\kappa (N, Y)$, and let Player A make a right-κ-duplication move and choose a set $B \subseteq^\kappa \texttt{Dom}(N)$. Then there exists a set $A \subseteq^\kappa \texttt{Dom}(M)$ such that $(M, X[A/x_i]) \Rrightarrow_{n-1}^\kappa (N, Y[B/x_i])$: indeed, suppose instead that for each such set A there exists a formula ϕ^A of rank $\leq n-1$ such that $M \models_{X[A/x_i]} \phi^A$ but $N \not\models_{Y[B/x_i]} \phi^A$, and consider

$$\phi = \bigsqcup_{A \subseteq M, |A| \leq \kappa} \phi^A.$$

Then we would have that $\text{qr}(\forall^\kappa x_i \phi) \leq n$ and $M \models_X \forall^\kappa x_i \phi$; but since $(M, X) \Rrightarrow_n^\kappa (N, Y)$ this implies that $M' \models_Y \forall^\kappa x_i \phi$, and thus in particular $M' \models_{X[B/x_i]} \phi$ and hence $N \models_{X[B/x_i]} \phi^A$ for some A. But this is not possible, and therefore there exists an A_0 such that $(M, X[A_0/x_i]) \Rrightarrow_{n-1}^\kappa (N, Y[B/x_i])$. By induction hypothesis, this implies that Player E has a winning strategy in $EF_{n-1}^\kappa(M, X[A_0/x_i], N, Y[B/x_i])$, and thus she can win the current play by choosing this A_0 and then playing according to this winning strategy.

For the right to left direction, we assume that Player E has a winning strategy in $EF_n^\kappa(M, X, N, Y)$ and we prove, by structural induction on ϕ, that if $\text{qr}(\phi) \leq n$ and $M \models_X \phi$ then $N \models_Y \phi$ too. The only cases which we will consider will be the new ones, corresponding to classical disjunction and $\forall^\kappa$ quantification; for the others, once again, we refer to ([65], Theorem 6.44).

- Suppose that ϕ is of the form $\psi \sqcup \theta$, where $\text{qr}(\phi) = \max(\text{qr}(\psi), \text{qr}(\theta)) \leq n$ and $M \models_X \phi$. Then by the definition of the classical disjunction, $M \models_X \psi$ or $M \models_X \theta$: let us assume, without loss of generality, that $M \models_X \psi$. Then, by our induction hypothesis[5], $N \models_Y \psi$, and hence $N \models_Y \psi \sqcup \theta$ too.
- If ϕ is of the form $\forall^\kappa x_i \psi$ and $M \models_X \phi$ then, by definition, for all subsets $A \subseteq \texttt{Dom}(M)$ such that $|A| \leq \kappa$ we have that $M \models_{X[A/x_i]} \psi$. Suppose now that for some subset $B_0 \subseteq \texttt{Dom}(N)$ such that $|B_0| \leq \kappa$, $N \not\models_{X[B_0/x_i]} \psi$: then, as $\text{qr}(\psi) \leq n-1$ and by induction hypothesis, Player A has a winning strategy in $EF_{n-1}^\kappa(M, X[A/x_i], N, Y[B_0/x_i])$ for all sets A as above.[6] But then A can win $EF_n^\kappa(M, X[A/x_i], N, Y[B_0/x_i])$ by selecting this B_0 and playing the strategy corresponding to the A picked in answer by Player E. This contradicts our assumption: therefore, there is no such B_0 and for all $B \subseteq^\kappa N$ it holds that $N \models_{Y[B/x_i]} \psi$, so in conclusion $B \models \forall^\kappa \psi$, as required.

□

One may wonder if there exists an Ehrenfeucht-Fraïssé game for $\mathcal{D}(\sqcup, \delta^\kappa)$. It turns out that such a game exists, and it is obtained simply changing the right-κ-duplication of $EF_n^\kappa(M, X, N, Y)$ into the following *right-κ-selection* rule:

Right-κ-selection: E chooses a variable $x \in \texttt{Dom}(X) = \texttt{Dom}(Y)$ and a set of elements $B \subseteq N$ such that $|B| \leq \kappa$. Then E chooses a set of elements $A \subseteq \texttt{Dom}(M)$ with $|A| \leq \kappa$, and the next position is $(X^i_{x_i \in A}, Y^i_{x_i \in B}, i+1)$.

The proof that this rule captures correctly the δ^κ connective is entirely analogous to the previous one.

---

[5]As here we are working by *structural induction* on ϕ rather than by induction on $\text{qr}(\phi)$, the fact that $\text{qr}(\psi)$ is not necessarily smaller than $\text{qr}(\phi)$ is not an issue.

[6]As the EF games are finite games of perfect information which do not allow for draws, by Zermelo's Theorem ([76]) one of the two players has a winning strategy in $EF_{n-1}(M, X[A/x_i], N, Y[B/x_i])$. As $(M, X[A/x_i]) \not\Rightarrow^U_{n-1} (N, Y[B_0/x_i])$, by induction hypothesis Player E does not have a winning strategy. Hence, Player A does.

One may also wonder which connectives correspond to the *left-κ-duplications* and *left-κ-selection* rules:

Left-κ-duplication: `A` chooses a set of elements $A \subseteq M$ such that $|A| \leq \kappa$. Then `E` chooses a set of elements $B \subseteq N$ with $|B| \leq \kappa$, and the next position is $(X^i[A/x_i], Y^i[B/x_i], i+1)$.

Left-κ-selection: `A` chooses a variable $x \in \mathtt{Dom}(X) = \mathtt{Dom}(Y)$ and a set of elements $A \subseteq M$ such that $|A| \leq \kappa$. Then `E` chooses a set of elements $B \subseteq N$ with $|B| \leq \kappa$, and the next position is $(X^i_{x_i \in A}, Y^i_{x_i \in B}, i+1)$.

However, these rules do not correspond to anything interesting: indeed, if `A` uses them then, since $(M, X) \Rrightarrow^\kappa_n (N, \emptyset)$ for all n, M and X, Player `E` can always win the play choosing $B = \emptyset$.

3.3.2 Uniform Definability

In this subsection, we will clarify and answer (negatively) the problem of the uniform definability of the $\forall^1$ quantifier which we mentioned in Subsection 3.1.1.

Definition 3.3.10. Let Σ be a signature and let $n \in \mathbb{N}$. Then a *context* of signature Σ is a Dependence Logic formula $\Psi[\Xi]$ and type[7] $\langle n \rangle$ over the signature $\Sigma \cup \{\Xi\}$, where Ξ is a new n-ary relation symbol which occurs only positively in Ψ.

Given a context $\Phi[\Xi]$ and a formula ψ, $\Phi[\psi]$ is defined in the obvious way:

Definition 3.3.11. Let $\Phi[\Xi]$ be a context of signature Σ and type $\langle n \rangle$, let $\psi(x_1 \ldots x_n)$ be a Dependence Logic formula, and let $\overline{x} = (x_1 \ldots x_n)$ be a fixed ordering of the free variables of ψ. Then $\Phi[\psi(\overline{x})]$ is the formula obtained from $\Phi[\Xi]$ by substituting each atomic subformula of the form $\Xi(t_1 \ldots t_n)$ with the subformula $\psi(t_1/v_1 \ldots t_n/v_n)$.

Given these definitions, we will prove that

Theorem 3.3.12. *There exists no Dependence Logic context $\Phi[\Xi]$ such that $\Phi(\psi(x))$ is logically equivalent to $\forall^1 x \psi(x)$ for all Dependence Logic formulas $\psi(x)$ with $\mathtt{Free}(\psi) = \{x\}$.*

The following lemma will give us a necessary condition for definability of operators as contexts:

[7] It is possible to define contexts of type $\langle n_1 \ldots n_k \rangle$, which take k formulas as inputs, in the obvious way. But as we have no need for such contexts here, we will limit ourselves to contexts taking only one formula as input.

Lemma 3.3.13. *Let $\Phi[\Xi]$ be a positive context of type $\langle n\rangle$, and let $\{x_1 \dots x_n\}$ be a set of n variables. Then there exists an integer k such that, for every model M and team X, there is a function $W_{M,X,\Phi}$ from k-tuples of teams to $\{0,1\}$ such that, for all formulas $\psi(x_1 \dots x_n)$, $M \models_X \Phi[\psi]$ if and only if there exist teams $Y_1 \dots Y_k$ with domain $\{x_1 \dots x_n\}$ such that $W_{M,X,\Phi}(Y_1 \dots Y_n) = 1$ and, for all $i = 1 \dots n$, $M \models_{Y_i} \psi$.*

Proof. The proof is a straightforward induction over the form of the context $\Phi[\Xi]$:

- If $\Phi[\Xi]$ is $\Xi t_1 \dots t_n$, let $k = 1$ and let $W_{M,X,\Phi}$ be such that $W_{M,X,\Phi}(Y) = 1$ if and only if, for all $s \in X$, the assignment $(x_1 : t_1\langle s\rangle, \dots, x_n : t_n\langle s\rangle)$ is in Y;

- If the variable Ξ does not occur in $\Phi[\Xi]$ and Φ is a literal, let $k = 0$ and let
$$W_{M,X,\Phi}() = \begin{cases} 1 & \text{if } M \models_X \Phi, \\ 0 & \text{otherwise;} \end{cases}$$

- If $\Phi[\Xi]$ is $\Psi_1[\Xi] \vee \Psi_2[\Xi]$, by induction hypothesis we have that, for all teams Z, there exist functions $W_{M,Z,\Psi_1}(Y_1 \dots Y_{k_1})$ and $W_{M,Z,\Psi_2}(Y'_1 \dots Y'_{k_2})$ with the required property. Then, let $k = k_1+k_2$ and let $W_{M,X,\Psi_1\vee\Psi_2}(Y_1 \dots Y_{k_1}, Y'_1 \dots Y'_{k_2}) = 1$ if and only if there exist teams X_1, X_2 such that $X = X_1 \cup X_2$ and
$$W_{M,X_1,\Psi_1}(Y_1 \dots Y_{k_1}) = W_{M,X_1,\Psi_2}(Y'_1 \dots Y'_{k_2}) = 1;$$

- If $\Phi[\Xi]$ is $\Psi_1[\Xi] \wedge \Psi_2[\Xi]$, again, by induction hypothesis there exist functions $W_{M,X,\Psi_1}(Y_1 \dots Y_{k_1})$ and $W_{M,X,\Psi_2}(Y'_1 \dots Y'_{k_2})$ as required. Then, let $k = k_1 + k_2$ and let $W_{M,X,\Psi_1\wedge\Psi_2}(Y_1 \dots Y_{k_1}, Y'_1 \dots Y'_{k_2}) = 1$ if and only if
$$W_{M,X,\Psi_1}(Y_1 \dots Y_{k_1}) = W_{M,X,\Psi_2}(Y'_1 \dots Y'_{k_2}) = 1;$$

- If $\Phi[\Xi]$ is $\exists x \Psi_1[\Xi]$ and, by induction hypothesis, there is a $k_1 \in \mathbb{N}$ such that for each $F : X \to \texttt{Dom}(M)$ there exists a function $W_{M,X[F/x],\Psi_1}(Y_1 \dots Y_{k_1})$ with the required property, let $k = k_1$ and let $W_{M,X,\Psi}(Y_1 \dots Y_k) = 1$ if and only if
$$\exists F : X \to \texttt{Dom}(M) \text{ s.t. } W_{M,X[F/x],\Psi}(Y_1 \dots Y_k) = 1;$$

- If $\Phi[\Xi]$ is $\forall x \Psi_1[\Xi]$ and, by induction hypothesis, there exists a function

$W_{M,X[M/x],\Psi_1}(Y_1 \ldots Y_{k_1})$ as required, let $k = k_1$ and let

$$W_{M,X,\Phi}(Y_1 \ldots Y_k) = W_{M,X[M/x],\Psi_1}(Y_1 \ldots Y_k).$$

□

Finally, we can give a proof of Theorem 3.3.12.

Proof. Suppose that there existed a context $\Phi[\Xi]$, of type $\langle 1 \rangle$, such that

$$M \models_X \Phi[\psi] \Leftrightarrow \text{ for all } m \in \texttt{Dom}(M), M \models_{X[m/x]} \psi$$

for all suitable models M, teams X and formulas $\psi(x)$.

Then, in particular, we have that

$$\mathbb{N} \models_{\{\emptyset\}} \Phi[\psi(x)] \Leftrightarrow \text{ for all } m \in \mathbb{N}, M \models_{\{(x:m)\}} \psi(x)$$

where $\mathbb{N}$ is the model whose domain is the set of all natural numbers, and whose signature contains a constant for every natural number n.

By the lemma, there are an integer k and a function $W_{\mathbb{N},\{\emptyset\},\Phi}$ such that $\mathbb{N} \models_{\{\emptyset\}} \Phi[\psi]$ if and only if there exist teams $Y_1 \ldots Y_k$ with domain $\{x\}$ such that $W_{\mathbb{N},\{\emptyset\},\Phi}(Y_1 \ldots Y_k) = 1$ and $\mathbb{N} \models_{Y_i} \psi(x)$ for all $i = 1 \ldots n$.

Now, since for all $m \in \mathbb{N}$ it holds that $\mathbb{N} \models_{\{(x:m)\}} =\!(x)$, there exist $Y_1 \ldots Y_k$ such that $\mathbb{N} \models_{Y_i} =\!(x)$ for all i and $W_{\mathbb{N},\{\emptyset\},\Phi}(Y_1 \ldots Y_k) = 1$. But then the value of x must be constant in each team Y_i, and therefore there is a natural number m_0 such that $\mathbb{N} \models_{Y_i} (x \neq m_0)$.

Hence, we would have that $\mathbb{N} \models_{\{\emptyset\}} \Phi[x \neq m_0]$; but this contradicts our hypothesis, and hence no context $\Phi[\Xi]$ representing the $\forall^1 x$ operator exists. □

As an aside, it is easy to see that Lemma 3.3.13 holds, with essentially the same proof, also for Independence Logic (Subsection 2.4.1) or for the variants of Dependence Logic through other forms of dependencies of Chapter 4: and hence, the $\forall^1$ operator is not uniformly definable in any of these logics either.

This concludes the proof of Theorem 3.3.12, which answers negatively the question about the uniform definability of the $\forall^1$ quantifier asked in [50]. But perhaps even more significant than the result itself is the manner in which we arrived at it: Lemma 3.3.13, the main ingredient of our proof, is a statement about the *dynamics* of Dependence Logic contexts, or, more precisely, about the ways in which such contexts affect the meanings of formulas. This idea will be at the root of the systems of *dynamic semantics* for Dependence Logic which we will examine in Chapter 6.

Chapter 4

Dependencies in Team Semantics

This chapter is dedicated to the study of various forms of dependency in the framework of Team Semantics. In this, it can be thought of as an examination of the properties of some "generalized atoms" in the sense mentioned by Antti Kuusisto in [53].

First, in Section 4.1, we will examine the fragment of Dependence Logic which only contains *constancy atoms* $=(x)$ and prove that it is equivalent to First Order Logic. Then, in Section 4.2, we will bring into focus the multivalued dependence atoms of [19], and prove that the resulting Multivalued Dependence Logic is equivalent to Independence Logic. After this, in Section 4.3, we will examine the logics obtained by considering *inclusion* and *exclusion* dependencies, or variants thereof; and in Section 4.5, we will characterize the expressive powers of Multivalued Dependence Logic, Independence Logic and Inclusion/Exclusion Logic with respect to open formulas.

Finally, in Section 4.6, we will use many of the results developed in the previous sections to decompose Inclusion Logic and Inclusion/Exclusion Logic in terms of the announcement operators of Chapter 3, of constancy atoms and of a kind of *inconstancy atoms* $\neq(x)$.

4.1 Constancy Logic

In this section, we will present and examine a simple fragment of Dependence Logic. This fragment, which we will call *Constancy Logic*, consists of all the formulas of Dependence Logic in which only dependence atoms of the form $=(t)$ occur; or, equivalently, it can be defined as the extension of First Order Logic obtained by adding *constancy atoms* to it, with the semantics given by the following definition:

Definition 4.1.1. Let M be a first order model, let X be a team over it, and let t be a term over the signature of M and with variables in $\texttt{Dom}(X)$. Then

TS-const: $M \models_X =\!(t)$ if and only if, for all $s, s' \in X$, $t\langle s\rangle = t\langle s'\rangle$.

Clearly, Constancy Logic is contained in Dependence Logic.

Constancy atoms are not expressible in First Order Logic: indeed, by Proposition 2.2.9, the satisfaction conditions of any first-order ϕ are *closed by union* in the sense that

$$M \models_X \phi \text{ and } M \models_Y \phi \Rightarrow M \models_{X\cup Y} \phi$$

whereas this is clearly not the case for $=\!(x)$.

The question then arises whether, with respect to sentences, Constancy Logic is properly contained in Dependence Logic or coincides with it. This will be answered through the following results:

Proposition 4.1.2. *Let ϕ be a Constancy Logic formula, let z be a variable not occurring in ϕ, and let ϕ' be obtained from ϕ by substituting one instance of $=\!(t)$ with the expression $z = t$.*

Then $M \models_X \phi \Leftrightarrow M \models_X \exists z(=\!(z) \wedge \phi')$.

Proof. The proof is by induction on ϕ.

1. If the expression $=\!(t)$ does not occur in ϕ, then $\phi' = \phi$ and we trivially have that $\phi \equiv \exists z(=\!(z) \wedge \phi)$, as required.

2. If ϕ is $=\!(t)$ itself then ϕ' is $z = t$, and

$$M \models_X \exists z(=\!(z) \wedge z = t) \Leftrightarrow \exists m \in \texttt{Dom}(M) \text{ s.t. } M \models_{X[m/z]} z = t \Leftrightarrow$$
$$\Leftrightarrow \exists m \in \texttt{Dom}(M) \text{ s.t. } t\langle s\rangle = m \text{ for all } s \in X \Leftrightarrow M \models_X =\!(t)$$

   as required, where we used $X[m/z]$ as a shorthand for $\{s(m/z) : s \in X\}$.

3. If ϕ is $\psi_1 \vee \psi_2$, let us assume without loss of generality that the instance of $=\!(t)$ that we are considering is in ψ_1. Then $\psi_2' = \psi_2$, and since z does

not occur in ψ_2

$$M \models_X \exists z(=(z) \wedge (\psi_1' \vee \psi_2)) \Leftrightarrow \exists m \text{ s.t. } M \models_{X[m/z]} \psi_1' \vee \psi_2 \Leftrightarrow$$
$$\Leftrightarrow \exists m, X_1, X_2 \text{ s.t. } X_1 \cup X_2 = X, M \models_{X_1[m/z]} \psi_1' \text{ and } M \models_{X_2[m/z]} \psi_2 \Leftrightarrow$$
$$\Leftrightarrow \exists m, X_1, X_2 \text{ s.t. } X_1 \cup X_2 = X, M \models_{X_1[m/z]} \psi_1' \text{ and } M \models_{X_2} \psi_2 \Leftrightarrow$$
$$\Leftrightarrow X_1, X_2 \text{ s.t. } X_1 \cup X_2 = X, M \models_{X_1} \exists z(=(z) \wedge \psi_1') \text{ and } M \models_{X_2} \psi_2 \Leftrightarrow$$
$$\Leftrightarrow X_1, X_2 \text{ s.t. } X_1 \cup X_2 = X, M \models_{X_1} \psi_1 \text{ and } M \models_{X_2} \psi_2 \Leftrightarrow$$
$$\Leftrightarrow M \models_X \psi_1 \vee \psi_2$$

as required.

4. If ϕ is $\psi_1 \wedge \psi_2$, let us assume again that the instance of $=(t)$ that we are considering is in ψ_1. Then $\psi_2' = \psi_2$, and

$$M \models_X \exists z(=(z) \wedge \psi_1' \wedge \psi_2) \Leftrightarrow$$
$$\Leftrightarrow \exists m \text{ s.t. } M \models_{X[m/z]} \psi_1' \text{ and } M \models_{X[m/z]} \psi_2 \Leftrightarrow$$
$$\Leftrightarrow M \models_X \exists z(=(z) \wedge \psi_1') \text{ and } M \models_X \psi_2 \Leftrightarrow$$
$$\Leftrightarrow M \models_X \psi_1 \text{ and } M \models_X \psi_2 \Leftrightarrow$$
$$\Leftrightarrow M \models_X \psi_1 \wedge \psi_2.$$

5. If ϕ is $\exists x \psi$,

$$M \models_X \exists z(=(z) \wedge \exists x \psi') \Leftrightarrow$$
$$\Leftrightarrow \exists m \text{ s.t. } M \models_{X[m/z]} \exists x \psi' \Leftrightarrow$$
$$\Leftrightarrow \exists m, \exists H : X[m/z] \to \texttt{Parts}(\texttt{Dom}(M)) \backslash \{\emptyset\} \text{ s.t. } M \models_{X[m/z][H/x]} \psi' \Leftrightarrow$$
$$\Leftrightarrow \exists H' : X \to \texttt{Parts}(\texttt{Dom}(M)) \backslash \{\emptyset\}, \exists m \text{ s.t. } M \models_{X[H'/x][m/z]} \psi' \Leftrightarrow$$
$$\Leftrightarrow \exists H' : X \to \texttt{Parts}(\texttt{Dom}(M)) \backslash \{\emptyset\} \text{ s.t. } M \models_{X[H'/x]} \exists z(=(z) \wedge \psi') \Leftrightarrow$$
$$\Leftrightarrow \exists H' : X \to \texttt{Parts}(\texttt{Dom}(M)) \backslash \{\emptyset\}, \text{ s.t. } M \models_{X[H'/x]} \psi \Leftrightarrow$$
$$\Leftrightarrow M \models_X \exists x \psi.$$

6. If ϕ is $\forall x \psi$,

$$\begin{aligned}
&M \models_X \exists z(=(z) \wedge \forall x \psi') \Leftrightarrow \\
&\Leftrightarrow \exists m \text{ s.t. } M \models_{X[m/z]} \forall x \psi' \Leftrightarrow \\
&\Leftrightarrow \exists m \text{ s.t. } M \models_{X[m/z][M/x]} \psi' \Leftrightarrow \\
&\Leftrightarrow \exists m \text{ s.t. } M \models_{X[M/x][m/z]} \psi' \Leftrightarrow \\
&\Leftrightarrow M \models_{X[M/x]} \exists z(=(z) \wedge \psi') \Leftrightarrow \\
&\Leftrightarrow M \models_{X[M/x]} \psi \Leftrightarrow \\
&\Leftrightarrow M \models_X \forall x \psi.
\end{aligned}$$

□

As a corollary of this result, we get the following normal form theorem for Constancy Logic:[1]

Corollary 4.1.3. *Let ϕ be a Constancy Logic formula. Then ϕ is logically equivalent to a Constancy Logic formula of the form*

$$\exists z_1 \ldots z_n \left(\bigwedge_{i=1}^{n} =(z_i) \wedge \psi(z_1 \ldots z_n) \right)$$

for some tuple of variables $\vec{z} = z_1 \ldots z_n$ and some first order formula ψ.

Proof. Repeatedly apply Proposition 4.1.2 to "push out" all constancy atoms from ϕ, thus obtaining a formula, equivalent to it, of the form

$$\exists z_1(=(z_1) \wedge \exists z_2(=(z_2) \wedge \ldots \wedge \exists z_n(=(z_n) \wedge \psi(z_1 \ldots z_n)))$$

for some first order formula $\psi(z_1 \ldots z_n)$. It is then easy to see, from the semantics of our logic, that this is equivalent to

$$\exists z_1 \ldots z_n(=(z_1) \wedge \ldots \wedge =(z_n) \wedge \psi(z_1 \ldots z_n))$$

as required. □

The following result shows that, over sentences, Constancy Logic is precisely as expressive as First Order Logic:

[1] This normal form theorem is very similar to the one of Dependence Logic proper found in [65]. See also [17] for a similar, but not identical result, developed independently, which Durand and Kontinen use in that paper in order to characterize the expressive powers of subclasses of Dependence Logic in terms of the maximum allowed width of their dependence atoms.

Corollary 4.1.4. *Let* $\phi = \exists \vec{z} (\bigwedge_i =\!(z_i) \wedge \psi(\vec{z}))$ *be a Constancy Logic sentence in normal form.*

Then ϕ *is logically equivalent to* $\exists \vec{z} \psi(\vec{z})$.

Proof. By the rules of our semantics, $M \models_{\{\emptyset\}} \psi$ if and only if there exists a family $A_1 \ldots A_n$ of nonempty sets of elements in $\mathtt{Dom}(M)$ such that, for

$$X = \{(z_1 := m_1 \ldots z_n := m_n) : (m_1 \ldots m_n) \in A_1 \times \ldots \times A_n\},$$

it holds that $M \models_X \psi$. But ψ is first-order, and therefore, by Proposition 2.2.9, this is the case if and only if for all $m_1 \in A_1, \ldots, m_n \in A_n$ it holds that $M \models_{\{(z_1:m_1,\ldots z_n:m_n)\}} \psi$.

But then $M \models_{\{\emptyset\}} \phi$ is and only if there exist $m_1 \ldots m_n$ such that this holds;[2] and therefore, $M \models_{\{\emptyset\}} \phi$ if and only if $M \models_{\emptyset} \exists z_1 \ldots z_n \psi(z_1 \ldots z_n)$ according to Tarski's semantics, or equivalently, if and only if $M \models_{\{\emptyset\}} \exists z_1 \ldots z_n \psi(z_1 \ldots z_n)$ according to Team Semantics. □

Since Dependence Logic is strictly stronger than First Order Logic over sentences, this implies that Constancy Logic is strictly weaker than Dependence Logic over sentences (and, since sentences are a particular kind of formulas, over formulas too).

The relation between First Order Logic and Constancy Logic, in conclusion, appears somewhat similar to that between Dependence Logic and Independence Logic – that is, in both cases we have a pair of logics which are reciprocally translatable on the level of sentences, but such that one of them is strictly weaker than the other on the level of formulas. This discrepancy between translatability on the level of sentences and translatability on the level of formulas is, in the opinion of the author, one of the most intriguing aspects of logics of imperfect information, and it deserves further investigation.

4.2 Multivalued Dependence Logic is Independence Logic

In [19], Engström introduced the following *multivalued dependence atoms*, based on the multivalued dependencies of Database Theory [21]:

[2] Indeed, if this is the case we can just choose $A_1 = \{m_1\}, \ldots, A_n = \{m_n\}$, and conversely if $A_1 \ldots A_n$ exist with the required properties we can simply select arbitrary elements of them for $m_1 \ldots m_n$.

TS-multidep : $M \models_X \vec{x} \twoheadrightarrow \vec{y}$ if and only if, for $\vec{z}$ listing all variables in the domain of X but not in $\vec{x}\vec{y}$ and for all $s, s' \in X$ with $s(\vec{x}) = s'(\vec{x})$, there exists a $s'' \in X$ with $s''(\vec{x}\vec{y}) = s(\vec{x}\vec{y})$ and $s''(\vec{x}\vec{z}) = s'(\vec{x}\vec{z})$;

This rule violates our locality principle: indeed, by definition, whether an atom $\vec{x} \twoheadrightarrow \vec{y}$ holds in a team depends also on the values of variables which are not among $\vec{x}$ and $\vec{y}$.[3] However, it is a very natural concept which is widely used in the study of databases [58, 13].

In this section, we will prove that the "Multivalued Dependence Logic" obtanined by adding multivalued dependence atoms to First Order Logic is, in fact, equivalent to Independence Logic.

One direction is easy to show: indeed, the truth condition for the Multivalued Dependence Logic is expressible in Σ_1^1, and hence any class of teams (wrt a fixed domain) which is definable through one Multivalued Dependence Logic formulas is also definable through some Independence Logic formula. We can even give an explicit translation: if $\vec{z} = \texttt{Dom}(X) \backslash \{\vec{x}, \vec{y}\}$ then it is not difficult to see that

$$M \models_X \vec{x} \twoheadrightarrow \vec{y} \text{ if and only if } M \models_X \vec{y} \perp_{\vec{x}} \vec{z}.$$

The other direction is slightly more delicate, and in order to prove it we will first need a definition and a couple of lemmas:

Definition 4.2.1. An Independence Logic atom $\vec{t}_2 \perp_{\vec{t}_1} \vec{t}_3$ is said to be *normal* if and only if

1. $\vec{t}_1$, $\vec{t}_2$ and $\vec{t}_3$ are tuples of *variables*, and not just tuples of terms;
2. $\vec{t}_1$, $\vec{t}_2$ and $\vec{t}_3$ are pairwise disjoint.

Lemma 4.2.2. *Any independence atom is expressible in terms of normal independence atoms.*

Proof. Let $\vec{t}_2 \perp_{\vec{t}_1} \vec{t}_3$ be any independence atom, and let $\vec{x}_1$, $\vec{x}_2$ and $\vec{x}_3$ be three tuples of new variables, of the same lengths of $\vec{t}_1$, $\vec{t}_2$ and $\vec{t}_3$ respectively. Then

$$\vec{t}_2 \perp_{\vec{t}_1} \vec{t}_3 \equiv \exists \vec{x}_1 \vec{x}_2 \vec{x}_3 (\vec{x}_1 = \vec{t}_1 \wedge \vec{x}_2 = \vec{t}_2 \wedge \vec{x}_3 = \vec{t}_3 \wedge \vec{x}_2 \perp_{\vec{x}_1} \vec{x}_3).$$

Indeed, suppose that $M \models_X \vec{t}_2 \perp_{\vec{t}_1} \vec{t}_3$: then, choose the functions F_i so that $F_i(s) = \{\vec{t}_i\langle s\rangle\}$ and let $Y = X[F_1F_2F_3/\vec{x}_1\vec{x}_2\vec{x}_3]$. Then $M \models_Y \vec{x}_1 = \vec{t}_1 \wedge \vec{x}_2 = \vec{t}_2 \wedge \vec{x}_3 = \vec{t}_3$, trivially, and furthermore $M \models_Y \vec{x}_2 \perp_{\vec{x}_1} \vec{x}_3$, since $M \models_X \vec{t}_2 \perp_{\vec{t}_1} \vec{t}_3$.

[3]For example, compare the values of $x \twoheadrightarrow y$ in $X = \{(x := 0, y := 0, z := 0), (x := 1, y := 1, z := 1)\}$ and in $Y = \{(x := 0, y := 0, z := 0), (x := 1, y := 1, z := 0)\}$.

Conversely, suppose that $M \models_{X[F_1F_2F_3/\vec{x}_1\vec{x}_2\vec{x}_3]} (\vec{x}_1 = \vec{t}_1 \wedge \vec{x}_2 = \vec{t}_2 \wedge \vec{x}_3 = \vec{t}_3 \wedge \vec{x}_2 \perp_{\vec{x}_1} \vec{x}_3)$. Then, again for $Y = X[F_1F_2F_3/\vec{x}_1\vec{x}_2\vec{x}_3]$ and all $i = 1 \ldots 3$, it must hold that $Y(\vec{x}_i) = \{t_i\langle s\rangle\}$. But then, since $M \models_Y \vec{x}_2 \perp_{\vec{x}_1} \vec{x}_3$, we have that $M \models_Y \vec{t}_2 \perp_{\vec{t}_1} \vec{t}_3$ too. But all variables occurring in $\vec{t}_1\vec{t}_2\vec{t}_3$ are already in $\mathtt{Dom}(X)$, and therefore

$$M \models_X \vec{t}_2 \perp_{\vec{t}_1} \vec{t}_3$$

□

Lemma 4.2.3. *Let $\vec{y} \perp_{\vec{x}} \vec{z}$ be a normal independence atom, let X be a team whose domain includes $\vec{x}$, $\vec{y}$ and $\vec{z}$, and let $\vec{w} = \mathtt{Dom}(X) \backslash \{\vec{x}, \vec{y}, \vec{z}\}$. Then*

$$M \models_X \vec{y} \perp_{\vec{x}} \vec{z} \Leftrightarrow M \models_X \forall \vec{w}(\vec{x} \twoheadrightarrow \vec{y}).$$

Proof. Suppose that $M \models_X \vec{y} \perp_{\vec{x}} \vec{z}$: then, by definition, for all $s, s' \in X$ with $s(\vec{x}) = s'(\vec{x})$ there exists a $s'' \in X$ with $s''(\vec{x}\vec{y}) = s(\vec{x}\vec{y})$ and $s''(\vec{x}\vec{z}) = s'(\vec{x}\vec{z})$.

Now consider any two assignments $h, h' \in X[M/\vec{w}]$ with $h(\vec{x}) = h'(\vec{x})$: by definition, there exist $s, s' \in X$ and $\vec{m}_1, \vec{m}_2 \in \mathtt{Dom}(M)^{|\vec{w}|}$ such that $h = s[\vec{m}_1/\vec{w}]$ and $h' = s'[\vec{m}_2/\vec{w}]$. But $s(\vec{x}) = s'(\vec{x})$, so by hypothesis there exists a s'' with $s''(\vec{x}\vec{y}) = s(\vec{x}\vec{y})$ and $s''(\vec{x}\vec{z}) = s'(\vec{x}\vec{z})$. Then consider $h'' = s''[\vec{m}_2/\vec{w}]$: we have that $h'' \in X[M/\vec{w}]$, since $s'' \in X$, and furthermore

$$h''(\vec{x}\vec{y}) = s''(\vec{x}\vec{y}) = s(\vec{x}\vec{y}) = h(\vec{x}\vec{y});$$
$$h''(\vec{x}\vec{z}\vec{w}) = s''(\vec{x}\vec{z})\vec{m}_2 = s'(\vec{x}\vec{z})h'(\vec{w}) = h'(\vec{x}\vec{z}\vec{w}).$$

Therefore $M \models_{X[M/\vec{w}]} \vec{x} \twoheadrightarrow \vec{y}$ and $M \models_X \forall \vec{w}(\vec{x} \twoheadrightarrow \vec{y})$, as required.

Conversely, suppose that $M \models_{X[M/\vec{w}]} \vec{x} \twoheadrightarrow \vec{y}$, and let $s, s' \in X$ be such that $s(\vec{x}) = s'(\vec{x})$. Then take any tuple $\vec{m} \in \mathtt{Dom}(M)^{|\vec{w}|}$, and consider

$$h = s[\vec{m}/\vec{w}];$$
$$h' = s'[\vec{m}/\vec{w}].$$

Now, $\mathtt{Dom}(X) \backslash \{\vec{x}\vec{y}\}$ is precisely $\vec{z}\vec{w}$: therefore, by the definition of the multivalued dependence atom there exists a $h'' \in X[M/\vec{w}]$ with $h''(\vec{x}\vec{y}) = h(\vec{x}\vec{y})$ and $h''(\vec{x}\vec{z}\vec{w}) = h'(\vec{x}\vec{z}\vec{w})$. Since $h'' \in X[M/\vec{w}]$, we must have that $h'' = s''[\vec{m}/\vec{w}]$ for some $s'' \in X$; and for this s'', we have that

$$s''(\vec{x}\vec{y}) = h''(\vec{x}\vec{y}) = h(\vec{x}\vec{y}) = s(\vec{x}\vec{y})$$

and that

$$s''(\vec{x}\vec{z}) = h''(\vec{x}\vec{z}) = h'(\vec{x}\vec{z}) = s'(\vec{x}\vec{z}).$$

□

Theorem 4.2.4. *Multivalued Dependence Logic is precisely as expressive as Independence Logic, over sentences and over open formulas considered in teams with finite domain.*

Proof. Obvious from the previous results. □

As an aside, this result is independent on the choice between the usual semantics for the existential quantifier and the "lax" one **TS-∃-lax** described at the end of Subsection 2.2.1: indeed, in Lemma 4.2.2 nothing can be gained by selecting more than one possible value per existentially quantified formula and assignment, and no existential quantifier is needed for Lemma 4.2.3. Hence, the equivalence between these logics holds even if, as we will suggest in the next section, Rule **TS-∃-lax** is to be preferred to Rule **TS-∃** for non downwards-closed logics such as Independence Logic.

4.3 Inclusion and Exclusion in Logic

This section is the central part of the present chapter. We will begin it by recalling some forms of non-functional dependency which have been studied in Database Theory, and some of their known properties. Then we will briefly discuss their relevance in the framework of logics of imperfect information, and then, in Subsection 4.3.2, we will examine the properties of the logic obtained by adding atoms corresponding to the first sort of non-functional dependency to the basic language of Team Semantics. Afterward, in Subsection 4.3.3 we will see that nothing is lost if we only consider a simpler variant of this kind of dependency: in either case, we obtain essentially the same logic, which – as we will see – is strictly more expressive than First Order Logic, strictly weaker than Independence Logic, but incomparable with Dependence Logic. In Subsection 4.3.4, we will then study the other notion of non-functional dependency that we are considering, and see that the corresponding logic is instead equivalent, in a very strong sense, to Dependence Logic; and finally, in Subsection 4.3.5 we will examine the logic obtained by adding *both* forms of non-functional dependency to our language, and see that it is equivalent to Independence Logic.

4.3.1 Inclusion and Exclusion Dependencies

Functional dependencies are the forms of dependency which attracted the most interest from database theorists, but they certainly are not the only ones ever considered in that field. Therefore, studying the effect of substituting the dependence atoms with ones corresponding to other forms of dependency, and examining the relationship between the resulting logics, may be – in the author's opinion, at least – a very promising, and hitherto not sufficiently explored, direction of research in the field of logics of imperfect information. First of all, as we will discuss in more detail in Chapter 7 but as the Game Theoretic Semantics of Subsection 2.2.3 and the interpretation of the announcement operators of Chapter 3 already suggest, teams correspond to states of knowledge. But often, relations obtained from a database correspond precisely to information states of this kind;[4] and therefore, some of the dependencies studied in Database Theory may correspond to constraints over the agent's beliefs which often occur in practice, as is certainly the case for functional dependencies.[5]

Moreover, and perhaps more pragmatically, database researchers have already performed a vast amount of research about the properties of many of these non-functional dependencies, and it does not seem unreasonable to hope that this might allow us to derive, with little additional effort of our own, some useful results about the corresponding logics.

This chapter will, for the most part, focus on *inclusion* ([22], [9]) and *exclusion* ([10]) dependencies and on the properties of the corresponding logics of imperfect information. Let us start by recalling and briefly discussing these dependencies:

Definition 4.3.1. Let R be a relation, and let $\vec{x}$, $\vec{y}$ be tuples of attributes of R of the same length. Then $R \models \vec{x} \subseteq \vec{y}$ if and only if $R(\vec{x}) \subseteq R(\vec{y})$, where

$$R(\vec{z}) = \{r(\vec{z}) : r \text{ is a tuple in } R\}.$$

In other words, an inclusion dependency $\vec{x} \subseteq \vec{y}$ states that all values taken by the attributes $\vec{x}$ are also taken by the attributes $\vec{y}$. It is easy to think of practical examples of inclusion dependencies: one might for instance think of the database consisting of the relations (Person, Date_of_Birth), (Father, Child_of_Father) and (Mother, Child_of_Mother).[6] Then, in order to express

[4] As a somewhat naive example, let us consider the problem of finding a spy, knowing that yesterday he took a plane from London's Heathrow airport and that he had at most 100 EUR available to buy his plane ticket. We might then decide to obtain, from the airport systems, the list of the destinations of all the planes which left Heathrow yesterday and whose ticket the spy could have afforded; and this list – that is, the list of all the places that the spy might have reached – would be a state of information of the type which we are discussing.

[5] For example, our system should be able to represent the assertion that the flight code always determines the destination of the flight.

[6] Equivalently, one may consider the Cartesian product of these relations, as per the universal relation model ([23]).

the statement that every father, every mother and every child in our knowledge base are people and have a date of birth, we may impose the restrictions

$$\left\{\begin{array}{l}\text{Father} \subseteq \text{Person},\ \text{Mother} \subseteq \text{Person},\\ \text{Child\_of\_Father} \subseteq \text{Person},\ \text{Child\_of\_Mother} \subseteq \text{Person}\end{array}\right\}.$$

Furthermore, inclusion dependencies can be used to represent the assertion that every child has a father and a mother, or, in other words, that the attributes Child_of_Father and Child_of_Mother take the same values:

$$\{\text{Child\_of\_Father} \subseteq \text{Child\_of\_Mother},\ \text{Child\_of\_Mother} \subseteq \text{Child\_of\_Father}\}.$$

Note, however, that inclusion dependencies do not allow us to express all "natural" dependencies of our example. For instance, we need to use functional dependencies in order to assert that everyone has exactly one birth date, one father and one mother:[7]

$$\left\{\begin{array}{l}\text{Person} \rightarrow \text{Date\_of\_Birth},\ \text{Child\_of\_Father} \rightarrow \text{Father},\\ \text{Child\_of\_Mother} \rightarrow \text{Mother}\end{array}\right\}.$$

In [9], a sound and complete axiom system for the implication problem of inclusion dependencies was developed. This system consists of the three following rules:

I1: For all $\vec{x}$, $\vdash \vec{x} \subseteq \vec{x}$;

I2: If $|\vec{x}| = |\vec{y}| = n$ then, for all $m \in \mathbb{N}$ and all $\pi : 1 \ldots m \rightarrow 1 \ldots n$,

$$\vec{x} \subseteq \vec{y} \vdash x_{\pi(1)} \ldots x_{\pi(m)} \subseteq y_{\pi(1)} \ldots y_{\pi(m)};$$

I3: For all tuples of attributes of the same length $\vec{x}$, $\vec{y}$, and $\vec{z}$,

$$\vec{x} \subseteq \vec{y}, \vec{y} \subseteq \vec{z} \vdash \vec{x} \subseteq \vec{z}.$$

Theorem 4.3.2 (Soundness and completeness of inclusion axioms [9])**.** *Let Γ be a set of inclusion dependencies and let $\vec{x}$, $\vec{y}$ be tuples of relations of the same length. Then*

$$\Gamma \vdash \vec{x} \subseteq \vec{y}$$

[7] The simplest way to verify that these conditions are not expressible in terms of inclusion dependencies is probably to observe that inclusion dependencies are *closed under unions*: if the relations R and S respect $\vec{x} \subseteq \vec{y}$, so does $R \cup S$. Since functional dependencies as the above ones are clearly *not* closed under unions, they cannot be represented by inclusions.

*can be derived from the axioms **I1**, **I2** and **I3** if and only if all relations which respect all dependencies of* Γ *also respect* $\vec{x} \subseteq \vec{y}$.

However, the combined implication problem for inclusion and functional dependencies is undecidable ([55], [11]).

Whereas inclusion dependencies state that all values of a given tuple of attributes also occur as values of another tuple of attributes, *exclusion* dependencies state that two tuples of attributes have no values in common:

Definition 4.3.3. Let R be a relation, and let $\vec{x}$, $\vec{y}$ be tuples of attributes of R of the same length. Then $R \models \vec{x} \mid \vec{y}$ if and only if $R(\vec{x}) \cap R(\vec{y}) = \emptyset$, where

$$R(\vec{z}) = \{r(\vec{z}) : r \text{ is a tuple in } R\}.$$

Exclusion dependencies can be thought of as a way of partitioning the elements of our domain into *data types*, and of specifying which type corresponds to each attribute. For instance, in the example

(Person, Date_of_birth) × (Father, Child_of_Father) × (Mother, Child_of_Mother)

considered above we have two types, corresponding respectively to *people* (for the attributes Person, Father, Mother, Child_of_Father and Child_of_Mother) and *dates* (for the attribute Date_of_birth). The requirement that no date of birth should be accepted as a name of person, nor vice versa, can then be expressed by the set of exclusion dependencies

$$\{A \mid \text{Date\_of\_birth} : A = \text{Person}, \text{Father}, \text{Mother}, \ldots\}.$$

Other uses of exclusion dependencies are less common, but they still exist: for example, the statement that no one is both a father and a mother might be expressed as Father | Mother.

In [10], the axiom system for inclusion dependencies was extended to deal with both inclusion and exclusion dependencies as follows:

1. *Axioms for inclusion dependencies:*

   I1: For all $\vec{x}$, $\vdash \vec{x} \subseteq \vec{x}$;

   I2: If $|\vec{x}| = |\vec{y}| = n$ then, for all $m \in \mathbb{N}$ and all $\pi : 1 \ldots m \to 1 \ldots n$,

   $$\vec{x} \subseteq \vec{y} \vdash x_{\pi(1)} \ldots x_{\pi(m)} \subseteq y_{\pi(1)} \ldots y_{\pi(m)};$$

I3: For all tuples of attributes of the same length $\vec{x}$, $\vec{y}$ and $\vec{z}$,

$$\vec{x} \subseteq \vec{y}, \vec{y} \subseteq \vec{z} \vdash \vec{x} \subseteq \vec{z};$$

2. *Axioms for exclusion dependencies:*

E1: For all $\vec{x}$ and $\vec{y}$ of the same length, $\vec{x} \mid \vec{y} \vdash \vec{y} \mid \vec{x}$;

E2: If $|\vec{x}| = |\vec{y}| = n$ then, for all $m \in \mathbb{N}$ and all $\pi : 1 \ldots m \to 1 \ldots n$,

$$x_{\pi(1)} \ldots x_{\pi(m)} \mid y_{\pi(1)} \ldots y_{\pi(m)} \vdash \vec{x} \mid \vec{y};$$

E3: For all $\vec{x}$, $\vec{y}$ and $\vec{z}$ such that $|\vec{y}| = |\vec{z}|$, $\vec{x} \mid \vec{x} \vdash \vec{y} \mid \vec{z}$;

3. *Axioms for inclusion/exclusion interaction:*

IE1: For all $\vec{x}$, $\vec{y}$ and $\vec{z}$ such that $|\vec{y}| = |\vec{z}|$, $\vec{x} \mid \vec{x} \vdash \vec{y} \subseteq \vec{z}$;

IE2: For all $\vec{x}, \vec{y}, \vec{z}, \vec{w}$ of the same length, $\vec{x} \mid \vec{y}, \vec{z} \subseteq \vec{x}, \vec{w} \subseteq \vec{y} \vdash \vec{z} \mid \vec{w}$.

Theorem 4.3.4 ([10])**.** *The above system is sound and complete for the implication problem for inclusion and exclusion dependencies.*

It is not difficult to transfer the definitions of inclusion and exclusion dependencies to Team Semantics, thus obtaining *inclusion atoms* and *exclusion atoms*:

Definition 4.3.5. Let M be a first order model, let $\vec{t_1}$ and $\vec{t_2}$ be two finite tuples of terms of the same length over the signature of M, and let X be a team whose domain contains all variables occurring in $\vec{t_1}$ and $\vec{t_2}$. Then

TS-inc: $M \models_X \vec{t_1} \subseteq \vec{t_2}$ if and only if for every $s \in X$ there exists a $s' \in X$ such that $\vec{t_1}\langle s \rangle = \vec{t_2}\langle s' \rangle$;

TS-exc: $M \models_X \vec{t_1} \mid \vec{t_2}$ if and only if for all $s, s' \in X$, $\vec{t_1}\langle s \rangle \neq \vec{t_2}\langle s' \rangle$.

Returning for a moment to the agent metaphor, the interpretation of these conditions is as follows.

A team X satisfies $\vec{t_1} \subseteq \vec{t_2}$ if and only if all possible values that the agent believes possible for $\vec{t_1}$ are also believed by him or her as possible for $\vec{t_2}$ – or, by contraposition, that the agent cannot exclude any value for $\vec{t_2}$ which he cannot also exclude as a possible value for $\vec{t_1}$. In other words, from this point of view an inclusion atom is a way of specify a state of *ignorance* of the agent: for example, if the agent is a chess player who is participating to a tournament, we may want to represent the assertion that the agent *does not know* whether he will play against a given opponent using the black pieces or the white ones. In other words, if he believes that he *might* play against a given opponent when using the white pieces, he should also consider it possible that he played against him or her using the black ones, and vice versa; or, in our formalism, that his belief set satisfies the conditions

$$\text{Opponent\_as\_White} \subseteq \text{Opponent\_as\_Black},$$
$$\text{Opponent\_as\_Black} \subseteq \text{Opponent\_as\_White}.$$

This very example can be used to introduce a new dependency atom $\vec{t}_1 \bowtie \vec{t}_2$, which might perhaps be called an *equiextension atom*, with the following rule:

Definition 4.3.6. Let M be a first order model, let $\vec{t}_1$ and $\vec{t}_2$ be two finite tuples of terms of the same length over the signature of M, and let X be a team whose domain contains all variables occurring in $\vec{t}_1$ and $\vec{t}_2$. Then

TS-equ: $M \models_X \vec{t}_1 \bowtie \vec{t}_2$ if and only if $X(\vec{t}_1) = X(\vec{t}_2)$.

It is easy to see that this atom is different, and strictly weaker, from the first order formula

$$\vec{t}_1 = \vec{t}_2 := \bigwedge_i ((\vec{t}_1)_i = (\vec{t}_2)_i).$$

Indeed, the former only requires that the sets of all possible values for $\vec{t}_1$ and for $\vec{t}_2$ are the same, while the latter requires that $\vec{t}_1$ and $\vec{t}_2$ coincide in all possible states of things: and hence, for example, the team $X = \{(x : 0, y : 1), (x : 1, y : 0)\}$ satisfies $x \bowtie y$ but not $x = y$.

As we will see later, it is possible to recover inclusion atoms from equiextension atoms and the connectives of our logics.

Conversely, an exclusion atom describes a state of *knowledge*. More precisely, a team X satisfies $\vec{t}_1 \mid \vec{t}_2$ if and only if the agent can confidently exclude all values that he believes possible for $\vec{t}_1$ from the list of the possible values for $\vec{t}_2$. For example, let us suppose that our agent is also aware that a boxing match will be had at the same time of the chess tournament, and that he knows that no one of the participants to the match will have the time to play in the tournament too – he has seen the lists of the participants to the two events, and they are disjoint. Then, in particular, our agent knows that no potential winner of the boxing match is also a potential winner of the chess tournament, even though he is not aware of who these winners will be. In our framework, this can be represented by stating our agent's beliefs respect the exclusion atom

$$\text{Winner\_Boxing} \mid \text{Winner\_Chess}.$$

This is a different, and stronger, condition than the first order expression Winner_Boxing $\neq$ Winner_Chess: indeed, the latter merely requires that, in any possible state of things, the winners of the boxing match and of the chess tournament are different, while the former requires that *no possible* winner of the boxing match is a potential winner for the chess tournament. So, for example, only the first condition excludes the scenario in which our agent does not know whether T. Dovramadjiev, a Bulgarian chessboxing[8] champion, will play in the chess tournament or in the boxing match, represented by the team of the form

$X =$

	Winner_Boxing	Winner_Chess
s_0	T. Dovramadjiev	V. Anand
s_1	T. Woolgar	T. Dovramadijev
...	...	...

4.3.2 Inclusion Logic

In this section, we will begin to examine the properties of *Inclusion Logic* – that is, the logic obtained by adding to the language of First Order Logic the *inclusion atoms* $\vec{t}_1 \subseteq \vec{t}_2$ with the semantics of Definition 4.3.5.

A first, easy observation is that this logic does not respect the downwards closure property. For example, consider the two assignments $s_0 = (x : 0, y : 1)$ and $s_1 = (x : 1, y : 0)$: then, for $X = \{s_0, s_1\}$ and $Y = \{s_0\}$, it is easy to see by rule **TS-inc** that $M \models_X x \subseteq y$ but $M \not\models_Y x \subseteq y$.

Hence, the question arises whether the "strict" and the "lax" semantics for the existential quantifier discussed in Subsection 2.2.1 are equivalent for the case of Inclusion Logic, and, if they are not, which one should be preferred.

As the next proposition shows, lax and strict semantics are indeed different for this logic:

Proposition 4.3.7. *There exist a model M, a team X and a formula ϕ of Inclusion Logic such that $M \models_X \exists x\phi$ according to the lax semantics of Rule **TS-∃-Lax** but not according to the strict semantics of Rule **TS-∃**.*

Proof. Let $\texttt{Dom}(M) = \{0, 1\}$, let X be the team

$X =$

	y	z
s_0	0	1

and let ϕ be $y \subseteq x \wedge z \subseteq x$.

[8] Chessboxing is a hybrid sport, in which chess and boxing rounds are alternated.

- $M \models_X \exists x\phi$ according to the lax semantics:
  Let $H : X \to \texttt{Parts}(\texttt{Dom}(M))$ be such that $H(s_0) = \{0, 1\}$.

  Then

$$X[H/x] = \begin{array}{c|ccc} & y & z & x \\ \hline s'_0 & 0 & 1 & 0 \\ s'_1 & 0 & 1 & 1 \end{array}$$

  and hence $X[H/x](y), X[H/x](z) \subseteq X[H/x](x)$, as required.

- $M \not\models_X \exists x\psi$ according to the strict semantics:
  Let F be any function from X to $\texttt{Dom}(M)$. Then

$$X[F/x] = \begin{array}{c|ccc} & y & z & x \\ \hline s''_0 & 0 & 1 & F(s_0) \end{array}$$

  But $F(s_0) \neq 0$ or $F(s_0) \neq 1$; and in the first case $M \not\models_{X[F/x]} y \subseteq x$, while in the second one $M \not\models_{X[F/x]} z \subseteq x$.

□

Therefore, when studying the properties Inclusion Logic it is necessary to specify whether we are are using the strict or the lax semantics for existential quantification. However, only one of these choices preserves *locality* in the sense of Proposition 2.2.7, as the two following results show:

Proposition 4.3.8. *The strict semantics does not respect locality in Inclusion Logic (or in any extension thereof). In other words, with respect to it there exists a model M, a team X and a formula ξ such that $M \models_X \exists x\xi$, but for $X' = X_{\upharpoonright \texttt{Free}(\exists x\xi)}$ we have that $M \not\models_{X'} \exists x\xi$ instead.*

Proof. Let $\texttt{Dom}(M) = \{0, 1\}$, let ξ be $y \subseteq x \wedge z \subseteq x$, and let

$$X = \begin{array}{c|ccc} & y & z & u \\ \hline s_0 & 0 & 1 & 0 \\ s_1 & 0 & 1 & 1 \end{array}$$

Then $M \models_X \exists x\xi$: indeed, for $F : X \to \texttt{Dom}(M)$ defined as

$$F(s_0) = 0;$$
$$F(s_1) = 1;$$

we have that

$$X[F/x] = \begin{array}{c|cccc} & y & z & u & x \\ \hline s'_0 & 0 & 1 & 0 & 0 \\ s'_1 & 0 & 1 & 1 & 1 \end{array}$$

and it is easy to check that this team satisfies ξ. However, the restriction X' of X to $\texttt{Free}(\exists x \xi) = \{y, z\}$ is the team considered in the proof of Proposition 4.3.7, and – again, as shown in that proof – $M \not\models_X \exists x \xi$. □

Theorem 4.3.9 (Inclusion Logic with lax semantics is local)**.** *Let M be a first order model, let ϕ be any Inclusion Logic formula, and let V be a set of variables with $\texttt{Free}(\phi) \subseteq V$. Then, for all suitable teams X,*

$$M \models_X \phi \Leftrightarrow M \models_{X_{\upharpoonright V}} \phi$$

with respect to the lax interpretation of existential quantification.

Proof. The proof is by structural induction on ϕ.

In Section 4.3.5, Theorem 4.3.23, we will prove the same result for an extension of Inclusion Logic; so we refer to that theorem for the details of the proof. □

Because of these results, for the rest of this chapter we will exclusively concern ourselves with the lax semantics for existential quantification.

Since, as we saw, Inclusion Logic is not downwards closed, by Proposition 2.2.7 it is not contained in Dependence Logic. It is then natural to ask whether Dependence Logic is contained in Inclusion Logic, or if Dependence and Inclusion Logic are two incomparable extensions of First Order Logic.

This is answered by the following result, and by its corollary:

Theorem 4.3.10. *Let ϕ be any Inclusion Logic formula, let M be a first order model and let $(X_i)_{i \in I}$ be a family of teams with the same domain such that $M \models_{X_i} \phi$ for all $i \in I$. Then, for $X = \bigcup_{i \in I} X_i$, we have that $M \models_X \phi$.*

Proof. By structural induction on ϕ.

1. If ϕ is a first order literal, this is obvious.

2. Suppose that $M \models_{X_i} \vec{t}_1 \subseteq \vec{t}_2$ for all $i \in I$. Then $M \models_X \vec{t}_1 \subseteq \vec{t}_2$. Indeed, let $s \in X$: then $s \in X_i$ for some $i \in I$, and hence there exists another $s' \in X_i$ with $s'(\vec{t}_2) = s(\vec{t}_1)$. Since $X_i \subseteq X$ we then have that $s' \in X$, as required.

3. Suppose that $M \models_{X_i} \psi \vee \theta$ for all $i \in I$. Then each X_i can be split into two subteams Y_i and Z_i with $M \models_{Y_i} \psi$ and $M \models_{Z_i} \theta$. Now, let $Y = \bigcup_{i \in I} Y_i$ and $Z = \bigcup_{i \in I} Z_i$: by induction hypothesis, $M \models_Y \psi$ and $M \models_Z \theta$. Furthermore, $Y \cup Z = \bigcup_{i \in I} Y_i \cup \bigcup_{i \in I} Z_i = \bigcup_{i \in I}(Y_i \cup Z_i) = X$, and hence $M \models_X \psi \vee \theta$, as required.

4. Suppose that $M \models_{X_i} \psi \wedge \theta$ for all $i \in I$. Then for all such i, $M \models_{X_i} \psi$ and $M \models_{X_i} \theta$; but then, by induction hypothesis, $M \models_X \psi$ and $M \models_X \theta$, and therefore $M \models_X \psi \wedge \theta$.

5. Suppose that $M \models_{X_i} \exists x \psi$ for all $i \in I$, that is, that for all such i there exists a function $H_i : X_i \to \texttt{Parts}(\texttt{Dom}(M)) \backslash \{\emptyset\}$ such that $M \models_{X_i[H_i/x]} \psi$. Then define the function $H : X \to \texttt{Parts}(\texttt{Dom}(M)) \backslash \{\emptyset\}$ so that, for all $s \in X$, $H(s) = \bigcup \{H_i(s) : s \in X_i\}$. Now, $X[H/x] = \bigcup_{i \in I}(X_i[H_i/x])$, and hence by induction hypothesis $M \models_{X[H/x]} \psi$, and therefore $M \models_X \exists x \psi$.

6. Suppose that $M \models_{X_i} \forall x \psi$ for all $i \in I$, that is, that $M \models_{X_i[M/x]} \psi$ for all such i. Then, since $\bigcup_{i \in I}(X_i[M/x]) = \left(\bigcup_{i \in I} X_i\right)[M/x] = X[M/x]$, by induction hypothesis $M \models_{X[M/x]} \psi$ and therefore $M \models_X \forall x \psi$, as required.

□

Corollary 4.3.11. *There exist Constancy Logic formulas which are not equivalent to any Inclusion Logic formula.*

Proof. This follows at once from the fact that the constancy atom $=(x)$ is not closed under unions.

Indeed, let M be any model with two elements 0 and 1 in its domain, and consider the two teams $X_0 = \{(x : 0)\}$ and $X_1 = \{(x : 1)\}$: then $M \models_{X_0} =(x)$ and $M \models_{X_1} =(x)$, but $M \not\models_{X_0 \cup X_1} =(x)$. □

Therefore, not only Inclusion Logic does not contain Dependence Logic, it does not even contain Constancy Logic!

As discussed in Subsection 2.4.1, it is known that Dependence Logic is properly contained in Independence Logic. As the following result shows, Inclusion Logic is also (properly, because dependence atoms are expressible in Independence Logic) contained in Independence Logic:

Theorem 4.3.12. *Inclusion atoms are expressible in terms of Independence Logic formulas. More precisely, an inclusion atom $\vec{t}_1 \subseteq \vec{t}_2$ is equivalent to the Independence Logic formula*

$$\phi := \forall v_1 v_2 \vec{z}((\vec{z} \neq \vec{t}_1 \wedge \vec{z} \neq \vec{t}_2) \vee (v_1 \neq v_2 \wedge \vec{z} \neq \vec{t}_2) \vee ((v_1 = v_2 \vee \vec{z} = \vec{t}_2) \wedge \vec{z} \perp v_1 v_2)).$$

where v_1, v_2 and $\vec{z}$ do not occur in $\vec{t_1}$ or $\vec{t_2}$ and where $\vec{z} \perp v_1 v_2$ is a shorthand for $\vec{z} \perp_{\emptyset} v_1 v_2$.

Proof. Suppose that $M \models_X \vec{t_1} \subseteq \vec{t_2}$. Then split the team $X' = X[M/v_1 v_2 \vec{z}]$ into three teams Y, Z and W as follows:

- $Y = \{s \in X' : s(\vec{z}) \neq \vec{t_1}\langle s\rangle \text{ and } s(\vec{z}) \neq \vec{t_2}\langle s\rangle\}$;
- $Z = \{s \in X' : s(v_1) \neq s(v_2) \text{ and } s(\vec{z}) \neq \vec{t_2}\langle s\rangle\}$;
- $W = X' \backslash (Y \cup Z) = \{s \in X' : s(\vec{z}) = \vec{t_2}\langle s\rangle \text{ or } (s(\vec{z}) = \vec{t_1}\langle s\rangle \text{ and } s(v_1) = s(v_2))\}$.

Clearly, $X' = Y \cup Z \cup W$, $M \models_Y z \neq t_1 \wedge z \neq t_2$ and $M \models_Z v_1 \neq v_2 \wedge z \neq t_2$; hence, if we can prove that

$$M \models_W ((v_1 = v_2 \vee \vec{z} = \vec{t_2})) \wedge \vec{z} \perp v_1 v_2$$

we can conclude that $M \models_X \phi$, as required.

Now, suppose that $s \in W$ and $s(v_1) \neq s(v_2)$: then necessarily $s(\vec{z}) = \vec{t_2}$, since otherwise we would have that $s \in Z$ instead. Hence, the first conjunct $v_1 = v_2 \vee \vec{z} = \vec{t_2}$ is satisfied by W.

Now, consider two assignments s and s' in W: in order to conclude this direction of the proof, we need to show that there exists a $s'' \in W$ such that $s''(\vec{z}) = s(\vec{z})$ and $s''(v_1 v_2) = s'(v_1 v_2)$. There are two distinct cases to examine:

1. If $s(\vec{z}) = \vec{t_2}\langle s\rangle$, consider the assignment

   $$s'' = s[s'(v_1)/v_1][s'(v_2)/v_2] :$$

   by construction, $s'' \in X'$. Furthermore, since $s''(\vec{z}) = \vec{t_2}\langle s\rangle = \vec{t_2}\langle s''\rangle$, s'' is neither in Y nor in Z. Hence, it is in W, as required.

2. If $s(\vec{z}) \neq \vec{t_2}\langle s\rangle$ and $s \in W$, then necessarily $s(\vec{z}) = \vec{t_1}\langle s\rangle$ and $s(v_1) = s(v_2)$.

   Since $s \in W \subseteq X[M/v_1 v_2 \vec{z}]$, there exists an assignment $o \in X$ such that

   $$\vec{t_1}\langle o\rangle = \vec{t_1}\langle s\rangle = s(\vec{z});$$

   and since $M \models_X \vec{t_1} \subseteq \vec{t_2}$, there also exist an assignment $o' \in X$ such that

   $$\vec{t_2}\langle o'\rangle = \vec{t_1}\langle o\rangle = s(\vec{z}).$$

Now consider the assignment $s'' = o'[s'(v_1)/v_1][s'(v_2)/v_2][s(\vec{z})/\vec{z}]$: by construction, $s'' \in X'$, and since

$$s''(\vec{z}) = s(\vec{z}) = \vec{t}_2\langle o'\rangle = \vec{t}_2\langle s''\rangle$$

we have that $s'' \in W$, that $s''(\vec{z}) = s(\vec{z})$ and that $s''(v_1v_2) = s'(v_1v_2)$, as required.

Conversely, suppose that $M \models_X \phi$, let 0 and 1 be two distinct elements of the domain of M, and let $s \in X$.

By the definition of ϕ, the fact that $M \models_X \phi$ implies that the team $X[M/v_1v_2\vec{z}]$ can be split into three teams Y, Z and W such that

$$M \models_Y \vec{z} \neq \vec{t}_1 \wedge \vec{z} \neq \vec{t}_2;$$
$$M \models_Z v_1 \neq v_2 \wedge \vec{z} \neq \vec{t}_2;$$
$$M \models_W (v_1 = v_2 \vee \vec{z} = \vec{t}_2) \wedge \vec{z} \perp v_1v_2.$$

Then consider the assignments

$$h = s[0/v_1][0/v_2][\vec{t}_1\langle s\rangle/\vec{z}]$$

and

$$h' = s[0/v_1][1/v_2][\vec{t}_2\langle s\rangle/\vec{z}]$$

Clearly, h and h' are in $X[M/v_1v_2\vec{z}]$. However, neither of them is in Y, since $h(\vec{z}) = \vec{t}_1\langle h\rangle$ and $h'(\vec{z}) = \vec{t}_2\langle h'\rangle$, nor in Z, since $h(v_1) = h(v_2)$ and, again, since $h'(\vec{z}) = \vec{t}_2\langle h'\rangle$. Hence, both of them are in W.

But we know that $M \models_W \vec{z} \perp v_1v_2$, and thus there exists an assignment $h'' \in W$ with

$$h''(\vec{z}) = h(\vec{z}) = \vec{t}_1\langle s\rangle$$

and

$$h''(v_1v_2) = h'(v_1v_2) = 01.$$

Now, since $h''(v_1) \neq h''(v_2)$, since $h'' \in W$ and since

$$M \models_W v_1 = v_2 \vee \vec{z} = \vec{t}_2,$$

it must be the case that $h''(\vec{z}) = \vec{t}_2\langle h''\rangle$.

Finally, this h'' corresponds to some $s'' \in X$; and for this s'',

$$\vec{t}_2\langle s''\rangle = \vec{t}_2\langle h''\rangle = h''(\vec{z}) = h(\vec{z}) = \vec{t}_1\langle s\rangle.$$

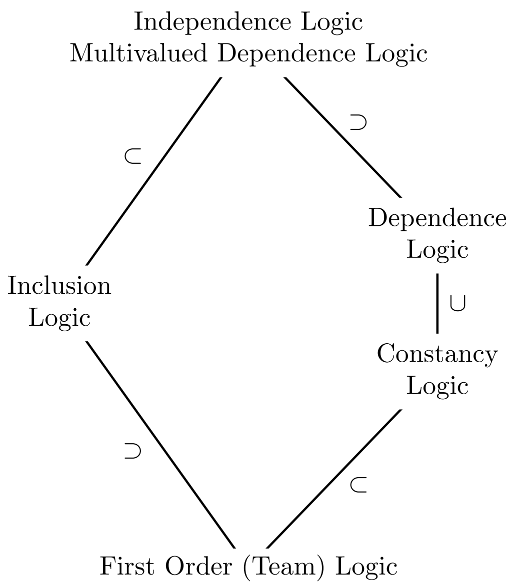

Figure 4.1: Translatability relations between logics (wrt formulas)

This concludes the proof. □

The relations between First Order Logic with Team Semantics, Constancy Logic, Dependence Logic, Inclusion Logic and Independence Logic discovered so far are then represented by Figure 4.1.

However, things change if we take in consideration the the expressive power of these logics with respect to their sentences only. Then, as we saw, First Order Logic and Constancy Logic have the same expressive power, in the sense that every Constancy Logic formula is equivalent to some first order formula and vice versa, and so do Dependence and Independence Logic. What about Inclusion Logic sentences?

At the moment, relatively little is known by the author about this. In essence, all that we know is the following result:

Proposition 4.3.13. *Let $\psi(\vec{x}, \vec{y})$ be any first order formula, where $\vec{x}$ and $\vec{y}$ are tuples of disjoint variables of the same arity. Furthermore, let $\psi'(\vec{x}, \vec{y})$ be the result of writing $\neg\psi(\vec{x}, \vec{y})$ in negation normal form. Then, for all suitable* *models M and all suitable pairs $\vec{a}$, $\vec{b}$ of constant terms of the model,*

$$M \models_{\{\emptyset\}} \exists \vec{z}(\vec{a} \subseteq \vec{z} \wedge \vec{z} \neq \vec{b} \wedge \forall \vec{w}(\psi'(\vec{z}, \vec{w}) \vee \vec{w} \subseteq \vec{z}))$$

if and only if $M \models \neg[TC_{\vec{x},\vec{y}}\ \psi](\vec{a}, \vec{b})$, that is, if and only if the pair of tuples of elements corresponding to $(\vec{a}, \vec{b})$ is not in the transitive closure of $\{(\vec{m}_1, \vec{m}_2) : M \models \psi(\vec{m}_1, \vec{m}_2)\}$.

Proof. Suppose that $M \models_{\{\emptyset\}} \exists \vec{z}(\vec{a} \subseteq \vec{z} \wedge \vec{z} \neq \vec{b} \wedge \forall \vec{w}(\psi'(\vec{z}, \vec{w}) \vee \vec{w} \subseteq \vec{z}))$. Then, by definition, there exists a tuple of functions $\vec{H} = H_1 \ldots H_n$ such that

1. $M \models_{\{\emptyset\}[\vec{H}/\vec{z}]} \vec{a} \subseteq \vec{z}$, that is, $\vec{a} \in \vec{H}(\{\emptyset\})$;
2. $M \models_{\{\emptyset\}[\vec{H}/\vec{z}]} \vec{z} \neq \vec{b}$, and therefore $\vec{b} \notin \vec{H}(\{\emptyset\})$;
3. $M \models_{\{\emptyset\}[\vec{H}/\vec{z}][\vec{M}/\vec{w}]} \psi'(\vec{z}, \vec{w}) \vee \vec{w} \subseteq \vec{z}$.

Now, the third condition implies that whenever $M \models \psi(\vec{m}_1, \vec{m}_2)$ and $\vec{m}_1$ is in $\vec{H}(\{\emptyset\})$, $\vec{m}_2$ is in $\vec{H}(\{\emptyset\})$ too. Indeed, let $Y = \{\emptyset\}[\vec{H}/\vec{z}][\vec{M}/\vec{w}]$: then, by the semantics of our logic, we know that $Y = Y_1 \cup Y_2$ for two subteams Y_1 and Y_2 such that $M \models_{Y_1} \psi'(\vec{z}, \vec{w})$ and $M \models_{Y_2} \vec{w} \subseteq \vec{z}$. But ψ' is logically equivalent to the negation of ψ, and therefore we know that, for all $s \in Y_1$, $M \not\models \psi(s(\vec{z}), s(\vec{w}))$ in the usual Tarskian semantics.

Suppose now that $\vec{m}_1 \in \vec{H}(\{\emptyset\})$ and that $M \models \psi(\vec{m}_1, \vec{m}_2)$. Then $s = (\vec{z} := \vec{m}_1, \vec{w} := \vec{m}_2)$ is in Y; but it cannot be in Y_1, as we saw, and hence it must belong to Y_2. But $M \models_{Y_2} \vec{w} \subseteq \vec{z}$, and therefore there exists another assignment $s' \in Y_2$ such that $s'(\vec{z}) = s(\vec{w}) = \vec{m}_2$. But we necessarily have that $s'(\vec{z}) \in \vec{H}(\{\emptyset\})$, and therefore $\vec{m}_2 \in \vec{H}(\{\emptyset\})$, as required.

So, $\vec{H}(\{\emptyset\})$ is an set of tuples of elements of our models which contains the interpretation of $\vec{a}$ but not that of $\vec{b}$ and such that

$$\vec{m}_1 \in H(\{\emptyset\}), M \models \psi(\vec{m}_1), \vec{M}_2 \Rightarrow \vec{m}_2 \in H(\{\emptyset\}).$$

This implies that $M \models \neg[\mathrm{TC}_{\vec{x},\vec{y}}\ \psi](\vec{a}, \vec{b})$, as required.

Conversely, suppose that $M \models \neg[\mathrm{TC}_{\vec{x},\vec{y}}\ \psi](\vec{a}, \vec{b})$: then there exists a set A of tuples of elements of the domain of M which contains the interpretation of $\vec{a}$ but not that of $\vec{b}$, and such that it is closed by transitive closure for $\psi(\vec{x}, \vec{y})$. Then, by choosing the functions $\vec{H}$ so that $\vec{h}(\{\emptyset\}) = A$, it is easy to verify that M satisfies our Inclusion Logic sentence. □

As a corollary, we have that Inclusion Logic is strictly more expressive than First Order Logic over sentences: for example, for all finite linear orders $M =$

$(\mathtt{Dom}(M), <, S, 0, e)$, where S is the successor function, 0 is the first element of the linear order and e is the last one, we have that

$$M \models \exists z(0 \subseteq z \wedge z \neq e \wedge \forall w(w \neq S(S(z)) \vee w \subseteq z))$$

if and only if $|M|$ is odd. It is not difficult to see, for example through the Ehrenfeucht-Fraïssé method ([41]), that this property is not expressible in First Order Logic.

4.3.3 Equiextension Logic

Let us now consider *Equiextension Logic*, that is, the logic obtained by adding to First Order Logic equiextension atoms $\vec{t}_1 \bowtie \vec{t}_2$ with the semantics of Definition 4.3.6.

It is easy to see that Equiextension Logic is contained in Inclusion Logic:

Proposition 4.3.14. *Let $\vec{t}_1$ and $\vec{t}_2$ be any two tuples of terms of the same length. Then, for all suitable models M and teams X,*

$$M \models_X \vec{t}_1 \bowtie \vec{t}_2 \Leftrightarrow M \models_X \vec{t}_1 \subseteq \vec{t}_2 \wedge \vec{t}_2 \subseteq \vec{t}_1.$$

Proof. Obvious. □

Translating in the other direction, however, requires a little more care:

Proposition 4.3.15. *Let $\vec{t}_1$ and $\vec{t}_2$ be any two tuples of terms of the same length. Then, for all suitable models M and teams X, $M \models_X \vec{t}_1 \subseteq \vec{t}_2$ if and only if*

$$M \models_X \forall u_1 u_2 \exists \vec{z}(\vec{t}_2 \bowtie \vec{z} \wedge (u_1 \neq u_2 \vee \vec{z} = \vec{t}_1))$$

where u_1, u_2 and $\vec{z}$ do not occur in $\vec{t}_1$ and $\vec{t}_2$.

Proof. Suppose that $M \models_X \vec{t}_1 \subseteq \vec{t}_2$. Then let $X' = X[M/u_1 u_2]$, and pick the tuple of functions $\vec{H}$ used to choose $\vec{z}$ so that

$$\vec{H}(s) = \begin{cases} \{\vec{t}_1\langle s\rangle\}, & \text{if } s(\vec{u}_1) = s(\vec{u}_2); \\ \{\vec{t}_2\langle s\rangle\}, & \text{otherwise} \end{cases}$$

for all $s \in X'$.

Then, for $Y = X'[\vec{H}/\vec{z}]$, by definition we have that $M \models_Y u_1 \neq u_2 \vee \vec{z} = \vec{t}_1$, and it only remains to verify that $M \models_Y \vec{t}_2 \bowtie \vec{z}$, that is, that $Y(\vec{t}_2) = Y(\vec{z})$.

- $Y(\vec{t}_2) \subseteq Y(\vec{z})$:
  Let $h \in Y$. Then there exists an assignment $s \in X$ with $\vec{t}_2\langle s\rangle = \vec{t}_2\langle h\rangle$.

Now let 0 and 1 be two distinct elements of M, and consider the assignment $h' = s[0/u_1][1/u_2][\vec{H}/\vec{z}]$. By construction, $h' \in Y$; and furthermore, by the definition of $\vec{H}$ we have that $h'(\vec{z}) = \vec{t}_2\langle s\rangle = \vec{t}_2\langle h\rangle$, as required.

- $Y(\vec{z}) \subseteq Y(\vec{t}_2)$:
  Let $h \in Y$. Then, by construction, $h(\vec{z})$ is $\vec{t}_1\langle h\rangle$ or $\vec{t}_2\langle h\rangle$. But since $X(\vec{t}_1) \subseteq X(\vec{t}_2)$, in either case there exists an assignment $s \in X$ such $\vec{t}_2\langle s\rangle = h(\vec{z})$. Now consider $h' = s[0/u_1][1/u_2][\vec{H}/\vec{z}]$: again, $h' \in Y$ and $h'(\vec{z}) = \vec{t}_2\langle h'\rangle = \vec{t}_2\langle s\rangle = h(\vec{z})$, as required.

Conversely, suppose that $M \models_X \forall u_1 u_2 \exists \vec{z}(\vec{t}_2 \bowtie \vec{z} \wedge (u_1 \neq u_2 \vee \vec{z} = \vec{t}_1))$, and that therefore there exists a tuple of functions $\vec{H}$ such that, for $Y = X[M/u_1u_2][\vec{H}/\vec{z}]$, $M \models_Y \vec{t}_2 \bowtie \vec{z} \wedge (u_1 \neq u_2 \vee \vec{z} = \vec{t}_1)$. Then consider any assignment $s \in X$, and let $h = s[0/u_1][0/u_2][\vec{H}/\vec{z}]$. Now, $h \in Y$ and $h(\vec{z}) = \vec{t}_1\langle s\rangle$; but since $M \models_Y \vec{t}_2 \bowtie \vec{z}$, this implies that there exists an assignment $h' \in Y$ such that $\vec{t}_2\langle h'\rangle = h(\vec{z}) = \vec{t}_1\langle s\rangle$. Finally, h' derives from some assignment $s' \in X$, and for this assignment we have that $\vec{t}_2\langle s\rangle = \vec{t}_2\langle h'\rangle = \vec{t}_1\langle s\rangle$ as required. □

As a consequence, Inclusion Logic is precisely as expressive as Equiextension Logic:

Corollary 4.3.16. *Any formula of Inclusion Logic is equivalent to some formula of Equiextension Logic, and vice versa.*

4.3.4 Exclusion Logic

With the name of *Exclusion Logic* we refer to First Order Logic supplemented with the *exclusion atoms* $\vec{t}_1 \mid \vec{t}_2$, with the satisfaction condition given in Definition 4.3.5.

As the following results show Exclusion Logic is, in a very strong sense, equivalent to Dependence Logic:

Theorem 4.3.17. *For all tuples of terms $\vec{t}_1$ and $\vec{t}_2$, of the same length, there exists a Dependence Logic formula ϕ such that*

$$M \models_X \phi \Leftrightarrow M \models_X \vec{t}_1 \mid \vec{t}_2$$

for all suitable models M and teams X.

Proof. This follows immediately from Theorem 2.2.14, since the satisfaction condition for the exclusion atom is downwards monotone and expressible in Σ^1_1.

For the sake of completeness, let us write a direct translation of exclusion atoms into Dependence Logic anyway.

Let $\vec{t}_1$ and $\vec{t}_2$ be as in our hypothesis, let $\vec{z}$ be a tuple of new variables, of the same length of $\vec{t}_1$ and $\vec{t}_2$, and let u_1, u_2 be two further unused variables. Then $M \models_X \vec{t}_1 \mid \vec{t}_2$ if and only if

$$M \models_X \forall \vec{z} \exists u_1 u_2 (=(\vec{z}, u_1) \wedge =(\vec{z}, u_2) \wedge ((u_1 = u_2 \wedge \vec{z} \neq \vec{t}_1) \vee (u_1 \neq u_2 \wedge \vec{z} \neq \vec{t}_2))).$$

Indeed, suppose that $M \models_X \vec{t}_1 \mid \vec{t}_2$, let $X' = X[M/\vec{z}]$, and let $0, 1$ be two distinct elements in $\texttt{Dom}(M)$.

Then define the functions H_1 and H_2 as follows:

- For all $s' \in X'$, $H_1(s') = \{0\}$;
- For all $s'' \in X'[H_1/u_1]$, $H_2(s'') = \begin{cases} \{0\} & \text{if } s''(\vec{z}) \notin X(\vec{t}_1); \\ \{1\} & \text{if } s''(\vec{z}) \in X(\vec{t}_1). \end{cases}$

Then, for $Y = X'[H_1 H_2/u_1 u_2]$, we have that $M \models_Y =(\vec{z}, u_1)$ and that $M \models_Y = (\vec{z}, u_2)$, since the value of u_1 is constant in Y and the value of u_2 in Y is functionally determined by the value of $\vec{z}$.

Now split Y into the two subteams Y_1 and Y_2 defined as

$$Y_1 = \{s \in Y : s(u_2) = 0\};$$
$$Y_2 = \{s \in Y : s(u_2) = 1\}.$$

Clearly, $M \models_{Y_1} u_1 = u_2$ and $M \models_{Y_2} u_1 \neq u_2$; hence, we only need to verify that $M \models_{Y_1} \vec{z} \neq \vec{t}_1$ and that $M \models_{Y_2} \vec{z} \neq \vec{t}_2$.

For the first case, let h be any assignment in Y_1: then, by definition, $h(\vec{z}) \neq \vec{t}_1\langle s \rangle$ for all $s \in X$. But then $h(\vec{z}) \neq \vec{t}_1\langle h' \rangle$ for all $h' \in Y_1$, and since this is true for all $h \in Y_1$ we have that $M \models_{Y_1} \vec{z} \neq \vec{t}_1$, as required.

For the second case, let h be in Y_2 instead: then, again by definition, $h(\vec{z}) = \vec{t}_1\langle s \rangle$ for some $s \in X$. But $M \models_X \vec{t}_1 \mid \vec{t}_2$, and hence $h(\vec{z}) \neq \vec{t}_2\langle s' \rangle$ for all $s' \in X$; and as in the previous case, this implies that $h(\vec{z}) \neq \vec{t}_2(h')$ for all $h' \in Y_2$ and, since this argument can be made for all $h \in Y_2$, $M \models_{Y_2} \vec{z} \neq \vec{t}_2$.

Conversely, suppose that

$$M \models_X \forall \vec{z} \exists u_1 u_2 (=(\vec{z}, u_1) \wedge =(\vec{z}, u_2) \wedge ((u_1 = u_2 \wedge \vec{z} \neq \vec{t}_1) \vee (u_1 \neq u_2 \wedge \vec{z} \neq \vec{t}_2))).$$

Then there exist two functions H_1 and H_2 such that, for $Y = X[M/\vec{z}][H_1 H_2/u_1 u_2]$,

$$M \models_Y =(\vec{z}, u_1) \wedge =(\vec{z}, u_2) \wedge ((u_1 = u_2 \wedge \vec{z} \neq \vec{t}_1) \vee (u_1 \neq u_2 \wedge \vec{z} \neq \vec{t}_2)).$$

Now, let s_1 and s_2 be any two assignments in X: in order to conclude the proof, I only need to show that $\vec{t}_1\langle s_1\rangle \neq \vec{t}_2\langle s_2\rangle$. Suppose instead that $\vec{t}_1\langle s_1\rangle = \vec{t}_2\langle s_2\rangle = \vec{m}$ for some tuple of elements $\vec{m}$, and consider two assignments h_1, h_2 such that

$$h_1 \in \{s_1[\vec{m}/\vec{z}]\}[H_1H_2/u_1u_2];^{9}$$

and

$$h_2 \in \{s_2[\vec{m}/\vec{z}]\}[H_1H_2/u_1u_2].$$

Then $h_1, h_2 \in Y$; and furthermore, since $h_1(\vec{z}) = h_2(\vec{z})$ and $M \models =(\vec{z}, u_1) \wedge =(\vec{z}, u_2)$, it must hold that $h_1(\vec{u}_1) = h_2(\vec{u}_1)$ and $h_1(\vec{u}_2) = h_2(\vec{u}_2)$.

Moreover, $M \models_Y (u_1 = u_2 \wedge \vec{z} \neq \vec{t}_1) \vee (u_1 \neq u_2 \wedge \vec{z} \neq \vec{t}_2)$, and therefore Y can be split into two subteams Y_1 and Y_2 such that

$$M \models_{Y_1} (u_1 = u_2 \wedge \vec{z} \neq \vec{t}_1)$$

and

$$M \models_{Y_2} (u_1 \neq u_2 \wedge \vec{z} \neq \vec{t}_2).$$

Now, as we saw, the assignments h_1 and h_2 coincide over u_1 and u_2, and hence either $\{h_1, h_2\} \subseteq Y_1$ or $\{h_1, h_2\} \subseteq Y_2$. But neither case is possible, because

$$h_1(\vec{z}) = \vec{m} = \vec{t}_1\langle s_1\rangle = \vec{t}_1\langle h_1\rangle$$

and therefore h_1 cannot be in Y_1, and because

$$h_2(\vec{z}) = \vec{m} = \vec{t}_2\langle s_2\rangle = \vec{t}_2\langle h_2\rangle$$

and therefore h_2 cannot be in Y_2.

So we reached a contradiction, and this concludes the proof. □

Theorem 4.3.18. *Let $t_1 \dots t_n$ be terms, and let z be a variable not occurring in any of them. Then the dependence atom $=(t_1 \dots t_n)$ is equivalent to the Exclusion Logic expression*

$$\phi = \forall z(z = t_n \vee (t_1 \dots t_{n-1}z \mid t_1 \dots t_{n-1}t_n)),$$

for all suitable models M and teams X.

Proof. Suppose that $M \models_X =(t_1 \dots t_n)$, and consider the team $X[M/z]$. Now, let $Y = \{s \in X[M/z] : s(z) = t_n\langle s\rangle\}$ and let $Z = X[M/z] \backslash Y$.

[9] This team and the next one are actually singletons, because H_1 and H_2 must satisfy the dependency conditions.

Clearly, $Y \cup Z = X[M/x]$ and $M \models_Y z = t_n$; hence, if we show that $Z \models t_1 \ldots t_{n-1} z \mid t_1 \ldots t_{n-1} t_n$ we can conclude that $M \models_X \phi$, as required.

Now, consider any two $s, s' \in Z$, and suppose that $t_i\langle s \rangle = t_i\langle s' \rangle$ for all $i = 1 \ldots n-1$. But then $s(z) \neq t_n\langle s' \rangle$: indeed, since $M \models_X =\!(t_1 \ldots t_n)$, by the locality of Dependence Logic and by the downwards closure property we have that $M \models_Z =\!(t_1 \ldots t_n)$ and hence that $t_n\langle s \rangle = t_n\langle s' \rangle$.

Therefore, if we had that $s(z) = t_n\langle s' \rangle$, it would follow that $s(z) = t_n\langle s' \rangle = t_n\langle s \rangle$ and s would be in Y instead.

So $s(z) \neq t_n\langle s' \rangle$, and since this holds for all s and s' in Z which coincide over $t_1 \ldots t_{n-1}$ we have that

$$M \models_Z t_1 \ldots t_{n-1} z \mid t_1 \ldots t_{n-1} t_n,$$

as required.

Conversely, suppose that $M \models_X \phi$, and let $s, s' \in X$ assign the same values to $t_1 \ldots t_{n-1}$. Now, by the definition of ϕ, $X[M/z]$ can be split into two subteams Y and Z such that $M \models_Y z = t_n$ and
$M \models_Z (t_1 \ldots t_{n-1} z \mid t_1 \ldots t_{n-1} t_n)$.

Now, suppose that $t_n\langle s \rangle = m$ and $t_n\langle s' \rangle = m'$, and that $m \neq m'$: then $s[m'/z]$ and $s'[m/z]$ are in $s[M/z]$ but not in Y, and hence they are both in Z. But then, since $\vec{t}_i\langle s \rangle = \vec{t}_i\langle s' \rangle$ for all $i = 1 \ldots n-1$,

$$t_n\langle s' \rangle = m' = s[m'/z](z) \neq t_n\langle s'[m/z] \rangle = t_n\langle s' \rangle$$

which is a contradiction. Therefore, $m = m'$, as required. □

Corollary 4.3.19. *Dependence Logic is precisely as expressive as Exclusion Logic, both with respect to definability of sets of teams and with respect to sentences.*

4.3.5 Inclusion/Exclusion Logic

Now that we have some information about Inclusion Logic and about Exclusion Logic, let us study *Inclusion/Exclusion Logic* (I/E logic for short), that is, the formalism obtained by adding both inclusion and exclusion atoms to the language of First Order Logic.

By the results of the previous sections, we already know that inclusion atoms are expressible in Independence Logic and that exclusion atoms are expressible in Dependence Logic; furthermore, as we saw in Subsection 2.4.1, dependence atoms are expressible in Independence Logic.

Then it follows at once that I/E Logic is contained in Independence Logic:

Corollary 4.3.20. *For every I/E Logic formula ϕ there exists an Independence Logic formula ϕ^* such that*

$$M \models_X \phi \Leftrightarrow M \models_X \phi^*$$

for all suitable models M and teams X.

Now, is I/E Logic properly contained in Independence Logic?

As the following result illustrates, this is not the case:

Theorem 4.3.21. *Let $\vec{t}_2 \perp_{\vec{t}_1} \vec{t}_3$ be an independence atom, and let ϕ be the formula*

$$\forall \vec{p}\vec{q}\vec{r}\, \exists u_1 u_2 u_3 u_4 \left(\bigwedge_{i=1}^{4} =\!(\vec{p}\vec{q}\vec{r}, u_i) \wedge ((u_1 \neq u_2 \wedge (\vec{p}\vec{q} \mid \vec{t}_1\vec{t}_2)) \vee \right.$$
$$\left. \vee (u_1 = u_2 \wedge u_3 \neq u_4 \wedge (\vec{p}\vec{r} \mid \vec{t}_1\vec{t}_3)) \vee (u_1 = u_2 \wedge u_3 = u_4 \wedge (\vec{p}\vec{q}\vec{r} \subseteq \vec{t}_1\vec{t}_2\vec{t}_3)))\right)$$

where the dependence atoms are used as shorthands for the corresponding Exclusion Logic expressions, which exist because of Theorem 4.3.18, and where all the quantified variables are new.

Then, for all suitable models M and teams X,

$$M \models_X \vec{t}_2 \perp_{\vec{t}_1} \vec{t}_3 \Leftrightarrow M \models_X \phi.$$

Proof. Suppose that $M \models_X \vec{t}_2 \perp_{\vec{t}_1} \vec{t}_3$, and consider the team $X' = X[M/\vec{p}\vec{q}\vec{r}]$.

Now, let 0 and 1 be two distinct elements of the domain of M, and let the functions $H_1 \ldots H_4$ be defined as follows:

- For all $s \in X'$, $H_1(s) = \{0\}$;
- For all $s \in X'[H_1/u_1]$,

$$H_2(s) = \begin{cases} \{0\} & \text{if there exists a } s' \in X \text{ such that } \vec{t}_1\langle s'\rangle\vec{t}_2\langle s'\rangle = s(\vec{p})s(\vec{q}); \\ \{1\} & \text{otherwise;} \end{cases}$$

- For all $s \in X'[H_1/u_1][H_2/u_2]$, $H_3(s) = \{0\}$;
- For all $s \in X'[H_1/u_1][H_2/u_2][H_3/u_3]$,

$$H_4(s) = \begin{cases} \{0\} & \text{if there exists a } s' \in X \text{ such that } \vec{t}_1\langle s'\rangle\vec{t}_3\langle s'\rangle = s(\vec{p})s(\vec{r}); \\ \{1\} & \text{otherwise.} \end{cases}$$

Now, let $Y = X'[H_1/u_1][H_2/u_2][H_3/u_3][H_4/u_4]$: by the definitions of $H_1 \ldots H_4$, it holds that all dependencies are respected. Let then Y be split into Y_1, Y_2 and Y_3 according to:

- $Y_1 = \{s \in Y : s(u_1) \neq s(u_2)\}$;
- $Y_2 = \{s \in Y : s(u_3) \neq s(u_4)\} \backslash Y_1$;
- $Y_3 = Y \backslash (Y_1 \cup Y_2)$.

Now, let s be any assignment of Y_1: then, since $s(u_1) \neq s(u_2)$, by the definitions of H_1 and H_2 we have that

$$\forall s' \in Y, s(\vec{p})s(\vec{q}) \neq \vec{t}_1\langle s'\rangle\vec{t}_2\langle s'\rangle$$

and, in particular, that the same holds for all the $s' \in Y_1$. Hence,

$$M \models_{Y_1} u_1 \neq u_2 \wedge (\vec{p}\vec{q} \mid \vec{t}_1\vec{t}_2),$$

as required.

Analogously, let s be any assignment of Y_2: then $s(u_1) = s(u_2)$, since otherwise s would be in Y_1, $s(u_3) \neq s(u_4)$ and

$$\forall s' \in Y, s(\vec{p})s(\vec{r}) \neq \vec{t}_1\langle s'\rangle\vec{t}_3\langle s'\rangle$$

and therefore

$$M \models_{Y_2} u_1 = u_2 \wedge u_3 \neq u_4 \wedge (\vec{p}\vec{r} \mid \vec{t}_1\vec{t}_3).$$

Finally, suppose that $s \in Y_3$: then, by definition, $s(u_1) = s(u_2)$ and $s(u_3) = s(u_4)$. Therefore, there exist two assignments s' and s'' in X such that

$$\vec{t}_1\langle s'\rangle\vec{t}_2\langle s'\rangle = s(\vec{p})s(\vec{q})$$

and

$$\vec{t}_1\langle s''\rangle\vec{t}_3\langle s''\rangle = s(\vec{p})s(\vec{r})$$

But by hypothesis we know that $M \models_X \vec{t}_2 \perp_{\vec{t}_1} \vec{t}_3$, and s' and s'' coincide over $\vec{t}_1$, and therefore there exists a new assignment $h \in X$ such that

$$\vec{t}_1\langle h\rangle\vec{t}_2\langle h\rangle\vec{t}_3\langle h\rangle = s(\vec{p})s(\vec{q})s(\vec{r}).$$

Now, let o be the assignment of Y given by

$$o = h[\vec{t}_1\langle h\rangle\vec{t}_2\langle h\rangle\vec{t}_3\langle h\rangle/\vec{p}\vec{q}\vec{r}][H_1 \ldots H_4/u_1 \ldots u_4] :$$

by the definitions of $H_1 \dots H_4$ and by the construction of o, we then get that

$$o(u_1) = o(u_2) = o(u_3) = o(u_4) = 0$$

and therefore that $o \in Y_3$.

But by construction,

$$\vec{t}_1\langle o\rangle \vec{t}_2\langle o\rangle \vec{t}_3\langle o\rangle = \vec{t}_1\langle h\rangle \vec{t}_2\langle h\rangle \vec{t}_3\langle h\rangle = s(\vec{p})s(\vec{q})s(\vec{r}),$$

and hence

$$M \models_{Y_3} \vec{p}\vec{q}\vec{r} \subseteq \vec{t}_1\vec{t}_2\vec{t}_3$$

as required.

Conversely, suppose that $M \models_X \phi$, and let $s, s' \in X$ be such that $\vec{t}_1\langle s\rangle = \vec{t}_1\langle s'\rangle$. Now, consider the two assignments $h, h' \in X' = X[M/\vec{p}\vec{q}\vec{r}]$ given by

$$h = s[\vec{t}_1\langle s\rangle/\vec{p}][\vec{t}_2\langle s\rangle/\vec{q}][\vec{t}_3\langle s'\rangle/\vec{r}]$$

and

$$h' = s'[\vec{t}_1\langle s\rangle/\vec{p}][\vec{t}_2\langle s\rangle/\vec{q}][\vec{t}_3\langle s'\rangle/\vec{r}].$$

Now, since $M \models_X \phi$, there exist functions $H_1 \dots H_4$, depending only on $\vec{p}$, $\vec{q}$ and $\vec{r}$, such that $Y = X'[H_1/u_1][H_2/u_2][H_3/u_3][H_4/u_4]$ can be split into three subteams Y_1, Y_2 and Y_3 and

$$\begin{aligned} &M \models_{Y_1} (u_1 \neq u_2 \wedge (\vec{p}\vec{q} \mid \vec{t}_1\vec{t}_2));\\ &M \models_{Y_2} (u_1 = u_2 \wedge u_3 \neq u_4 \wedge (\vec{p}\vec{r} \mid \vec{t}_1\vec{t}_3));\\ &M \models_{Y_3} (u_1 = u_2 \wedge u_3 = u_4 \wedge (\vec{p}\vec{q}\vec{r} \subseteq \vec{t}_1\vec{t}_2\vec{t}_3)). \end{aligned}$$

Now, let

$$o \in h[H_1/u_1][H_2/u_2][H_3/u_3][H_4/u_4]$$

and

$$o' \in h'[H_1/u_1][H_2/u_2][H_3/u_3][H_4/u_4]:$$

since the H_i are functionally dependent on $\vec{p}\vec{q}\vec{r}$ and the values of these variables are the same for h and for h', we have that o and o' have the same values for $u_1 \dots u_4$, and therefore that they belong to the same Y_i.

But they cannot be in Y_1 nor in Y_2, since

$$o(\vec{p})o(\vec{q}) = o'(\vec{p})o'(\vec{q}) = \vec{t}_1\langle s\rangle \vec{t}_2\langle s\rangle = \vec{t}_1\langle o\rangle \vec{t}_2\langle o\rangle$$

and

$$o(\vec{p})o(\vec{r}) = o'(\vec{p})o'(\vec{r}) = \vec{t}_1\langle s'\rangle\vec{t}_3\langle s'\rangle = \vec{t}_1\langle o'\rangle\vec{t}_3\langle o'\rangle;$$

therefore, o and o' are in Y_3, and there exists an assignment $o'' \in Y_3$ with

$$\vec{t}_1\langle o''\rangle\vec{t}_2\langle o''\rangle\vec{t}_3\langle o''\rangle = o(\vec{p})o(\vec{q})o(\vec{r}) = \vec{t}_1\langle s\rangle\vec{t}_2\langle s\rangle\vec{t}_3\langle s'\rangle$$

and, finally, there exists a $s'' \in X$ such that $\vec{t}_1\langle s''\rangle\vec{t}_2\langle s''\rangle\vec{t}_3\langle s''\rangle = \vec{t}_1\langle s\rangle\vec{t}_2\langle s\rangle\vec{t}_3\langle s'\rangle$, as required. □

Independence Logic and I/E Logic are therefore equivalent:

Corollary 4.3.22. *Any Independence Logic formula is equivalent to some I/E Logic formula, and any I/E Logic formula is equivalent to some Independence Logic formula.*

Figure 4.2 summarizes the translatability[10] relations between the logics of imperfect information which have been considered in this work.

Let us finish this section verifying that I/E Logic (and, as a consequence, also Inclusion Logic, Equiextension Logic and Independence Logic) with the lax semantics is local:

Theorem 4.3.23. *Let M be a first order model, let ϕ be any I/E Logic formula and let V be a set of variables such that $\texttt{Free}(\phi) \subseteq V$. Then, for all suitable teams X,*

$$M \models_X \phi \Leftrightarrow M \models_{X_{\upharpoonright V}} \phi$$

Proof. The proof is by structural induction on ϕ.

1. If ϕ is a first order literal, an inclusion atom or an exclusion atom then the statement follows trivially from the corresponding semantic rule;

2. Let ϕ be of the form $\psi \vee \theta$, and suppose that $M \models_X \psi \vee \theta$. Then, by definition, $X = Y \cup Z$ for two subteams Y and Z such that $M \models_Y \psi$ and $M \models_Z \theta$. Then, by induction hypothesis, $M \models_{Y_{\upharpoonright V}} \psi$ and $M \models_{Z_{\upharpoonright V}} \theta$. But $X_{\upharpoonright V} = Y_{\upharpoonright V} \cup Z_{\upharpoonright V}$: indeed, $s \in X$ if and only if $s \in Y$ or $s \in Z$, and hence $s_{\upharpoonright V} \in X_{\upharpoonright V}$ if and only if it is in $Y_{\upharpoonright V}$ or in $Z_{\upharpoonright V}$. Hence, $M \models_{X_{\upharpoonright V}} \psi \vee \theta$, as required.

   Conversely, suppose that $M \models_{X_{\upharpoonright V}} \psi \vee \theta$, that is, that $X_{\upharpoonright V} = Y' \cup Z'$ for two subteams Y' and Z' such that $M \models_{Y'} \psi$ and $M \models_{Z'} \theta$. Then

[10] To be more accurate, Figure 4.2 represents the translatability relations between the logics which we considered, *with respect to all formulas*. Considering sentences only would lead to a different graph.

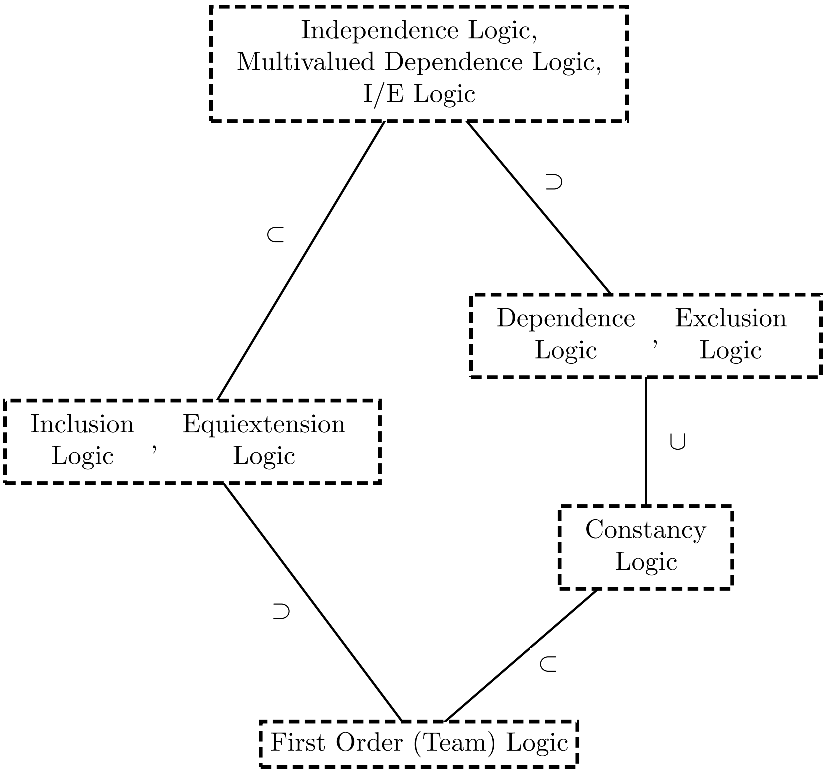

Figure 4.2: Relations between logics of imperfect information (wrt formulas)

define $Y = \{s \in X : s_{\upharpoonright V} \in Y'\}$ and $Z = \{s \in X : s_{\upharpoonright V} \in Z'\}$. Now, $X = Y \cup Z$: indeed, if $s \in X$ then $s_{\upharpoonright V}$ is in $X_{\upharpoonright V}$, and hence it is in Y' or in Z', and on the other hand if s is in Y or in Z then it is in X by definition. Furthermore, $Y_{\upharpoonright V} = Y'$ and $Z_{\upharpoonright V} = Z'$,[11] and hence by induction hypothesis $M \models_Y \psi$ and $M \models_Z \theta$, and finally $M \models_X \psi \vee \theta$.

3. Let ϕ be of the form $\psi \wedge \theta$. Then $M \models_X \psi \wedge \theta$ if and only if $M \models_X \psi$ and $M \models_X \theta$, that is, by induction hypothesis, if and only if $M \models_{X_{\upharpoonright V}} \psi$ and $M \models_{X_{\upharpoonright V}} \theta$. But this is the case if and only if $M \models_{X_{\upharpoonright V}} \psi \wedge \theta$, as required.

4. Let ϕ be of the form $\exists x \psi$, and suppose that $M \models_X \exists x \psi$. Then there exists a function $H : X \to \texttt{Parts}(\texttt{Dom}(M))\backslash\{\emptyset\}$ such that $M \models_{X[H/x]} \psi$. Then, by induction hypothesis, $M \models_{(X[H/x])_{\upharpoonright V \cup \{x\}}} \psi$.

   Now consider the function $H' : X_{\upharpoonright V} \to \texttt{Parts}(\texttt{Dom}(M))\backslash\emptyset$ which assigns to every $s' \in X_{\upharpoonright V}$ the set

   $$H'(s') = \bigcup\{H(s) : s \in X, s' = s_{\upharpoonright V}\}.$$

   Then H' assigns a nonempty set to every $s' \in X_{\upharpoonright V}$, as required; and furthermore, $X_{\upharpoonright V}[H'/x]$ is precisely $(X[H/x])_{\upharpoonright V \cup \{x\}}$.[12] Therefore, $M \models_{X_{\upharpoonright V}} \exists x \psi$, as required.

   Conversely, suppose that $M \models_{X_{\upharpoonright V}} \exists x \psi$, that is, that $M \models_{X_{\upharpoonright V}[H'/x]} \psi$ for some H'. Then define the function $H : X \to \texttt{Parts}(\texttt{Dom}(M))\backslash\{x\}$ so that $H(s) = H'(s_{\upharpoonright V})$ for all $s \in X$; now, $X_{\upharpoonright V}[H'/x] = (X[H/x])_{\upharpoonright V \cup \{x\}}$,[13] and hence by induction hypothesis $M \models_X \exists x \psi$.

5. For all suitable teams X, $X[M/x]_{\upharpoonright V \cup \{x\}} = X_{\upharpoonright V}[M/x]$; and hence, $M \models_{X_{\upharpoonright V}} \forall x \psi \Leftrightarrow M \models_{X[M/x]_{\upharpoonright V \cup \{x\}}} \psi \Leftrightarrow M \models_{X[M/x]} \psi \Leftrightarrow M \models_X \forall x \psi$, as required.

□

[11] By definition, $Y_{\upharpoonright V} \subseteq Y'$ and $Z_{\upharpoonright V} \subseteq Z'$. On the other hand, if $s' \in Y'$ then $s' \in X_{\upharpoonright V}$, and hence s' is of the form $s_{\upharpoonright V}$ for some $s \in X$, and therefore this s is in Y too, and finally $s' = s_{\upharpoonright V} \in Y_{\upharpoonright V}$. The same argument shows that $Z' \subseteq Z_{\upharpoonright V}$.

[12] Indeed, suppose that $s' \in X[H/x]$: then there exists a $s \in X$ such that $s' = s[m/x]$ for some $m \in H(s)$. Then $s_{\upharpoonright V} \in X_{\upharpoonright V}$, and moreover $m \in H'(s_{\upharpoonright V})$ by the definition of H', and hence $s'_{\upharpoonright V \cup \{x\}} = s_{\upharpoonright V}[m/x] \in X_{\upharpoonright V}[H'/x]$.

Conversely, suppose that $h' \in X_{\upharpoonright V}[H'/x]$: then there exists a $h \in X_{\upharpoonright V}$ such that $h' = h[m/x]$ for some $m \in H'(h)$. But then there exists a $s \in X$ such that $h = s_{\upharpoonright V}$ and such that $m \in H(s)$; and therefore, $s[m/x] \in X[H/x]$, and finally $h' = h[m/x] = (s[m/x])_{\upharpoonright V \cup \{x\}} \in (X[H/x])_{\upharpoonright V \cup \{x\}}$.

[13] In brief, for all $s \in X$ and all $m \in \texttt{Dom}(M)$ we have that $m \in H'(s_{\upharpoonright V})$ if and only if $m \in H(s)$, by definition. Hence, for all such s and m, $s_{\upharpoonright V}[m/x] \in X_{\upharpoonright V}[H'/x]$ if and only if $s[m/x] \in X[H/x]$.

4.4 Game Theoretic Semantics for I/E Logic

In this section, we will adapt the Game Theoretic Semantics of Subsection 2.2.3 to the case of Inclusion/Exclusion Logic.

As for the case of dependence atoms, we will fix

$$\texttt{Player}(\vec{t}_1 \subseteq \vec{t}_2) = \texttt{Player}(\vec{t}_1 \mid \vec{t}_2, s) = \texttt{E};$$
$$\texttt{Succ}(\vec{t}_1 \subseteq \vec{t}_2) = \texttt{Succ}(\vec{t}_1 \mid \vec{t}_2, s) = (\lambda, s).$$

The uniformity condition will be changed in the obvious way:

Definition 4.4.1. 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

1. For all $\vec{p} \in P$ and for all $i \in \mathbb{N}$ such that $p_i = (\vec{t}_1 \subseteq \vec{t}_2, s)$ there exists a $\vec{q} \in P$ and a $j \in \mathbb{N}$ such that $q_j = (\vec{t}_1 \subseteq \vec{t}_2, s')$ for the same instance of the inclusion atom and $\vec{t}_1\langle s\rangle = \vec{t}_2\langle s'\rangle$;

2. For all $\vec{p}, \vec{q} \in P$ and for all $i, j \in \mathbb{N}$ such that $p_i = (\vec{t}_1 \mid \vec{t}_2, s)$ and $p_j = (\vec{t}_1 \mid \vec{t}_2, s')$ for the same instance of the exclusion atom, $\vec{t}_1\langle s\rangle \neq \vec{t}_2\langle s'\rangle$.

The other modification which we need to make, in order to account for the **TS-∃-lax** rule, is that we must now be able to consider *nondeterministic* strategies:

Definition 4.4.2. Let $G_X^M(\phi)$ be a semantic game and let ψ be any expression such that (ψ, s') is a possible position for some s'. Then a *nondeterministic local strategy* for ψ is a function $\mathfrak{f}_\psi$ sending each s' into a nonempty subset of $\texttt{Succ}_M(\psi, s')$.

Definition 4.4.3. Let $G_X^M(\phi)$ be a semantic game, let $\vec{p} = p_1 \ldots p_n$ be a play in it, and let $\mathfrak{f}_\psi$ be a local strategy for some ψ. Then $\vec{p}$ is said to *follow* $\mathfrak{f}_\psi$ if and only if for all $i \in 1 \ldots n-1$ and all s',

$$p_i = (\psi, s') \Rightarrow p_{i+1} \in \mathfrak{f}_\psi(s').$$

A global nondeterministic strategy for a game is simply a collection of local nondeterministic strategies for all positions of the game in which E moves, and such a strategy is said to be uniform or winning if and only if the set of all complete plays in which E follows it is so.

Once these modifications are made, we can easily generalize Theorem 2.2.28 to I/E Logic:

Theorem 4.4.4. *Let M be a first-order model, let X be a team, and let ϕ be any I/E Logic formula. Then $M \models_X \phi$ if and only if the existential player* `E` *has a uniform winning strategy for $G^M_X(\phi)$.*

Proof. The proof is by structural induction on ϕ, and follows exactly the same pattern of the proof of Theorem 2.2.28.

We report here only the cases in which some modification is necessary:

2. If ϕ is an inclusion atom $\vec{t_1} \subseteq \vec{t_2}$ then the only strategy available to `E` sends each initial position $(\vec{t_1} \subseteq \vec{t_2}, s)$ into the winning terminal position (λ, s). This strategy is uniform if and only if the set X of all initial assignments satisfies the inclusion atom, as required.

   The case of the exclusion atom is entirely analogous.

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 nondeterministic, winning uniform strategies $\mathfrak{f}_1$ and $\mathfrak{f}_2$ for `E` in $G^M_{X_1}(\psi_1)$ and $G^M_{X_2}(\psi_2)$ respectively. Then define the strategy $\mathfrak{f}$ for `E` in $G^M_X(\psi_1 \vee \psi_2)$ as follows:

   - If θ is part of ψ_1 then $\mathfrak{f}_\theta = (\mathfrak{f}_1)_\theta$;
   - If θ is part of ψ_2 then $\mathfrak{f}_\theta = (\mathfrak{f}_2)_\theta$;
   - If θ is the initial formula $\psi_1 \vee \psi_2$ then

$$\mathfrak{f}_\theta(s) = \begin{cases} \{(\psi_1, s), (\psi_2, s)\} & \text{if } s \in X_1 \cap X_2; \\ \{(\psi_1, s)\} & \text{if } s \in X_1 \backslash X_2; \\ \{(\psi_2, s)\} & \text{if } s \in X_2 \backslash X_1; \end{cases}$$

   This strategy is clearly uniform, as any violation of the uniformity condition would be a violation for $\mathfrak{f}_1$ or $\mathfrak{f}_2$ too.[14] Furthermore, it is winning: indeed, any play of $G^M_X(\psi_1 \vee \psi_2)$ in which `E` follows $\mathfrak{f}$ strictly contains a play of $G^M_{X_1}(\psi_1)$ in which `E` follows $\mathfrak{f}_1$ or a play of $G^M_{X_2}(\psi_2)$ in which `E` follows $\mathfrak{f}_2$, and in either case the game ends in a winning position.

   Conversely, suppose that $\mathfrak{f}$ is a nondeterministic uniform winning strategy for `E` in $G^M_X(\phi)$. Now let $X_1 = \{s \in X : (\psi_1, s) \in \mathfrak{f}_\phi(s)\}$, let $X_2 = \{s \in X : (\psi_2, s) \in \mathfrak{f}_\phi(s)\}$, and let $\mathfrak{f}_1$ and $\mathfrak{f}_2$ be the restrictions of $\mathfrak{f}$ to the subgames corresponding to ψ_1 and ψ_2 respectively. Then $\mathfrak{f}_1$ and $\mathfrak{f}_2$ are uniform and

[14] Note that here it is vital that *all* possible plays of $G^M_{X_i}(\psi_i)$ in which `E` follows $\mathfrak{f}_i$ are part of some possible play of $G^M_X(\psi)$ in which `E` follows $\mathfrak{f}$. Otherwise, it would not be guaranteed that the uniformity conditions corresponding to inclusion atoms are respected.

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 $M \models_X \phi$.

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

   - If θ is part of ψ then $\mathfrak{f}_\theta = \mathfrak{g}_\theta$;
   - $\mathfrak{f}_\phi(\exists v \psi, s) = \{(\psi, s[m/v]) : m \in H(s)\}$.

   Then any play of $G^M_X(\phi)$ in which $\texttt{E}$ follows $\mathfrak{f}$ contains a play of $G^M_{X[H/v]}(\psi)$ in which $\texttt{E}$ follows $\mathfrak{g}$, and every such play is contained in some play following $\mathfrak{f}$ as above; and hence, $\mathfrak{f}$ is uniform and winning.

   Conversely, suppose that $\texttt{E}$ has a nondeterministic uniform winning strategy $\mathfrak{f}$ for $G^M_X(\exists v \psi)$. Then define the function $H : X \to \texttt{Parts}(\texttt{Dom}(M)) \backslash \{\emptyset\}$ so that for all $s \in X$, $\mathfrak{f}_\phi(\exists v \psi, s) = \{(\psi, s[m/v]) : m \in H(s)\}$, and let $\mathfrak{g}$ be the restriction of $\mathfrak{f}$ to ψ. Then $\mathfrak{g}$ is winning and uniform for $G^M_{X[H/v]}(\psi)$, and hence by induction hypothesis $M \models_{X[H/v]} \psi$, and finally $M \models_X \exists v \psi$.

□

In [24], Forster considers the distinction between deterministic and nondeterministic strategies for the case of the logic of branching quantifiers and points out that, in the absence of the Axiom of Choice, different truth conditions are obtained for these two cases. In the same paper, he then suggests that

> Perhaps advocates of branching quantifier logics and their descendents will tell us which semantics [**that is, the deterministic or nondeterministic one**] they have in mind.

Dependence Logic, Inclusion Logic, Inclusion/Exclusion Logic and Independence Logic can certainly be seen as descendents of Branching Quantifier Logic, and the present work strongly suggests that the semantics that we "have in mind" is the nondeterministic one. As we have just seen, the deterministic/nondeterministic distinction in Game Theoretic Semantics corresponds precisely to the strict/lax distinction in Team Semantics; and indeed, for Dependence Logic proper (which is expressively equivalent to branching quantifier logic), the lax and strict semantics are equivalent modulo the Axiom of Choice.

But for Inclusion Logic and its extensions, we have that lax and strict (and, hence, nondeterministic and deterministic) semantics are not equivalent, even in the presence of the Axiom of Choice (Proposition 4.3.7), and that only the lax one satisfies Locality in the sense of Proposition 2.2.7 (see Proposition 4.3.8 and Theorems 4.3.9, 4.3.23 for the proof).

Furthermore, as stated before, Engström showed in [19] that the lax semantics for existential quantification arises naturally from his treatment of generalized quantifiers in Dependence Logic.

All of this, in the opinion of the author at least, makes a convincing case for the adoption of the nondeterministic semantics (or, in terms of Team Semantics, of the lax one) as the natural semantics for the study of logics of imperfect information, thus suggesting an answer to Forster's question.

4.5 Definability in I/E Logic (and in Independence Logic)

As we wrote in Subsection 2.2.2, in [50] Kontinen and Väänänen characterized the expressive power of dependence Logic formulas, and, in [49], Kontinen and Nurmi used a similar technique to prove that a class of teams is definable in Team Logic (Subsection 2.4.3) if and only if it is expressible in full Second Order Logic.

In this section, I will attempt to find an analogous result for I/E Logic (and hence, through Corollary 4.3.22, for Independence Logic). One direction of the intended result is straightforward:

Theorem 4.5.1. *Let $\phi(\vec{v})$ be a formula of I/E Logic with free variables in $\vec{v}$. Then there exists an existential second order Logic formula $\Phi(A)$, where A is a second order variable with arity $|\vec{v}|$, such that*

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

for all suitable models M and teams X.

Proof. The proof is an unproblematic induction over the formula ϕ, and follows closely the proof of the analogous results for dependence Logic ([65]) or independence Logic ([33]). □

The other direction, by contrast, requires some care:[15]

Theorem 4.5.2. *Let $\Phi(A)$ be a formula in Σ^1_1 such that $\mathtt{Free}(\Phi) = \{A\}$, and let $\vec{v}$ be a tuple of distinct variables with $|\vec{v}| = Arity(A)$. Then there exists an*

[15] The details of this proof are similar to the ones of [50] and [49].

I/E Logic formula $\phi(\vec{v})$ such that

$$M \models_X \phi(\vec{v}) \Leftrightarrow M \models \Phi(\texttt{Rel}_{\vec{v}}(X))$$

for all suitable models M and nonempty teams X.

Proof. It is easy to see that any $\Phi(A)$ as in our hypothesis is equivalent to the formula

$$\Phi^*(A) = \exists B(\forall \vec{x}(A\vec{x} \leftrightarrow B\vec{x}) \wedge \Phi(B)),$$

in which the variable A occurs only in the conjunct $\forall \vec{x}(A\vec{x} \leftrightarrow B\vec{x})$. Then, as in [50], it is possible to write $\Phi^*(A)$ in the form

$$\exists \vec{f}\, \forall \vec{x}\vec{y}((A\vec{x} \leftrightarrow f_1(\vec{x}) = f_2(\vec{x})) \wedge \psi(\vec{x}, \vec{y}, \vec{f})),$$

where $\vec{f} = f_1 f_2 \ldots f_n$, $\psi(\vec{f}, x, y)$ is a quantifier-free formula in which A does not appear, and each f_i occurs only as $f(\vec{w}_i)$ for some fixed tuple of variables $\vec{w}_i \subseteq \vec{x}\vec{y}$.

Now define the formula $\phi(\vec{v})$ as

$$\forall \vec{x}\vec{y}\, \exists \vec{z} \left(\bigwedge_i =\!(\vec{w}_i, z_i) \wedge (((\vec{v} \subseteq \vec{x} \wedge z_1 = z_2) \vee (\vec{v} \mid \vec{x} \wedge z_1 \neq z_2)) \wedge \psi'(\vec{x}, \vec{y}, \vec{z})) \right),$$

where $\psi'(\vec{x}, \vec{y}, \vec{z})$ is obtained from $\psi(\vec{x}, \vec{y}, \vec{f})$ by substituting each $f_i(\vec{w}_i)$ with z_i, and the dependence atoms are used as shorthands for the corresponding expressions of I/E Logic.

Now we have that $M \models_X \phi(\vec{v}) \Leftrightarrow M \models \Phi^*(\texttt{Rel}_{\vec{v}}(X))$:

Indeed, suppose that $M \models_X \phi(\vec{v})$. Then, by construction, for each $i = 1 \ldots n$ there exists a function H_i, choosing precisely one element for possible value of $\vec{w}_i$, such that for $Y = X[M/\vec{x}\vec{y}][\vec{H}/\vec{z}]$

$$M \models_Y ((\vec{v} \subseteq \vec{x} \wedge z_1 = z_2) \vee (\vec{v} \mid \vec{x} \wedge z_1 \neq z_2)) \wedge \psi'(\vec{x}, \vec{y}, \vec{z}).$$

Therefore, we can split Y into two subteams Y_1 and Y_2 such that $M \models_{Y_1} \vec{v} \subseteq \vec{x} \wedge z_1 = z_2$ and $M \models_{Y_2} \vec{v} \mid \vec{x} \wedge z_1 \neq z_2$.

Now, for each i define the function f_i so that, for every tuple $\vec{m}$ of the required arity, $f_i(\vec{m})$ corresponds to the only element of $H_i(s)$ for an arbitrary $s \in X[M/\vec{x}\vec{y}]$ with $s(\vec{w}_i) = \vec{m}$, and let o be any assignment with domain $\vec{x}\vec{y}$.

Thus, if we can prove that $M \models_o ((\texttt{Rel}_{\vec{v}}(X))\vec{x} \leftrightarrow f_1(\vec{x}) = f_2(\vec{x})) \wedge \psi(\vec{x}, \vec{y}, \vec{f})$ then the left-to-right direction of our proof is done.

First of all, suppose that $M \models_o (\texttt{Rel}_{\vec{v}}(X))\vec{x}$, that is, that $o(\vec{x}) = \vec{m} = s(\vec{v})$

for some $s \in X$.

Then choose an arbitrary tuple of elements $\vec{r}$ and consider the assignment $h = s[\vec{m}/\vec{x}][\vec{r}/\vec{y}][\vec{H}/\vec{z}] \in Y$. This h cannot belong to Y_2, since $h(\vec{v}) = s(\vec{v}) = \vec{m} = h(\vec{x})$, and therefore it is in Y_1 and $h(z_1) = h(z_2)$.

By the definition of the f_i, this implies that $f_1(\vec{m}) = f_2(\vec{m})$, as required.

Analogously, suppose that $M, \not\models_o (\texttt{Rel}_{\vec{v}}(X))\vec{x}$, that is, that $o(\vec{x}) = \vec{m} \neq s(\vec{v})$ for all $s \in X$. Then pick an arbitrary such $s \in X$ and an arbitrary tuple of elements $\vec{r}$, and consider the assignment

$$h = s[\vec{m}/\vec{x}][\vec{r}/\vec{y}][\vec{H}/\vec{z}] \in Y.$$

If h were in Y_1, there would exist an assignment $h' \in Y_1$ such that $h'(\vec{v}) = h(\vec{x}) = \vec{m}$; but this is impossible, and therefore $h \in Y_2$. Hence $h(z_1) \neq h(z_2)$, and therefore $f_1(\vec{m}) \neq f_2(\vec{m})$.

Putting everything together, we just proved that

$$M \models_o R\vec{x} \Leftrightarrow f_1(\vec{x}) = f_2(\vec{x})$$

for all assignments o with domain $\vec{x}\vec{y}$, and we still need to verify that $M \models_o \psi(\vec{x}, \vec{y}, f)$ for all such o.

But this is immediate: indeed, let s be an arbitrary assignment of X, and construct the assignment

$$h = s[o(\vec{x}\vec{y})/\vec{x}\vec{y}][\vec{H}/\vec{z}] \in X[M/\vec{x}\vec{y}][\vec{H}/\vec{z}].$$

Then, since $M \models_{X[M/\vec{x}\vec{y}][\vec{H}/\vec{z}]} \psi'(\vec{x}, \vec{y}, \vec{z})$ and $\psi'(\vec{x}, \vec{y}, \vec{z})$ is first order, $M \models_{\{h\}} \psi'(\vec{x}, \vec{y}, \vec{z})$; but $\psi'(\vec{x}, \vec{y}, \vec{f}(\vec{x}\vec{y}))$ is equivalent to $\psi(\vec{x}, \vec{y}, \vec{f})$ and $h(z_i) = f(h(\vec{w}_i)) = f(o(\vec{w}_i))$, and therefore

$$M \models_o \psi(\vec{x}, \vec{y}, \vec{f})$$

as required.

Conversely, suppose that $M \models_s (\texttt{Rel}_{\vec{v}}(X))\vec{x} \leftrightarrow (f_1(\vec{x}) = f_2(\vec{x})) \wedge \psi(\vec{x}, \vec{y}, \vec{f})$ for all assignments s with domain $\vec{x}\vec{y}$ and for some fixed choice of the tuple of functions $\vec{f}$.

Then let $\vec{H}$ be such that, for all assignments h and for all i,

$$H_i(h) = \{f_i(h(\vec{w}_i))\}$$

and consider $Y = X[M/\vec{x}\vec{y}][H/\vec{z}]$.

Clearly, Y satisfies the dependency conditions; furthermore, it satisfies $\psi'(\vec{x}, \vec{y}, \vec{z})$, because for every assignment $h \in Y$ and every $i \in 1 \ldots n$ we have that $H_i(h) = \{h(z_i)\} = \{f_i(h(\vec{w}_i))\}$.

Finally, we can split Y into two subteams Y_1 and Y_2 as follows:

$$Y_1 = \{o \in Y : o(\vec{z}_1) = o(\vec{z}_2)\};$$
$$Y_2 = \{o \in Y : o(\vec{z}_1) \neq o(\vec{z}_2)\}.$$

It is then trivially true that $M \models_{Y_1} z_1 = z_2$ and $M \models_{Y_2} z_1 \neq z_2$, and all that is left to do is proving that $M \models_{Y_1} \vec{v} \subseteq \vec{x}$ and $M \models_{Y_2} \vec{v} \mid \vec{x}$.

As for the former, let $o \in Y_1$: then, since $o(z_1) = o(z_2)$, $f_1(o(\vec{x})) = f_2(o(\vec{x}))$. This implies that $o(\vec{x}) \in \mathtt{Rel}_{\vec{v}}(X)$, and hence that there exists an assignment $s' \in X$ with $s'(\vec{v}) = o(\vec{x})$.

Now consider the assignment

$$o' = s'[o(\vec{x}\vec{y})/\vec{x}\vec{y}][\vec{H}/\vec{z}] :$$

since in Y the values of $\vec{z}$ depend only on the values of $\vec{x}\vec{y}$ and since $o(z_1) = o(z_2)$, we have that $o'(z_1) = o'(z_2)$ and hence $o' \in Y_1$ too. But $o'(\vec{v}) = s'(\vec{v}) = o(\vec{x})$, and since o was an arbitrary assignment of Y_1, this implies that $M \models_{Y_1} \vec{v} \subseteq \vec{x}$.

Finally, suppose that $o \in Y_2$. Then, since $o(z_1) \neq o(z_2)$, we have that $f_1(o(\vec{x})) \neq f_2(o(\vec{x}))$; and therefore, $o(\vec{x}) \notin \mathtt{Rel}_{\vec{v}}(X)$, that is, for all assignments $s \in X$ it holds that $s(\vec{v}) \neq o(\vec{x})$. Then the same holds for all $o' \in Y_2$.

This concludes the proof. □

Since by Corollary 4.3.22 we already know Independence Logic and I/E Logic have the same expressive power, this has the following corollary:

Corollary 4.5.3. *Let $\Phi(A)$ be an existential second order formula with $\mathtt{Free}(\Phi) = A$, and let $\vec{v}$ be any set of variables such that $|\vec{v}| = Arity(A)$. Then there exists an Independence Logic formula $\phi(\vec{v})$ such that*

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

for all suitable models M and teams X.

Finally, by Fagin's Theorem ([20]) this gives an answer to Grädel and Väänänen's question:

Corollary 4.5.4. *All NP properties of teams are expressible in Independence Logic.*

This result has far-reaching consequences. First of all, it implies that Independence Logic (or, equivalently, I/E Logic) is the most expressive logic of imperfect information which only deals with existential second order properties. Extensions of Independence Logic can of course be defined; but unless they are capable of expressing some property which is not existential second order (as, for example, is the case for the Intuitionistic Dependence Logic of [74], or for the BID Logic of [3]), they will be expressively equivalent to Independence Logic proper. As (Jouko Väänänen, private communication) pointed out, this means that Independence Logic is *maximal* among the logics of imperfect information which always generate existential second order properties of teams. In particular, *any* dependency condition which is expressible as an existential second order property over teams can be expressed in Independence Logic: and as we will see in the next section, this entails that such a logic is capable of expressing a great amount of the notions of dependency considered by database theorists.

4.6 Announcements, Constancy Atoms, and Inconstancy Atoms

In the previous sections, we examined the relationship between Independence Logic and a number of other logics of imperfect information; and through this analysis, we succeeded in characterizing the expressive power of Independence Logic.

However, all of these logics add relatively complicated notions of dependence to the language of Dependence Logic. As we saw in Chapter 3, Dependence Logic $\mathcal{D}$ is equivalent to $\mathcal{FO}(\delta^1, =\!(\cdot))$, that is, to First Order Logic (with Team Semantics) augmented with announcement operators and constancy atoms: indeed, a dependence atom $=\!(x_1 \ldots x_n)$ can easily be decomposed as $\delta^1 x_1 \ldots \delta^1 x_{n-1} \ =\!(x_n)$, and on the other hand, as either of Theorem 2.2.14 or Proposition 3.1.3 demonstrate, announcement operators do not increase the expressive power of Dependence Logic.

In this last section, we will attempt to adapt this reduction to the cases of Inclusion Logic and Independence Logic. As we will see, this will be remarkably easy: using the results of the previous sections, it will be unproblematic to show that, in order to obtain Independence Logic, it suffices to add to the language of $\mathcal{FO}(\delta^1, =\!(\cdot))$ the following *inconstancy atoms*:

TS-inconst: For all terms t, $M \models_X \neq\!(t)$ if and only if for any $s \in X$ there exists an $s' \in X$ with $t\langle s\rangle \neq t\langle s'\rangle$.

In other words, a nonempty team X satisfies $\neq\!(t)$ if and only if $X = \emptyset$ or the value of t is not constant in X. Hence, an inconstancy atom $=\!(t)$ is equivalent to the Team Logic expression $0 \vee \sim\!=\!(t)$, where 0 represents the false formula (which holds only in the empty assignment).

The satisfaction conditions for inconstancy atoms are easily expressible in First Order Logic: and therefore, it follows at once from Theorem 4.5.2 and from the fact that inconstancy atoms satisfy the locality principle that $\mathcal{FO}(\delta^1, =\!(\cdot), \neq\!(\cdot))$ is contained in Independence Logic, in the sense that any formula of this logic is equivalent to some Independence Logic formula.

Does the opposite hold? Well, we already saw that dependence atoms are expressible in this logic; and therefore, by Theorem 4.3.17, we know that exclusion atoms are also expressible in it. If we could prove that inclusion atoms are expressible in $\mathcal{FO}(\delta^1, =\!(\cdot), \neq\!(\cdot))$, we could apply Theorem 4.3.21 and conclude at once that this logic is equivalent to Independence Logic.

First of all, let us define a couple of simple abbreviations:

Definition 4.6.1. Let $x_1 \dots x_n$ be variables, and let t be a term. Then we will write $\neq(x_1 \dots x_n, t)$ for $\delta^1 x_1 \dots \delta^1 x_n \neq(t)$.

Furthermore, let $t_1 \dots t_n, t'$ be terms, and let $v_1 \dots v_n$ be variables not occurring in them. Then we will write $\neq(t_1 \dots t_n, t')$ for

$$\exists v_1 \dots v_n \left(\bigwedge_{i=1}^{n} (v_i = t_i) \wedge \neq(v_1 \dots v_n, t) \right).$$

Proposition 4.6.2. *For all models M, teams X, tuples of terms $\vec{t}$ and terms t', $M \models_X \neq(\vec{t}, t')$ if and only if for any $s \in X$ there exists a $s' \in X$ which coincides with s over $\vec{t}$, but not over t'.*

Proof. Trivial. □

It is worth observing that $\neq(\vec{t}, t')$ is not equivalent to the contradictory negation $\sim=(\vec{t}, t')$ of $=(\vec{t}, t')$. Indeed, a team X satisfies the latter only if there exist two assignments $s, s' \in X$ which coincide on $\vec{t}$ but not on t', and this is clearly different from the condition of Proposition 4.6.2. This semantic condition was mentioned in an informal discussion between the author and Fausto Barbero on the different possible ways of "negating" a dependence atom; and the author thanks Barbero for drawing his attention to this interesting notion of non-dependence.

Now, it is easy enough to see that "non-dependencies" $\neq(t_1 \dots t_n, t')$ are expressible in Inclusion Logic:

Proposition 4.6.3. *Let $\vec{t}$ be a tuple of terms, let t' be a term, and let v be a new variable. Then $\neq(\vec{t}, t')$ is equivalent to $\exists v(v \neq t' \wedge \vec{t}v \subseteq \vec{t}t')$.*

Proof. Obvious. □

What about the converse?

We can rewrite the equiextension atoms of Subsection 4.3.3 in terms of nondependence atoms:

Proposition 4.6.4. *Let $\vec{t_1}$ and $\vec{t_2}$ be tuples of terms of the same length, let $\vec{u}$ be a tuple of new variables of this length, and let v_1, v_2, v_3 be three additional new variables. Then $\vec{t_1} \bowtie \vec{t_2}$ is equivalent to*

$$\forall v_1 v_2 v_3 ((v_1 = v_2) \vee (v_1 \neq v_2 \wedge v_1 \neq v_3 \wedge v_2 \neq v_3) \vee (((v_3 = v_1 \wedge v_3 \neq v_2) \vee (v_3 \neq v_1 \wedge v_3 = v_2)) \wedge \exists \vec{u}((v_3 \neq v_1 \vee \vec{u} = \vec{t_1}) \wedge (v_3 \neq v_2 \vee \vec{u} = \vec{t_2}) \wedge \neq(\vec{u} v_1 v_2 v_3)))).$$

Proof. Suppose that $M \models_X \vec{t}_1 \bowtie \vec{t}_2$, that is, that $X(\vec{t}_1) = X(\vec{t}_2)$, let $Y = X[M/v_1v_2v_3]$, and let

- $Y_1 = \{s \in Y : s(v_1) = s(v_2)\}$;
- $Y_2 = \{s \in Y : s(v_1), s(v_2), \text{ and } s(v_3) \text{ are all different}\}$;
- $Y_3 = Y \backslash (Y_1 \cup Y_2)$.

Clearly, $M \models_{Y_1} v_1 = v_2$, $M \models_{Y_2} v_1 \neq v_2 \wedge v_1 \neq v_3 \wedge v_2 \neq v_3$ and $M \models_{Y_3} (v_3 = v_1 \wedge v_3 \neq v_2) \vee (v_3 \neq v_1 \wedge v_3 = v_2)$.

Furthermore, let $\vec{H}$ be such that

$$\vec{H}(s) = \begin{cases} \{\vec{t}_1\langle s\rangle\} & \text{if } s(v_3) = s(v_1); \\ \{\vec{t}_2\langle s\rangle\} & \text{if } s(v_3) = s(v_2) \end{cases}$$

and consider $Z = Y_3[\vec{H}/\vec{u}]$. By construction, we have that $M \models_Z (v_3 \neq v_1 \vee \vec{u} = \vec{t}_1) \wedge (v_3 \neq v_2 \vee \vec{u} = \vec{t}_2)$. Furthermore, let $h \in Z$. There are two possibilities:

1. If $h(v_3) = h(v_1)$, then $h(\vec{u}) = \vec{t}_1\langle h\rangle = \vec{t}_1\langle s\rangle$ for some $s \in X$. Since $X(\vec{t}_1) = X(\vec{t}_2)$, there exists a $s' \in X$ with $\vec{t}_1\langle s\rangle = \vec{t}_2\langle s'\rangle$ Now consider $h' = s'[h(v_1)/v_1][h(v_2)/v_2][h(v_2)/v_3][\vec{H}/\vec{u}] \in Z$:[16] by the definition of $\vec{H}$, $h'(\vec{u}) = \vec{t}_2\langle h'\rangle = \vec{t}_1\langle h\rangle = h(\vec{u})$, and furthermore h and h' coincide over v_1 and v_2, but they do not coincide over v_3.

2. Similarly, if $h(v_3) = h(v_2)$ then $h(\vec{u}) = \vec{t}_2\langle h\rangle = \vec{t}_2\langle s\rangle$ for some $s \in X$. Since $X(\vec{t}_1) = X(\vec{t}_2)$, there exists a $s' \in X$ with $\vec{t}_1\langle s\rangle = \vec{t}_2\langle s'\rangle$ Now consider $h' = s'[h(v_1)/v_1][h(v_2)/v_2][h(v_1)/v_3][\vec{H}/\vec{u}] \in Z$: by the definition of $\vec{H}$, $h'(\vec{u}) = \vec{t}_1\langle h'\rangle = \vec{t}_2\langle h\rangle = h(\vec{u})$, and furthermore h and h' coincide over v_1 and v_2, but they do not coincide over v_3.

Therefore, $M \models_Z \neq(\vec{u}v_1v_2v_3)$, as required.

Conversely, suppose that a team X satisfies our expression. Then $Y = X[M/v_1v_2v_3]$ can be split into three teams Y_1, Y_2 and Y_3 satisfying $v_1 = v_2$, $v_1 \neq v_2 \wedge v_1 \neq v_3 \wedge v_2 \neq v_3$ and $(v_3 = v_1 \wedge v_3 \neq v_2) \vee (v_3 = v_2 \wedge v_3 \neq v_1)$ respectively, and it is easy to see that the only way to do that is to use the definitions of Y_1, Y_2 and Y_3 which we gave above. Furthermore, there exists a $\vec{H}$ such that, for $Z = Y_3[\vec{H}/\vec{u}]$, $M \models_Z (v_3 \neq v_1 \vee \vec{u} = \vec{t}_1) \wedge (v_3 \neq v_2 \vee \vec{u} = \vec{t}_2) \wedge \neq (\vec{u}v_1v_2v_3)$, and this implies that $\vec{H}$ is also necessarily as we stated before. Now pick any $s \in X$, and let $a, b \in \texttt{Dom}(M)$ be such that $a \neq b$.

[16] Strictly speaking, this expression defines a set of assignments of size one. The assignment h' is then chosen as its unique element; and it is in Z because, by definition, $s'[h(v_1)/v_1][h(v_2)/v_2][h(v_2)/v_3]$ is in Y_3.

1. Consider $h = s[a/v_1][b/v_2][a/v_3][\vec{t}_1\langle s\rangle/\vec{u}] \in Z$. Since $M \models_Z \neq(\vec{u}v_1v_2v_3)$, there exists a $h' \in Z$ which coincides with h over $\vec{u}v_1v_2$ but not over v_3. Since $h' \in Z$, this implies that $h'(v_3) = h'(v_2)$ and that $h'(\vec{u}) = \vec{t}_2\langle h'\rangle = \vec{t}_2\langle s'\rangle$ for some $s' \in X$. Hence, there exists a $s' \in X$ with $\vec{t}_2\langle s'\rangle = h'(\vec{u}) = h(\vec{u}) = \vec{t}_1\langle s\rangle$.

2. Consider $h = s[a/v_1][b/v_2][b/v_3][\vec{t}_2\langle s\rangle/\vec{u}] \in Z$. By a similar argument, we have that there exists a $h' \in Z$ such that $h'(\vec{u}) = h(\vec{u}) = \vec{t}_2\langle s\rangle$ and $h'(\vec{u}) = \vec{t}_1\langle s'\rangle$ for some $s' \in X$.

Hence, $M \models_X \vec{t}_1 \bowtie \vec{t}_2$, and this concludes the proof. □

From these results, Corollary 4.3.16 and Theorems 4.3.17, 4.3.21 it follows at once that

Theorem 4.6.5. *$\mathcal{FO}(\delta^1, \neq(\cdot))$ is logically equivalent to Inclusion Logic and Equiextension Logic, even with respect to open formulas.*

Theorem 4.6.6. *$\mathcal{FO}(\delta^1, =(\cdot), \neq(\cdot))$ is logically equivalent to Independence Logic and Inclusion/Exclusion Logic, even with respect to open formulas.*

As a consequence of these results dependence and independence atoms, as well as inclusion and exclusion atoms, are unnecessary as primitives of our language if we already have constancy atoms, inconstancy atoms, and announcement operators. This is surprising, since constancy/inconstancy atoms and announcement operators are extremely simple; and in a way, the decomposition of dependence and independence atoms into such atoms and operators can be seen as analogous to the known decomposition of dependence atoms into constancy atoms and intuitionistic implication of [3].

However, we certainly did not exhaust the argument of reductions between non-functional dependencies here. First of all, the problem of the expressive power of Inclusion Logic is still, to the knowledge of the author, open; and moreover, it is not difficult to define additional, and yet unclassified, fragments or variants of these logics.[17] The contents of this chapter can be thought of as a first attempt to provide a (partial) description of the lattice of reductions between logics of imperfect information; and we conclude it by expressing the hope that this description will be further expanded.

[17] One of the most interesting such ones is, in the opinion of the author, the variant of Independence Logic which only admits "pure" independence atoms $\vec{t}_1 \perp \vec{t}_2$. Another one might be Constancy/Inconstancy Logic without the announcement operators.

Chapter 5

Proof Theory

The validity problem for Dependence Logic (as well as for many of its variants examined in the previous chapters) is not decidable, as it follows at once by its equivalence to Σ_1^1 over sentences. One can develop axiomatic systems for *fragments* of these logics, as Kontinen and Väänänen did in [52] for the first order consequences of Dependence Logic formulas;[1] but it is not possible to generalize these results to full Dependence Logic under its usual semantics while preserving semidecidability.

However, Henkin developed in [35] a *General Semantics* for Second Order Logic, in which second order quantifiers range over an *universe of discourse* which is not necessarily the whole powerset; and furthermore, in the same paper, he developed a sound and complete axiom system for this logic.

In this chapter, we will first build a similar *General Team Semantics* in which not all teams belong to the universe of discourse; and afterwards, we will develop a proof system for Independence/Inclusion/Exclusion Logic $\mathcal{I}(\subseteq, |)$ which is sound and complete with respect to it. As we will see, the fact that our this formalism is contained in *Existential* Second Order Logic will be a big advantage for us, as it will allow us to focus exclusively on the *least* general models of our class.

5.1 General Models

Definition 5.1.1. Let Σ be a first order signature. A *general model* with signature Σ is a pair $(M, \mathcal{G})$, where M is a first order model with signature Σ

[1] Another proof system for a fragment of Dependence Logic is the one developed by Ville Nurmi in [57]. However, it is not known if Nurmi's system is complete for the corresponding fragment.

and $\mathcal{G}$ is a set of teams over finite – but not necessarily identical, nor of the same size – domains, respecting the condition

- If $n \in \mathbb{N}$ and $\phi(x_1 \ldots x_n, \vec{m}, \vec{R})$ is a first order formula, where $\vec{m}$ is a tuple of constant parameters in $\texttt{Dom}(M)$ and where $\vec{R}$ is a tuple of "relation parameters" corresponding to teams in $\mathcal{G}$, in the sense that each R_i is of the form
$$R_i = \texttt{Rel}(X_i) = \{s(\vec{z}) : s \in X_i\}$$
for some $X_i \in \mathcal{G}$, then for
$$\|\phi(x_1 \ldots x_n, \vec{m}, \vec{R})\|_M = \{s : \texttt{Dom}(s) = \{x_1 \ldots x_n\}, M \models_s \phi(x_1 \ldots x_n, \vec{m}, \vec{R})\}$$
it holds that $\|\phi(x_1 \ldots x_n, \vec{m}, \vec{R})\|_M \in \mathcal{G}$.

Lemma 5.1.2. *Let Σ be a first order signature and let $(M, \mathcal{G})$ be a general model with signature Σ. Then for all $X \in \mathcal{G}$ and all variables y, $X[M/y] \in \mathcal{G}$.*

Proof. Let $\texttt{Dom}(X) = \vec{x}$, let $R = \texttt{Rel}(X)$, and consider the formula $\phi(\vec{x}, y) = \exists y R(\vec{x})$. Then take any assignment s with domain $\vec{x}y$: by construction, $M \models_s \phi(\vec{x}, y) \Leftrightarrow \exists m$ s.t. $s[m/y]_{|\vec{x}} \in X \Leftrightarrow s \in X[M/y]$, as required.[2] □

We can easily adapt the standard Team Semantics to general models. We will report all the rules here, for ease of reference; but the only differences between this semantics and the previous one are in the cases **GMS-**$\vee$ and **GMS-**$\exists$.

In the case of the rule of the existential quantifier, a formulation somewhat different from the usual one will prove to be more conventient here:

Definition 5.1.3. Let X and X' be two teams on the same domain, and let $x \in \texttt{Var}$ be a variable. Then we write $X[x]X'$ if and only if

1. $\texttt{Dom}(X') = \texttt{Dom}(X) \cup \{x\}$;
2. $\texttt{Rel}_{\texttt{Dom}(X)}(X') = \texttt{Rel}(X)$.

Definition 5.1.4. Let $(M, \mathcal{G})$ be a general model and let X be a team over it. Then

GMS-lit: For all first order literals α, $(M, \mathcal{G}) \models_X \alpha$ if and only if $s \in X$, $M \models_s \alpha$ in the usual first order sense;

[2]Here by $s[m/y]_{|\vec{x}}$ we intend the restriction of $s[m/y]$ to the domain $\{x_1 \ldots x_n\}$. If y is among $x_1 \ldots x_n$, then this is the same of $s[m/y]$ itself; otherwise, it is simply s.

GMS-inc, GMS-exc, GMS-ind: For all inclusion atoms, exclusion atoms or independence atoms β, $(M, \mathcal{G}) \models_X \beta$ if and only if $M \models_X \beta$ in the usual Team Semantics sense;

GMS-$\vee$: For all ψ_1 and ψ_2, $(M, \mathcal{G}) \models_X \psi_1 \vee \psi_2$ if and only if $X = Y \cup Z$ for some two teams $Y, Z \in \mathcal{G}$ such that $(M, \mathcal{G}) \models_Y \psi_1$ and $(M, \mathcal{G}) \models_Z \psi_2$;

GMS-$\wedge$: For all ψ_1 and ψ_2, $(M, \mathcal{G}) \models_X \psi_1 \wedge \psi_2$ if and only if $(M, \mathcal{G}) \models_X \psi_1$ and $(M, \mathcal{G}) \models_X \psi_2$;

GMS-$\exists$: For all ψ and all $x \in \texttt{Var}$, $M \models_X \exists x \psi$ if and only if there exists a $X' \in \mathcal{G}$ such that $X[x]X'$ and $(M, \mathcal{G}) \models_{X'} \psi$;

GMS-$\forall$: For all ψ and all $x \in \texttt{Var}$, $M \models_X \forall x \psi$ if and only if $(M, \mathcal{G}) \models_{X[M/x]} \psi$.

Let us verify that the the same holds for General Model Semantics:

Lemma 5.1.5. *Let $(M, \mathcal{G})$ be a general model, and let $X \in \mathcal{G}$ be such that $\mathtt{Dom}(X) = \vec{x}\vec{y}$. Then $X_{|\vec{x}} = \{s : \mathtt{Dom}(s) = \vec{x}, \exists \vec{m} \text{ s.t. } s[\vec{m}/\vec{y}] \in X\}$ is in $\mathcal{G}$.*

Furthermore, let $Y \subseteq X_{|\mathcal{G}}$ be such that $Y \in \mathcal{G}$. Then the team

$$X(\vec{x} \in Y) = \{s \in X : s_{|\vec{x}} \in Y\}$$

is in $\mathcal{G}$.

Proof. By definition, $X_{|\vec{x}}$ is $\|\phi(\vec{x}, R)\|_M$, where ϕ is $\exists \vec{y}(R\vec{x}\vec{y})$ and $R = \texttt{Rel}(X)$. Therefore, $X_{|\vec{x}} \in \mathcal{G}$.

Similarly, $X(\vec{x} \in Y)$ is $\|\phi(\vec{x}\vec{y}, R_1, R_2)\|_M$, where ϕ is $R_1\vec{x}\vec{y} \wedge R_2\vec{x}$, R_1 is $\texttt{Rel}(X)$ and R_2 is $\texttt{Rel}(Y)$. ☐

Theorem 5.1.6 (Locality)**.** *Let $(M, \mathcal{G})$ be a general model, let $X \in \mathcal{G}$ and let ϕ be a formula over the signature of M with $\mathtt{Free}(\phi) = \vec{z} \subseteq \mathtt{Dom}(X)$. Then $(M, \mathcal{G}) \models_X \phi$ if and only if $(M, \mathcal{G}) \models_{X_{|\vec{z}}} \phi$.*

Proof. The proof is by structural induction on ϕ. We present only the passages corresponding to disjunction and existential quantification, as the others are trivial:

- Suppose that $(M, \mathcal{G}) \models_X \psi_1 \vee \psi_2$. Then, by definition, there exist teams Y and Z in $\mathcal{G}$ such that $X = Y \cup Z$, $(M, \mathcal{G}) \models_Y \psi_1$ and $M \models_Z \psi_2$. By induction hypothesis, this means that $(M, \mathcal{G}) \models_{Y_{|\vec{z}}} \psi_1$ and $(M, \mathcal{G}) \models_{Z_{|\vec{z}}} \psi_2$. But $Y_{|\vec{z}} \cup Z_{|\vec{z}} = X_{|\vec{z}}$, and hence $(M, \mathcal{G}) \models_{X_{|\vec{z}}} \psi_1 \vee \psi_2$.

Conversely, suppose that $(M, \mathcal{G}) \models_{X_{|\vec{z}}} \psi_1 \vee \psi_2$. Then there exist teams Y', Z' in $\mathcal{G}$ such that $(M, \mathcal{G}) \models_{Y'} \psi_1$, $(M, \mathcal{G}) \models_{Z'} \psi_2$ and $X_{|\vec{z}} = X' \cup Y'$. Now let Y be $X(\vec{z} \in Y')$ and Z be $X(\vec{z} \in Z')$; by construction, $Y \cup Z = X$, and furthermore $Y' = Y_{|\vec{z}}$ and $Z' = Z_{|\vec{z}}$, and, by the lemma, Y and Z are in $\mathcal{G}$. Thus, by induction hypothesis, $(M, \mathcal{G}) \models_Y \psi_1$ and $(M, \mathcal{G}) \models_Z \psi_2$, and finally $(M, \mathcal{G}) \models_X \psi_1 \vee \psi_2$, as required.

- Suppose that $(M, \mathcal{G}) \models_X \exists x \psi$. Then there exists a team $Y \in \mathcal{G}$ such that $X[x]Y$ and $(M, \mathcal{G}) \models_Y \psi$. By induction hypothesis, this means that $(M, \mathcal{G}) \models_{Y_{|\vec{z}x}} \psi$ too; and since $X_{|\vec{z}}[x]Y_{|\vec{z}x}$, this implies that $M \models_{X_{|\vec{z}}} \exists x \psi$, as required.

  Conversely, suppose that $(M, \mathcal{G}) \models_{X_{|\vec{z}}} \exists x \psi$. Then there exists a team Y', with domain $\vec{z}x$, such that $M \models_{Y'} \psi$ and $X_{|\vec{z}}[x]Y'$. Now let Y be $(X[M/x])(\vec{z}x \in Y')$. By the lemma, $Y \in \mathcal{G}$; furthermore, $Y_{|\vec{z}x} = Y'$, and hence by induction hypothesis $(M, \mathcal{G}) \models_Y \psi$. Finally, $X[x]Y$: indeed, if $s \in X$ then $s_{\vec{z}}[m/x] \in Y'$ for some $m \in \mathtt{Dom}(M)$, and hence $s[m/x] \in Y$ for the same m, and on the other hand, Y is contained in $X[M/x]$, and hence if $s[m/x] \in Y$ it follows that $s \in X$.

  Therefore $(M, \mathcal{G}) \models_X \exists x \psi$, as required.

□

As in the case of Second Order Logic, first-order models can be represented as a special kind of general model:

Definition 5.1.7. Let $(M, \mathcal{G})$ be a general model. Then it is said to be *full* if and only if $\mathcal{G}$ contains all teams over M.

The following result is then trivial.

Proposition 5.1.8. *Let $(M, \mathcal{G})$ be a full model. Then for all suitable teams X and formulas ϕ, $(M, \mathcal{G}) \models_X \phi$ in General Team Semantics if and only if $M \models_X \phi$ in the usual Team Semantics.*

Proof. Follows at once by comparing the rules of Team Semantics and General Team Semantics for the case that $\mathcal{G}$ contains all teams. □

How does the satisfaction relation in General Team Semantics change if we vary the set $\mathcal{G}$? The following definition and result give us some information about this:

Definition 5.1.9. Let $(M, \mathcal{G})$ and $(M, \mathcal{G}')$ be two general models. Then we say that $(M, \mathcal{G}')$ is a *refinement* of $(M, \mathcal{G})$, and we write $(M, \mathcal{G}) \subseteq (M, \mathcal{G}')$, if and only if $\mathcal{G} \subseteq \mathcal{G}'$.

Intuitively speaking, a refinement of a general model is another general model with more teams in it than the former. The following result shows that refinements preserve satisfaction relations:

Theorem 5.1.10. *Let $(M, \mathcal{G})$ and $(M, \mathcal{G}')$ be two general models with $(M, \mathcal{G}) \subseteq (M, \mathcal{G}')$, let $X \in \mathcal{G}$, and let ϕ be a formula over the signature of M with $\texttt{Free}(\phi) \subseteq \texttt{Dom}(X)$. Then*

$$(M, \mathcal{G}) \models_X \phi \Rightarrow (M, \mathcal{G}') \models_X \phi.$$

Proof. The proof is an easy induction on ϕ.

1. If ϕ is a first order literal or a non-first-order atom, the result is obvious, as the choice of the set of teams $\mathcal{G}$ (or $\mathcal{G}'$) does not enter into the definition of its satisfaction condition.

2. If $(M, \mathcal{G}) \models_X \psi_1 \vee \psi_2$ then there exist two teams $Y, Z \in \mathcal{G}$ such that $X = Y \cup Z$, $(M, \mathcal{G}) \models_Y \psi_1$ and $(M, \mathcal{G}) \models_Z \psi_2$. But Y and Z are also in $\mathcal{G}'$, and by induction hypothesis we have that $(M, \mathcal{G}') \models_Y \psi_1$ and $(M, \mathcal{G}') \models_Z \psi_2$, and therefore $(M, \mathcal{G}') \models_X \psi_1 \vee \psi_2$.

3. If $(M, \mathcal{G}) \models_X \psi_1 \wedge \psi_2$ then $(M, \mathcal{G}) \models_X \psi_1$ and $(M, \mathcal{G}) \models_X \psi_2$. Then, by induction hypothesis, $(M, \mathcal{G}') \models_X \psi_1$ and $(M, \mathcal{G}') \models_X \psi_2$, and finally $(M, \mathcal{G}') \models_X \psi_1 \wedge \psi_2$.

4. If $(M, \mathcal{G}) \models_X \exists x \psi$ then there exists a $X' \in \mathcal{G}$ such that $X[x]X'$ and $(M, \mathcal{G}) \models_{X'} \psi$. But then X' is also in $\mathcal{G}'$, and by induction hypothesis $(M, \mathcal{G}') \models_{X'} \psi$, and finally $(M, \mathcal{G}') \models_X \exists x \psi$.

5. If $(M, \mathcal{G}) \models_X \forall x \psi$ then $(M, \mathcal{G}) \models_{X[M/x]} \psi$. Then, by induction hypothesis, $(M, \mathcal{G}') \models_{X[M/x]} \psi$, and finally $(M, \mathcal{G}') \models_X \forall x \psi$.

□

This result shows us that, as was to be expected from the equivalence between independence logic and existential second order logic, if we are interested in formulas which hold in *all* general models over a certain first-order model we only need to pay attention to the *smallest* (in the sense of the refinement relation) ones. But do such "least general models" exist? As the following result shows, this is indeed the case:

Proposition 5.1.11. *Let $\{(M, \mathcal{G}_i) : i \in I\}$ be a family of general models with signature Σ and over the same first order model M. Then $(M, \bigcap_{i \in I} \mathcal{G}_i)$ is also a general model.*

Proof. Let $\phi(x_1 \ldots x_n, \vec{m}, \vec{R})$ be a first order formula with parameters, where each R_i is of the form $\texttt{Rel}(X)$ for some $X \in \cap_i \mathcal{G}_i$. Then $\|\phi(x_1 \ldots x_n, \vec{m}, \vec{R})\|_M$ is in $\mathcal{G}_i$ for all $i \in I$, and therefore it is in $\bigcap_{i \in I} \mathcal{G}$, as required. □

Therefore, it is indeed possible to talk about the *least general model* over a first order model.

Definition 5.1.12. Let M be a first order model. Then the *least general model* over M is the $(M, \mathcal{L})$, where

$$\mathcal{L} = \bigcap \{\mathcal{G} : (M, \mathcal{G}) \text{ is a general model.}\}$$

As an example of a least general model, let $n \in \mathbb{N}$, and let M_n be a model with empty signature and domain $\{1 \ldots n\}$. Then the least general model over M_n is actually the full general model $(M_n, \mathcal{G}_n)$, where $\mathcal{G}_n$ contains all teams over M_n. Indeed, let $\{v_1 \ldots v_k\}$ be a finite set of variables and let

$$X = \{s_1 \ldots s_q\} = \begin{array}{c|ccc} & v_1 & \ldots & v_k \\ \hline s_1 & a_{11} & \ldots & a_{1k} \\ \ldots & \ldots & \ldots & \ldots \\ s_q & a_{q1} & \ldots & a_{qk} \end{array}$$

be any team over M_n with domain $\{v_1 \ldots v_k\}$, where $s_i(v_j) = a_{ij}$ for all $i \in 1 \ldots q$ and all $j \in 1 \ldots k$. Then clearly $q \leq n^k$, and furthermore, for $\phi(v_1 \ldots v_k) = \bigvee_{i=1}^{q} \bigwedge_{j=1}^{k} v_i = a_{qi}$ we have that

$$\|\phi(v_1 \ldots v_k, a_{11} \ldots a_{qk})\|_M = \{s : \texttt{Dom}(s) = \{v_1 \ldots v_k\}, M \models_s \phi\| = X$$

as required.

As this example shows, if M is finite then the least (and only) general model over it is the full one. Hence, if we are only interested in finite models, General Model Semantics is equivalent to the standard Team Semantics, and the same can be said about the Entailment Semantics which we will develop later in this chapter.

What is the purpose of least general models? The answer comes as a consequence of Theorem 5.1.10, and can be summarized by the following corollary:

Corollary 5.1.13. *Let Σ be a first order signature, let M be a first order model over it and let $(M, \mathcal{L})$ be the least general model over it. Then, for all teams*

$X \in \mathcal{L}$ and all formulas ϕ with signature Σ and with free variables in $\texttt{Dom}(X)$,

$$(M, \mathcal{L}) \models_X \phi \Leftrightarrow (M, \mathcal{G}) \models_X \phi \text{ for all general models } (M, \mathcal{G}) \text{ over } M.$$

Proof. Suppose that $(M, \mathcal{L}) \models_X \phi$. Then take any general model $(M, \mathcal{G})$: by definition, we have that $(M, \mathcal{L}) \subseteq (M, \mathcal{G})$, and hence by Theorem 5.1.10 we have that $(M, \mathcal{G}) \models_X \phi$.

Conversely, suppose that $(M, \mathcal{G}) \models_X \phi$ for all general models $(M, \mathcal{G})$; then in particular $(M, \mathcal{L}) \models_X \phi$, as required. □

We can also find a more practical characterization of this "least general model".

Proposition 5.1.14. *Let M be a first order model. Then the least general model over it is $(M, \mathcal{L})$, where $\mathcal{L}$ is the set of all $\|\phi(\vec{x}, \vec{m})\|_M$, where ϕ ranges over all first order formulas and $\vec{m}$ ranges over all tuples of variables of suitable length.*

Proof. If $(M, \mathcal{G})$ is a general model then $\mathcal{L} \subseteq \mathcal{G}$ by definition; therefore, we only need to prove that $(M, \mathcal{L})$ is a general model.

Now, let $\phi(\vec{x}, \vec{m}, \vec{R})$ be a first order formula, and let each R_i be $\texttt{Rel}(X_i)$ for some $X_i \in \mathcal{L}$. So for each R_i, any assignment s and any suitable tuple of terms t, $M \models_s R_i\vec{t}$ if and only if $M \models_s \psi_i(\vec{t}, \vec{n}_i)$ for some first order formula ψ_i with parameters $\vec{n}_i$. Now let $\phi'(\vec{x}, \vec{m}, \vec{n}_1, \vec{n}_2, \ldots)$ be the expression obtained by replacing, in ϕ, each instance of $R_i\vec{t}$ by $\psi_i(\vec{t}, \vec{n}_i)$; by construction, we have that $M \models_s \phi(\vec{x}, \vec{m}, \vec{R})$ if and only if $M \models_s \phi'(\vec{x}, \vec{m}, \vec{n}_1, \ldots)$, and therefore

$$\|\phi(\vec{x}, \vec{m}, \vec{R})\|_M = \|\phi'(\vec{x}, \vec{m}, \vec{n}_1, \vec{n}_2, \ldots)\|_M \in \mathcal{L}$$

as required. □

Definition 5.1.15. Let Σ be a first order signature, let V be a finite set of variables, and let ϕ be a formula of our language with free variables in V. Then ϕ is *valid* with respect to general models if and only if $(M, \mathcal{G}) \models_X \phi$ for all general models $(M, \mathcal{G})$ with signature Σ and for all teams $X \in \mathcal{G}$ with $\texttt{Dom}(X) \supseteq \texttt{Free}(\phi)$. If this is the case, we write GMS $\models \phi$.

Definition 5.1.16. Let Σ be a first order signature, let V be a finite set of variables, and let ϕ be a formula of our language over this signature with free variables in V. Then ϕ is *valid* with respect to least general models if and only if $(M, \mathcal{L}) \models_X \phi$ for all least general models $(M, \mathcal{L})$ with signature Σ and for all teams $X \in \mathcal{L}$ with $\texttt{Dom}(X) \supseteq \texttt{Free}(\phi)$. If this is the case, we write LMS $\models \phi$.

Lemma 5.1.17. *Let M be a first order model with signature Σ, and let M' be another first order model with signature $\Sigma' \supseteq \Sigma$ such that the restriction of M' to Σ is precisely M. Then for all general models $\mathcal{G}$ for M', for all formulas ϕ with signature Σ and for all $X \in \mathcal{G}$,*

$$(M, \mathcal{G}) \models_X \phi \Leftrightarrow (M', \mathcal{G}) \models_X \phi.$$

Proof. First of all, if $(M', \mathcal{G})$ is a general model then $(M, \mathcal{G})$ is also a general model. Then, the result is proved by observing that the truth conditions of our semantics depend only on the interpretations of the symbols in the signature of the formula (and on the choice of $\mathcal{G}$, of course). □

Lemma 5.1.18. *Let $(M, \mathcal{G})$ be a general model with signature Σ, let $S \not\in \Sigma$ be a new relation symbol and let $X \in \mathcal{G}$. Furthermore, let $M' = M[\mathtt{Rel}(X)/S]$ be the extension of M to the signature $\Sigma \cup \{S\}$ such that $S^{M'} = \mathtt{Rel}(X)$. Then $(M', \mathcal{G})$ is a general model.*

Proof. Let $\phi(\vec{x}, \vec{m}, \vec{R})$ be a first order formula with signature $\Sigma \cup \{S\}$ and parameters $\vec{m}$ and $\vec{R}$, where each R_i is $\mathtt{Rel}(X_i)$ for some $X_i \in \mathcal{G}$. Then let $\phi'(\vec{x}, \vec{m}, \vec{R}, S)$ be the first order formula with signature Σ, where S now stands for the relation $\mathtt{Rel}(X)$. Now clearly

$$\|\phi(\vec{x}, \vec{m}, \vec{R})\|_{M'} = \|\phi'(\vec{x}, \vec{m}, \vec{R}, S)\|_M \in \mathcal{G},$$

as required. □

Theorem 5.1.19. *A formula ϕ is valid wrt general models if and only if it is valid wrt least general models.*

Proof. The left to right direction is obvious. For the right to left direction, suppose that LMS $\models \phi$, let $(M, \mathcal{G})$ be a general model whose signature contains the signature of ϕ, and let $X \in \mathcal{G}$ be a team whose domain $\{x_1 \ldots x_n\}$ contains all free variables of ϕ. Then consider the first order model $M' = M[\mathtt{Rel}(X)/S]$, where S is a new relation symbol, and take the least general model $(M', \mathcal{L})$ over it. We clearly have that $X \in \mathcal{L}$, since

$$X = \{s : \mathtt{Dom}(s) = \{x_1 \ldots x_n\}, M' \models_s Sx_1 \ldots x_n\}$$

and, therefore, $(M', \mathcal{L}) \models_X \phi$ by hypothesis. Now, by Lemma 5.1.18, $(M', \mathcal{G})$ is a general model, and therefore by definition $\mathcal{L} \subseteq \mathcal{G}$, and hence by Theorem 5.1.10 $(M', \mathcal{G}) \models_X \phi$ too. Finally, the relation symbol S does not occur in ϕ, and therefore by Lemma 5.1.17 $(M, \mathcal{G}) \models_X \phi$, as required. □

In the next section, we will develop another, more syntactic way of reasoning about least general models.

5.2 Entailment Semantics

Let M be a first order structure and let $(M, \mathcal{L})$ be the least general model over it. Then, as we saw, $\mathcal{L}$ is the set of all teams corresponding to first order formulas with parameters. Therefore, in order to reason about satisfaction in a least general model, there is no need to carry around the teams themselves: rather, we can use the corresponding first order formulas. In this section, we will develop this idea, building up a new "Entailment Semantics" and proving its correspondence with General Model Semantics over least general models.

We will then construct a proof system and prove its soundness and completeness with respect to this semantics. Then, since – as we saw already – validity with respect to least general models is equivalent to validity with respect to general models, this proof system will also seen to be sound and complete with respect to General Model Semantics.

For the purposes of this chapter, Entailment Semantics acts as a bridge between General Model Semantics and our proof system: by allowing us to abstract away from higher-order objects such as teams, it will make it significantly easier for us to establish a connection between semantics and proof theory.

Furthermore, the semantics which we will build, with its more syntactic flavor, is of independent interest. The phenomena of dependence and independence whose study is among the principal reasons of being of dependence logic and independence logic are present in it, but the intrinsically higher-order nature of the usual Team Semantics is not. Entailment Semantics, in other words, can be seen as an attempt of examining the content of the notions of dependence and independence from a first-order perspective, rather than from the higher-order perspective implicit in the formulation of Team Semantics.

Definition 5.2.1. Let $\mathbf{V}_P = \{p_1 \ldots p_n, \ldots\}$ and $\mathbf{V}_T = \{x, y, z, \ldots\}$ be fixed, disjoint, countably infinite sets of variables. We will call any $p \in V_P$ a *parameter variable*, and we will call any $x \in V_T$ a *team variable.* Furthermore, we will assume that any variable which occurs in any of our formulas is a team variable or a parameter variable.

Definition 5.2.2. Let ϕ be any formula. Then $\texttt{Free}_P(\phi) = \texttt{Free}(\phi) \cap \mathbf{V}_P$ and $\texttt{Free}_T(\phi) = \texttt{Free}(\phi) \cap \mathbf{V}_T$.

Parameter variables clarify the interpretation of such expressions such as $M \models_s \gamma(\vec{x}, \vec{m})$: this is simply a shorthand for $M \models_{h \cup s} \gamma(\vec{x}, \vec{p})$, where h is a *parameter assignment* with domain $\vec{p}$ and with $h(\vec{p}) = \vec{m}$. Team variables, instead, are going to be used in order to describe the variables in the domain of the team corresponding to a given first order expression: for any first order $\gamma(\vec{x}, \vec{p})$, where $\vec{x}$ are team variables and $\vec{p}$ are parameter variables, and for any h with domain $\vec{p}$, we will therefore have $\|\gamma(\vec{x}, \vec{p})\|_{M,h} = \|\gamma(\vec{x}, h(\vec{p}))\|_M = \{s : \mathtt{Dom}(s) = \vec{x}, M \models_{h \cup s} \gamma\}$. For this reason, parameter variables will never occur in the domain of a team, and, hence, from this point on we will always assume that parameter variables never occur in independence logic formulas, but only in the first order team definitions.

After these preliminaries, we can now give our main definition for this section:

Definition 5.2.3. For all first order models M, all first order formulas $\gamma(\vec{x}, \vec{p})$ with $\mathtt{Free}_T(\gamma) = \vec{x}$ and $\mathtt{Free}_P(\gamma) = \vec{p}$, and all parameter assignments h with domain $\vec{p}$

ES-lit: For all first order literals α, $M \models_{\gamma(h)} \alpha$ if and only if for all assignments s with domain $\mathtt{Free}_T(\gamma) \cup \mathtt{Free}_T(\alpha)$ such that $M \models_{h \cup s} \gamma$ it holds that $M \models_s \alpha$;

ES-inc, ES-exc, ES-ind: For all inclusion, exclusion or independence atoms β, $M \models_{\gamma(h)} \beta$ if and only if the team $\{s : \mathtt{Dom}(s) = \mathtt{Free}_T(\beta), M \models_{s \cup h} \gamma\}$ satisfies β in the usual sense;

ES-$\vee$: For all ψ_1 and ψ_2, $M \models_{\gamma(h)} \psi_1 \vee \psi_2$ if and only if there exists a parameter assignment h' extending[3] h and there exist first order formulas γ_1 and γ_2 such that

- $\mathtt{Free}_P(\gamma_1), \mathtt{Free}_P(\gamma_2) \subseteq \mathtt{Dom}(h')$;
- $M \models_{\gamma_1(h')} \psi_1$;
- $M \models_{\gamma_2(h')} \psi_2$;
- $M \models_{h'} \forall \vec{v}(\gamma \leftrightarrow \gamma_1 \vee \gamma_2)$, where $\vec{v}$ is $\mathtt{Free}_T(\gamma) \cup \mathtt{Free}_T(\gamma_1) \cup \mathtt{Free}_T(\gamma_2)$;

ES-$\wedge$: For all ψ_1 and ψ_2, $M \models_{\gamma(h)} \psi_1 \wedge \psi_2$ if and only if $M \models_{\gamma(h)} \psi_1$ and $M \models_{\gamma(h)} \psi_2$;

ES-$\exists$: For all $x_n \in \mathtt{Var}_T$ and all ψ, $M \models_{\gamma(h)} \exists x_n \psi$ if and only if there exist a parameter assignment h' extending h and a first order formula γ' with $\mathtt{Free}_P(\gamma') \subseteq \mathtt{Dom}(h')$ such that

- $M \models_{\gamma'(h')} \psi$;

[3] That is, $\mathtt{Dom}(h') \supseteq \mathtt{Dom}(h)$, and $h'(\vec{p}) = h(\vec{p})$.

- $M \models_{h'} \forall \vec{v}(\exists x_n \gamma' \leftrightarrow \exists x_n \gamma)$, where $\vec{v}$ is $\texttt{Free}_T(\gamma) \cup \texttt{Free}_T(\gamma')$;

ES-$\forall$: For all $x_n \in \texttt{Var}_T$ and all ψ, $M \models_{\gamma(h)} \forall x_n \psi$ if and only if there exists a parameter assignment h' extending h and a first order formula γ' with $\texttt{Free}_P(\gamma') \subseteq \texttt{Dom}(h')$ such that

- $M \models_{\gamma'(h')} \psi$;
- $M \models_{h'} \forall \vec{v}(\gamma' \leftrightarrow \exists x_n \gamma)$, where $\vec{v}$ is $\texttt{Free}_T(\gamma) \cup \texttt{Free}_T(\gamma')$.

The reason why the above semantics is called "Entailment Semantics" is because its satisfaction relation describes a sort of *entailment relation* between a first order formula with parameters, which takes the role that teams have in the usual Team Semantics, and an independence logic formula. In particular, it is easy to see that according to our rule **ES-lit**, for all first order literals $\phi(\vec{x}, \vec{y})$, first order formulas with parameters $\gamma(\vec{x})$ and parameter assignments h, $M \models_{\gamma(h)} \phi$ if and only if $M \models_h \forall \vec{x}\vec{y}(\gamma(\vec{x}) \rightarrow \phi(\vec{x}, \vec{y}))$.

Furthermore, one can notice some analogies between Entailment Semantics and Database Theory: in particular, the role of γ in an expression $M \models_{\gamma(h)} \phi$ is to specify a relation in terms of a first order formula, much as in the Tuple Relational Calculus expression $\{\langle x_1 \ldots x_n \rangle : M \models \gamma(x_1 \ldots x_n, h(\vec{p}))\}$.

Proposition 5.2.4. *Let M be a first order model with signature Σ, let $\gamma(\vec{x}, \vec{p})$ be a first order formula with $\texttt{Free}_P(\gamma) = \vec{p}$ and let h, h' be two parameter assignments with domains containing $\vec{p}$ such that $h(\vec{p}) = h'(\vec{p})$. Then, for all formulas ϕ,*

$$M \models_{\gamma(h)} \phi \Leftrightarrow M \models_{\gamma(h')} \phi.$$

Proof. The proof is a straightforward induction over ϕ. □

As the next result shows, Entailment Semantics is entirely equivalent to Least General Model Semantics:

Theorem 5.2.5. *Let Σ be a first order model, let $\gamma(\vec{x}, \vec{p})$ be a first order formula with $\texttt{Free}_P(\gamma) = \vec{p}$, let h be a parameter assignment with domain $\vec{p}$ and let ϕ be a formula over this signature and with free variables in $\vec{x}$.*

Furthermore, let $(M, \mathcal{L})$ be the least general model over M, and let $X = \|\gamma(\vec{x}, \vec{p})\|_{M,h} = \{s : \texttt{Dom}(s) = \vec{x}, M \models_{h \cup s} \gamma(\vec{x}, \vec{m})\}$. Then

$$(M, \mathcal{L}) \models_X \phi \Leftrightarrow M \models_{\gamma(h)} \phi.$$

Proof. The proof is by structural induction on ϕ, and presents no difficulties.

1. If ϕ is a first order literal, $(M,\mathcal{L}) \models_X \phi$ if and only if, for all $s \in X$, it holds that $M \models_s \phi$. But $s \in X$ if and only if $M \models_s \gamma(\vec{x}, h(\vec{p}))$, and hence $(M,\mathcal{L}) \models_X \phi$ if and only if $M \models_\gamma \phi$, as required.

2. If ϕ is an inclusion, exclusion or independence atom, the result is also obvious, and follows at once from a comparison of the rules **GMS-inc** (**GMS-exc**, **GMS-ind**) and **ES-inc** (**ES-exc**, **ES-ind**).

3. If ϕ is $\psi_1 \vee \psi_2$,

$$\begin{aligned}
&(M,\mathcal{L}) \models_X \psi_1 \vee \psi_2 \Leftrightarrow\\
&\Leftrightarrow \exists Y, Z \in \mathcal{L} \text{ s.t. } X = Y \cup Z, (M,\mathcal{L}) \models_Y \psi_1 \text{ and } (M,\mathcal{L}) \models_Z \psi_2 \Leftrightarrow\\
&\Leftrightarrow \exists h' = h[\vec{m}/\vec{q}] \text{ extending } h \text{ and } \exists \gamma_1\gamma_2 \text{ s.t., for } Y = \|\gamma_1(\vec{x}, \vec{p}\vec{q})\|_{M,h'},\\
&\quad Z = \|\gamma_2(\vec{x}, \vec{p}\vec{q})\|_{M,h'}, X = \|\gamma(\vec{x}, \vec{p})\|_{M,h} = \|\gamma(\vec{x}, \vec{p})\|_{M,h'} = Y \cup Z,\\
&\quad (M,\mathcal{L}) \models_Y \psi_1 \text{ and } (M,\mathcal{L}) \models_Z \psi_2 \Leftrightarrow\\
&\Leftrightarrow \exists h' = h[\vec{m}/\vec{q}] \text{ extending } h \text{ and } \exists \gamma_1\gamma_2 \text{ s.t. } M \models_{h'} \forall \vec{v}(\gamma \leftrightarrow \gamma_1 \vee \gamma_2),\\
&\quad M \models_{\gamma_1(h')} \psi \text{ and } M \models_{\gamma_2(h')} \theta \Leftrightarrow\\
&\Leftrightarrow M \models_{\gamma(h)} \psi \vee \theta.
\end{aligned}$$

4. If ϕ is $\psi \wedge \theta$,

$$\begin{aligned}
&(M,\mathcal{L}) \models_X \psi \wedge \theta \Leftrightarrow (M,\mathcal{L}) \models_X \psi \text{ and } (M,\mathcal{L}) \models_X \theta \Leftrightarrow\\
&\Leftrightarrow M \models_{\gamma(h)} \psi \text{ and } M \models_{\gamma(h)} \theta \Leftrightarrow M \models_{\gamma(h)} \psi \wedge \theta.
\end{aligned}$$

5. If ϕ is $\exists x_n \psi$,

$$\begin{aligned}
&(M,\mathcal{L}) \models_X \exists x_n \psi \Leftrightarrow \exists X' \in \mathcal{L} \text{ s.t. } X[x_n]X' \text{ and } (M,\mathcal{L}) \models_{X'} \psi \Leftrightarrow\\
&\Leftrightarrow \exists h' = h[\vec{m}/\vec{q}] \text{ extending } h \text{ and } \exists \gamma' \text{ s.t., for } X' = \|\gamma'(\vec{x}, \vec{p}\vec{q})\|_{M,h'},\\
&\quad X[x_n]X' \text{ and } (M,\mathcal{L}) \models_{X'} \psi \Leftrightarrow\\
&\Leftrightarrow \exists h' = h[\vec{m}/\vec{q}] \text{ extending } h \text{ and } \exists \gamma' \text{ s.t. } M \models_{h'} \forall \vec{v}(\exists x_n \gamma \leftrightarrow \exists x_n \gamma')\\
&\quad \text{and } M \models_{\gamma'(h')} \psi \Leftrightarrow\\
&\Leftrightarrow M \models_{\gamma(h)} \exists x_n \psi;
\end{aligned}$$

6. If ϕ is $\forall x_n \psi$,

$$(M, \mathcal{L}) \models_X \forall x_n \psi \Leftrightarrow \exists X' \in \mathcal{L} \text{ s.t. } X' = X[M/x_n] \text{ and } (M, \mathcal{L}) \models_{X'} \psi \Leftrightarrow$$
$$\Leftrightarrow \exists h' = h[\vec{m}/\vec{q}] \text{ extending } h \text{ and } \exists \gamma' \text{ s.t., for } X' = \|\gamma'(\vec{x}, \vec{p}q)\|_{M,h'},$$
$$X' = X[M/x_n] \text{ and } (M, \mathcal{L}) \models_{X'} \psi \Leftrightarrow$$
$$\Leftrightarrow \exists h' = h[\vec{m}/\vec{q}] \text{ extending } h \text{ and } \exists \gamma' \text{ s.t. } M \models_{h'} \forall \vec{v}(\gamma' \leftrightarrow \exists x_n \gamma)$$
$$\text{and } M \models_{\gamma'(h')} \psi \Leftrightarrow$$
$$\Leftrightarrow M \models_{\gamma(h)} \forall x_n \psi.$$

□

Definition 5.2.6. Let ϕ be a formula. Then ϕ is *valid* in Entailment Semantics if and only if $M \models_{\gamma(h)} \phi$ for all first order models M with signature containing that of ϕ, for all first order formulas $\gamma(\vec{x}, \vec{p})$ over the signature of M and for all parameter assignments h with domain $\vec{p}$. If this is the case, we write ENS $\models \phi$.

Corollary 5.2.7. *For all formulas ϕ, ENS $\models \phi$ if and only if LMS $\models \phi$ if and only if GMS $\models \phi$*

It will also be useful to have a slightly more general notion of validity in Entailment Semantics:

Definition 5.2.8. Let $\gamma(\vec{x}, \vec{p})$ be a first order formula and let ϕ be a formula. Then ϕ is *valid* with respect to γ if and only if $M \models_{\gamma(h)} \phi$ for all first order models M with signature containing those of γ and ϕ and for all parameter assignments h with domain $\vec{p}$. If this is the case, we write $\models_\gamma \phi$.

Proposition 5.2.9. *Let ϕ be a formula with $\mathtt{Free}_T(\phi) = \{x_1 \ldots x_k\}$, let $\vec{x} = x_1 \ldots x_k$, and let R be a k-ary relation symbol not occurring in γ. Then ENS $\models \phi$ if and only if $\models_{R\vec{x}} \phi$.*

Proof. Suppose that ENS $\models \phi$. Then in particular, for any model M whose signature contains that of ϕ and R we have that $M \models_{R\vec{x}} \phi$, and hence $\models_{R\vec{x}} \phi$.

Conversely, suppose that $\models_{R\vec{x}} \phi$, let M be a first order model[4], and let $X \in \mathcal{L}$ be any team with domain $\{x_1 \ldots x_k\}$. Let us then consider the model M' obtained by adding to M the k-ary symbol R with $R^{M'} = \mathtt{Rel}(X)$. By hypothesis, $M' \models_{R\vec{x}} \phi$, and furthermore since $R^{M'}$ is in $\mathcal{L}$ already the least general model over M' is $(M', \mathcal{L})$ for the same $\mathcal{L}$.

[4]Without loss of generality, we can assume that the signature of M does not contain the symbol R.

Now $(M', \mathcal{L}) \models_X \phi$, and therefore, as R occurs nowhere in ϕ, $(M, \mathcal{L}) \models_X \phi$ too. This holds for all X with domains $\{x_1 \ldots x_k\}$; therefore by the Locality Theorem (Theorem 5.1.6), the same holds for all domains containing $\mathtt{Free}_T(\phi)$, and hence LMS $\models \phi$. This implies that ENS $\models \phi$, as required. □

In the next section, we will develop a sound and complete proof system for this notion of validity with respect to a team definition.

5.3 The Proof System

In this section, we will develop a proof system for $\mathcal{I}(\subseteq, |)$ and prove its soundness and completeness.

Definition 5.3.1. Let Γ be a finite first order theory with only parameter variables among its free ones, let $\gamma(\vec{x}, \vec{p})$ be a first order formula and let ϕ be a formula with free variables in $\mathtt{Var}_T$. Then the expression

$$\Gamma \mid \gamma \vdash \phi$$

is a sequent.

The intended semantics of a sequent is the following one:

Definition 5.3.2. Let $\Gamma \mid \gamma \vdash \phi$ be a sequent. Then $\Gamma \mid \gamma \vdash \phi$ is *valid* if and only if for all models M and all parameter assignments h with domain $\mathtt{Free}_P(\Gamma) \cup \mathtt{Free}_P(\gamma)$ such that $M \models_h \Gamma$ it holds that $M \models_{\gamma(h)} \phi$.

For example, the sequent $\emptyset \mid y = f(x) \vdash =\!(x, y)$ is valid, as any team in which y is $f(x)$ satisfies the condition corresponding to $=\!(x, y)$ (or, equivalently, to the independence atom $y \perp_x y$); and similarly, the sequent $\exists qr \forall u(Rpu \rightarrow (u = q \vee u = r)) \mid Rpx \vdash =\!(x) \vee =\!(x)$ is valid, because if $|\{m \in \mathtt{Dom}(M) : M \models_h Rpm\}| \leq 2$ then the team $(Rpx)(h) = \{s : M \models_h Rpx\}$ assigns no more than two different values for x and hence satisfies $=\!(x) \vee =\!(x)$.

However, $\emptyset \mid Rpx \vdash =\!(x) \vee =\!(x)$ is *not* valid: indeed, let $\mathtt{Dom}(M) = \{1, 2, 3\}$, let R^M be $\mathtt{Dom}(M) \times \mathtt{Dom}(M)$, and let h be such that $h(p) = 1$. Then $(Rpx)(h)$ is exactly $\mathtt{Dom}(M) = \{1, 2, 3\}$, which does not satisfy $=\!(x) \vee =\!(x)$.

The following result is then clear:

Proposition 5.3.3. *For all γ and ϕ, $\models_\gamma \phi$ if and only if $\emptyset \mid \gamma \vdash \phi$ is valid.*

Now, all we need to do is develop some syntactic rules for finding valid sequents.

We can do this as follows:

Definition 5.3.4. The axioms of our proof system are

PS-lit: If ϕ is a first order literal with no free parameter variables (that is, $\mathtt{Free}_P(\phi) = \emptyset$) then

$$\forall \vec{v}(\gamma \to \phi) \mid \gamma \vdash \phi$$

for all first order formulas γ, where $\vec{v} = \mathtt{Free}_T(\gamma) \cup \mathtt{Free}_T(\phi)$;

PS-inc: If $\vec{t}_1$ and $\vec{t}_2$ are tuples of terms of the same length with no parameter variables then

$$\forall \vec{v}_1(\gamma(\vec{v}_1) \to \exists \vec{v}_2(\gamma(\vec{v}_2) \wedge \vec{t}_1(\vec{v}_1) = \vec{t}_2(\vec{v}_2))) \mid \gamma \vdash \vec{t}_1 \subseteq \vec{t}_2$$

for all γ, where $\vec{v}_1$ and $\vec{v}_2$ are tuples of variables of the same lengths of $\vec{v} = \mathtt{Free}_T(\gamma) \cup \mathtt{Free}_T(\vec{t}_1\vec{t}_2)$, $\vec{t}_i(\vec{v}_i)$ is the tuple obtained by substituting $\vec{v}_i$ for $\vec{v}$ in $\vec{t}_i$, and the same holds for $\gamma(\vec{v}_i)$;

PS-exc: If $\vec{t}_1$ and $\vec{t}_2$ are tuples of terms of the same length with no parameter variables then

$$\forall \vec{v}_1 \forall \vec{v}_2((\gamma(\vec{v}_1) \wedge \gamma(\vec{v}_2)) \to \vec{t}_1(\vec{v}_1) \neq \vec{t}_2(\vec{v}_2)) \mid \gamma \vdash \vec{t}_1 \mid \vec{t}_2;$$

PS-ind: If $\vec{t}_1$, $\vec{t}_2$ and $\vec{t}_3$ are tuples of terms with no parameter variables then

$$\forall \vec{v}_1\vec{v}_2((\gamma(\vec{v}_1) \wedge \gamma(\vec{v}_2) \wedge \vec{t}_1(\vec{v}_1) = \vec{t}_1(\vec{v}_2)) \to \exists \vec{v}_3(\gamma(v_3) \wedge \vec{t}_1\vec{t}_2(\vec{v}_3) = \vec{t}_1\vec{t}_2(\vec{v}_1) \wedge \vec{t}_1\vec{t}_3(\vec{v}_3) = \vec{t}_1\vec{t}_3(\vec{v}_2))) \mid \gamma \vdash \vec{t}_2 \perp_{\vec{t}_1} \vec{t}_3.$$

The rules of our proof system are

PS-∨: If $\Gamma_1 \mid \gamma_1 \vdash \phi_1$ and $\Gamma_2 \mid \gamma_2 \vdash \phi_2$ then, for all γ, we have

$$\Gamma_1, \Gamma_2, \forall \vec{v}(\gamma \leftrightarrow (\gamma_1 \vee \gamma_2)) \mid \gamma \vdash \phi_1 \vee \phi_2$$

where $\vec{v}$ is $\mathtt{Free}_T(\gamma) \cup \mathtt{Free}_T(\gamma_1) \cup \mathtt{Free}_T(\gamma_2)$;

PS-∧: If $\Gamma_1 \mid \gamma \vdash \phi_1$ and $\Gamma_2 \mid \gamma \vdash \phi_2$ then $\Gamma_1, \Gamma_2 \mid \gamma \vdash \phi_1 \wedge \phi_2$;

PS-∃: If $\Gamma \mid \gamma' \vdash \phi$ and x is a team variable then, for all γ,

$$\Gamma, \forall \vec{v}(\exists x \gamma' \leftrightarrow \exists x \gamma) \mid \gamma \vdash \exists x \phi$$

where $\vec{v} = \texttt{Free}_T(\gamma) \cup \texttt{Free}_T(\gamma')$;

PS-∀: If $\Gamma \mid \gamma' \vdash \phi$ and x is a team variable then, for all γ,

$$\Gamma, \forall \vec{v}(\gamma' \leftrightarrow \exists x \gamma) \mid \gamma \vdash \forall x \phi$$

where, as in the previous case, $\vec{v} = \texttt{Free}_T(\gamma) \cup \texttt{Free}_T(\gamma')$;

PS-ent: If $\Gamma \mid \gamma \vdash \phi$ and $\bigwedge \Gamma' \models \bigwedge \Gamma$ holds in First Order Logic then $\Gamma' \mid \gamma \vdash \phi$;

PS-depar: If $\Gamma \mid \gamma \vdash \phi$ and p is a parameter variable which does not occur free in γ then $\exists p \bigwedge \Gamma \mid \gamma \vdash \phi$;

PS-split: If $\Gamma_1 \mid \gamma \vdash \phi$ and $\Gamma_2 \mid \gamma \vdash \phi$ then $(\bigwedge \Gamma_1) \vee (\bigwedge \Gamma_2) \mid \gamma \vdash \phi$.

Definition 5.3.5. Let $\Gamma \mid \gamma \vdash \phi$ be a sequent. A *proof* of this sequent is a finite list of sequents

$$(\Gamma_1 \mid \gamma_1 \vdash \phi_1), \ldots, (\Gamma_n \mid \gamma_n \vdash \phi_n) = (\Gamma \mid \gamma \vdash \phi)$$

such that, for all $i = 1 \ldots n$, $\Gamma_i \mid \gamma_i \vdash \phi_i$ is either an instance of **PS-lit**, **PS-inc**, **PS-exc**, **PS-ind** or it follows from $\{\Gamma_j \mid \gamma_j \vdash \phi_j : j < i\}$ through one application of the rules of our proof system.

Given a proof $P = S_1 \ldots S_n$, where each S_i is a sequent, we define its *length* $|P|$ as $n-1$, that is, as the number of sequents in the proof minus one.

Before examining soundness and completeness for this proof system, it will be useful to derive a general rule for first order formulas.

Proposition 5.3.6. PS-FO: *If ϕ is a first order formula with no free parameter variables, $\forall \vec{v}(\gamma \to \phi) \mid \gamma \vdash \phi$ is provable for all γ, where $\vec{v} = \texttt{Free}_T(\gamma) \cup \texttt{Free}_T(\phi)$;*

Proof. The proof is by structural induction on ϕ.

1. If ϕ is a first order literal, this follows at once from rule **PS-lit**.

2. If ϕ is $\psi_1 \vee \psi_2$, by induction hypothesis we have that $\forall \vec{v}((\gamma \wedge \psi_1) \to \psi_1) \mid \gamma \wedge \psi_1 \vdash \psi_1$ and $\forall \vec{v}((\gamma \wedge \psi_2) \to \psi_2) \mid \gamma \wedge \psi_2 \vdash \psi_2$ are provable. But then we can prove $\forall \vec{v}(\gamma \to \phi_1 \vee \phi_2) \mid \gamma \vdash \phi$ as follows:

   (a) $\forall \vec{v}((\gamma \wedge \psi_1) \to \psi_1) \mid \gamma \wedge \psi_1 \vdash \psi_1$ (Derived before)

   (b) $\forall \vec{v}((\gamma \wedge \psi_2) \to \psi_2) \mid \gamma \wedge \psi_2 \vdash \psi_2$ (Derived before)

(c) $\mid \gamma \wedge \psi_1 \vdash \psi_1$ (**PS-ent**, from (a), because $\models \forall\vec{v}((\gamma \wedge \psi_1) \rightarrow \psi_1)$ in First Order Logic)

(d) $\mid \gamma \wedge \psi_2 \vdash \psi_2$ (**PS-ent**, from (b), because $\models \forall\vec{v}((\gamma \wedge \psi_2) \rightarrow \psi_2)$ in First Order Logic)

(e) $\forall\vec{v}(\gamma \leftrightarrow (\gamma \wedge \psi_1) \vee (\gamma \wedge \psi_2)) \mid \gamma \vdash \psi_1 \vee \psi_2$ (**PS-**$\vee$, from (c) and (d))

(f) $\forall\vec{v}(\gamma \rightarrow (\psi_1 \vee \psi_2)) \mid \gamma \vdash \psi_1 \vee \psi_2$ (**PS-ent**: from (e), because $\forall\vec{v}(\gamma \rightarrow (\psi_1 \vee \psi_2))$ entails $\forall\vec{v}(\gamma \leftrightarrow (\gamma \wedge \psi_1) \vee (\gamma \wedge \psi_2))$ in First Order Logic).

3. If ϕ is $\psi_1 \wedge \psi_2$, by induction hypothesis we have that $\forall\vec{v}(\gamma \rightarrow \psi_1) \mid \gamma \vdash \psi_1$ and $\forall\vec{v}(\gamma \rightarrow \psi_2) \mid \gamma \vdash \psi_2$ are provable. But then

   (a) $\forall\vec{v}(\gamma \rightarrow \psi_1) \mid \gamma \vdash \psi_1$ (derived before)

   (b) $\forall\vec{v}(\gamma \rightarrow \psi_2) \mid \gamma \vdash \psi_2$ (derived before)

   (c) $\forall\vec{v}(\gamma \rightarrow \psi_1), \forall\vec{v}(\gamma \rightarrow \psi_2) \mid \gamma \vdash \psi_1 \wedge \psi_2$ (**PS-**$\wedge$, (a), (b))

   (d) $\forall\vec{v}(\gamma \rightarrow \psi_1 \wedge \psi_2) \mid \gamma \vdash \psi_1 \wedge \psi_2$ (**PS-ent**, (c))

   as required.

4. If ϕ is $\exists x\psi$, by induction hypothesis we have that $\forall\vec{v}\forall x(((\exists x\gamma) \wedge \psi) \rightarrow \psi) \mid (\exists x\gamma) \wedge \psi \vdash \psi$ is provable. But then

   (a) $\forall\vec{v}\forall x(((\exists x\gamma) \wedge \psi) \rightarrow \psi) \mid (\exists x\gamma) \wedge \psi \vdash \psi$ (derived before)

   (b) $\mid (\exists x\gamma) \wedge \psi \vdash \psi$ (**PS-ent**, from (a))

   (c) $\forall\vec{v}(\exists x((\exists x\gamma) \wedge \psi) \leftrightarrow \exists x\gamma) \mid \gamma \vdash \exists x\psi$ (**PS-**$\exists$, from (b))

   (d) $\forall\vec{v}(((\exists x\gamma) \wedge (\exists x\psi)) \leftrightarrow \exists x\gamma) \mid \gamma \vdash \exists x\psi$ (**PS-ent**, from (c))

   (e) $\forall\vec{v}(\gamma \rightarrow \exists x\psi) \mid \gamma \vdash \psi$ (**PS-ent**, from (d))

   as required, where the last passage uses the fact that $\forall\vec{v}(\gamma \rightarrow \exists x\psi) \models \forall\vec{v}(((\exists x\gamma) \wedge (\exists x\psi)) \leftrightarrow \exists x\gamma)$ in First Order Logic.

5. If ϕ is $\forall x\psi$, by induction hypothesis we have that $\forall\vec{v}\forall x((\exists x\gamma) \rightarrow \psi) \mid \exists x\gamma \vdash \psi$ is provable. But then

   (a) $\forall\vec{v}\forall x((\exists x\gamma) \rightarrow \psi) \mid \exists x\gamma \vdash \psi$ (derived before)

   (b) $\forall\vec{v}\forall x((\exists x\gamma) \rightarrow \psi), \forall\vec{v}(\exists x\gamma \leftrightarrow \exists x\gamma) \mid \gamma \vdash \forall x\psi$ (**PS-**$\forall$, from (a))

   (c) $\forall\vec{v}\forall x((\exists x\gamma) \rightarrow \psi) \mid \gamma \vdash \forall x\psi$ (**PS-ent**, from (c))

   (d) $\forall\vec{v}(\gamma \rightarrow \forall x\psi) \mid \gamma \vdash \forall x\psi$ (**PS-ent**, from (d)) where the last two passages hold because $\forall\vec{v}(\exists x\gamma \leftrightarrow \exists x\gamma)$ is valid and because $\forall\vec{v}(\gamma \to \forall x\psi)$ entails $\forall\vec{v}\forall x((\exists x\gamma) \to \psi)$ in first order logic, where $\vec{v} = \mathtt{Free}_T(\gamma) \cup \mathtt{Free}_T(\psi)$ (and, therefore, if x is free in γ then x is in $\vec{v}$).

□

Theorem 5.3.7 (Soundness)**.** *Suppose that* $\Gamma \mid \gamma \vdash \phi$ *is provable. Then it is valid.*

Proof. If S is a provable sequent then there exists a proof $S_1 \dots S_n S$ for it. Then we go by induction of the length n of this proof:

Base case: Suppose that the proof has length 0. Then S is an instance of **PS-lit**, of **PS-inc**, of **PS-exc** or of **PS-ind**. Assume first that it is the former, that is, that

$$S = \forall\vec{v}(\gamma \to \phi) \mid \gamma \vdash \phi$$

for some first order γ and some first order literal ϕ, where $\vec{v} = \mathtt{Free}_T(\gamma) \cup \mathtt{Free}_T(\phi)$ and ϕ has no parameter variables. Now suppose that $M \models_h \forall\vec{x}(\gamma \to \phi)$; then, by definition, if s is an assignment over team variables such that $M \models_{h\cup s} \gamma$ then $M \models_s \phi$. Therefore, by **ES-lit**, $M \models_{\gamma(s)} \phi$ in Entailment Semantics, as required.

The other cases are treated in an entirely similar manner.

Induction case: Let $S_1 S_2 \dots S_n S$ be our proof. For each $i \leq n$ we have that $S_1 \dots S_i$ is a valid proof for S_i, and hence by induction hypothesis that S_i is valid. Now let us consider which rule r was been used to derive S from $S_1 \dots S_n$:

1. If r was **PS-lit** or **PS-ind** then (S) is a proof for S already, and hence by our base case S is valid;
2. If r was **PS-**$\vee$ then S is $\Gamma_1, \Gamma_2, \forall\vec{v}(\gamma \leftrightarrow (\gamma_1 \vee \gamma_2)) \mid \gamma \vdash \phi_1 \vee \phi_2$, and there exist two $i, j \leq n$ such that $S_i = (\Gamma_1 \mid \gamma_1 \vdash \phi_1)$ and $S_j = (\Gamma_2 \mid \gamma_2 \vdash \phi_2)$. By induction hypothesis, these sequents are valid.

   Now suppose that $M \models_h \Gamma_1, \Gamma_2, \forall\vec{v}(\gamma \leftrightarrow (\gamma_1 \vee \gamma_2))$. Then, since $M \models_h \Gamma_1$, we have that $M \models_{\gamma_1(h)} \phi_1$, and, analogously, since $M \models_h \Gamma_2$ we have that $M \models_{\gamma_2(h)} \phi_2$. Furthermore, $M \models_h \forall\vec{v}(\gamma \leftrightarrow \gamma_1 \vee \gamma_2)$, and therefore by rule **ES-**$\vee$ we have that $M \models_\gamma \phi_1 \vee \phi_2$, as required.

3. If r was **PS-$\wedge$** then S_n is of the form $\Gamma_1, \Gamma_2 \mid \gamma \vdash \phi_1 \wedge \phi_2$ and, by induction hypothesis, $\Gamma_1 \mid \gamma \vdash \phi_1$ and $\Gamma_2 \mid \gamma \vdash \phi_2$ are valid. Now suppose that $M \models_h \Gamma_1, \Gamma_2$; then $M \models_{\gamma(h)} \phi_1$ and $M \models_{\gamma(h)} \phi_2$, and therefore $M \models_{\gamma(h)} \phi_1 \wedge \phi_2$ by **ES-$\wedge$**.

4. If r was **PS-$\exists$** then S_n is of the form $\Gamma, \forall \vec{v}(\exists x\gamma' \leftrightarrow \exists x\gamma) \mid \gamma \vdash \exists x\phi$, where $\Gamma \mid \gamma' \vdash \phi$ is valid by induction hypothesis. Now suppose that $M \models_h \Gamma, \forall \vec{v}(\exists x\gamma \leftrightarrow \exists x\gamma')$; then $M \models_{\gamma'(h)} \phi$ and $M \models_h \forall \vec{v}(\exists x\gamma \leftrightarrow \exists x\gamma')$, and therefore $M \models_{\gamma(h)} \exists x\phi$ by rule **ES-$\exists$**.

5. If r was **PS-$\forall$** then S_n is of the form $\Gamma, \forall \vec{v}(\gamma' \leftrightarrow \exists x\gamma) \mid \gamma \vdash \forall x\phi$, where $\Gamma \mid \gamma' \vdash \phi$ is valid by induction hypothesis. Now, suppose that $M \models_h \Gamma, \forall \vec{v}(\gamma' \leftrightarrow \exists x\gamma)$. Then $M \models_{\gamma'(h)} \phi$, and furthermore $M \models_h \forall \vec{v}(\gamma' \leftrightarrow \exists x\gamma)$. Therefore, by rule **ES-$\forall$**, $M \models_{\gamma(h)} \forall x\phi$, as required.

6. If r was **PS-ent** then S_n is of the form $\Gamma' \mid \gamma \vdash \phi$, where $\Gamma \mid \gamma \vdash \phi$ is valid by induction hypothesis and where $\bigwedge \Gamma \models \bigwedge \Gamma'$ holds in first order logic. Now suppose that $M \models_h \Gamma'$; then $M \models_h \Gamma$, and hence $M \models_{\gamma(h)} \phi$, as required.

7. If r was **PS-depar** then S_n is of the form $\exists p \bigwedge \Gamma \mid \gamma \vdash \phi$, where $\Gamma \mid \gamma \vdash \phi$ holds by induction hypothesis and where the parameter variable p does not occur free in γ. Now suppose that $M \models_h \exists p \bigwedge \Gamma$; then there exists an element $m \in \texttt{Dom}(M)$ such that, for $h' = h[m/p]$, $M \models_{h'} \Gamma$. Then $M \models_{\gamma(h')} \phi$; but as p does not occur free in γ we then have, by Proposition 5.2.4, that $M \models_{\gamma(h)} \phi$ as required.

8. If r was **PS-split** then S_n is of the form $(\bigwedge \Gamma_1) \vee (\bigwedge \Gamma_2) \mid \gamma \vdash \phi$, where $\Gamma_1 \mid \gamma \vdash \phi$ and $\Gamma_2 \mid \gamma \vdash \phi$ by induction hypothesis. Now suppose that $M \models_h (\bigwedge \Gamma_1) \vee (\bigwedge \Gamma_2)$. Then $M \models_h \Gamma_1$ or $M \models_h \Gamma_2$; and in either case, $M \models_{\gamma(h)} \phi$, as required.

□

In order to prove completeness, we first need a lemma:

Lemma 5.3.8. *Suppose that $M \models_{\gamma(h)} \phi$. Then there exists a finite Γ such that $\Gamma \mid \gamma \vdash \phi$ is provable and such that $M \models_h \Gamma$.*

Proof. The proof is by structural induction on ϕ.

1. If ϕ is a first order literal or an inclusion/exclusion/independence atom, this follows immediately.

2. If ϕ is $\psi_1 \vee \psi_2$ and $M \models_{\gamma(h)} \phi$ then, by definition, there exists an assignment h' extending h and two first order formulas γ_1, γ_2 such that $M \models_{\gamma_1(h')} \psi_1$, $M \models_{\gamma_2(h')} \psi_2$ and $M \models_{h'} \forall\vec{v}(\gamma \leftrightarrow \gamma_1 \vee \gamma_2)$. Let $\vec{p}$ be the tuple of parameters in $\mathtt{Dom}(h')\backslash\mathtt{Dom}(h)$; now, by induction hypothesis we have that there exist Γ_1 and Γ_2 such that $\Gamma_1 \mid \gamma_1 \vdash \psi_1$ and $\Gamma_2 \mid \gamma_2 \vdash \psi_2$ are provable, and such that furthermore $M \models_{h'} \Gamma_1$ and $M \models_{h'} \Gamma_2$.

   But then the following is a correct proof:

   (a) $\Gamma_1 \mid \gamma_1 \vdash \psi_1$ (Derived before)

   (b) $\Gamma_2 \mid \gamma_2 \vdash \psi_2$ (Derived before)

   (c) $\Gamma_1, \Gamma_2, \forall\vec{v}(\gamma \leftrightarrow \gamma_1 \vee \gamma_2) \mid \gamma \vdash \phi$ (**PS-**$\vee$, (a), (b))

   (d) $\exists\vec{p}(\bigwedge\Gamma_1 \wedge \bigwedge\Gamma_2 \wedge \forall\vec{v}(\gamma \leftrightarrow \gamma_1 \vee \gamma_2)) \mid \gamma \vdash \phi$ (**PS-depar**, (c))[5]

   Finally, $M \models_h \exists\vec{p}(\bigwedge\Gamma_1 \wedge \bigwedge\Gamma_2 \wedge \forall\vec{v}(\gamma \leftrightarrow \gamma_1 \vee \gamma_2))$, as required, because there exists a tuple of elements $\vec{m}$ such that $h[\vec{m}/\vec{p}] = h'$.

3. If ϕ is $\psi_1 \wedge \psi_2$ and $M \models_{\gamma(h)} \phi$, then $M \models_{\gamma(h)} \psi_1$ and $M \models_{\gamma(h)} \psi_2$. Then, by induction hypothesis, there exist Γ_1 and Γ_2 such that $\Gamma_1 \mid \gamma \vdash \psi_1$ and $\Gamma_2 \mid \gamma \vdash \psi_2$ are provable and such that $M \models_h \Gamma_1\Gamma_2$. Then by rule **PS-**$\wedge$, $\Gamma_1\Gamma_2 \mid \gamma \vdash \psi_1 \wedge \psi_2$, as required.

4. If ϕ is $\exists x\psi$ and $M \models_{\gamma(h)} \phi$, then there exists a tuple $\vec{p}$ of parameter variables not in the domain of h, a tuple $\vec{m}$ of elements of the model and a formula γ' such that, for $h' = h[\vec{m}/\vec{p}]$, $M \models_{\gamma'(h')} \psi$ and $M \models_{h'} \forall\vec{v}(\exists x\gamma' \leftrightarrow \exists x\gamma)$. By induction hypothesis, we then have a Γ' such that $\Gamma' \mid \gamma' \vdash \psi$ and $M \models_{h'} \Gamma'$.

   Then the following is a valid proof:

   (a) $\Gamma' \mid \gamma' \vdash \psi$ (Derived before)

   (b) $\Gamma', \forall\vec{v}(\exists x\gamma' \leftrightarrow \exists x\gamma) \mid \gamma \vdash \exists x\psi$ (**PS-**$\exists$)

   (c) $\exists\vec{p}(\bigwedge\Gamma' \wedge \forall\vec{v}(\exists x\gamma' \leftrightarrow \exists x\gamma)) \mid \gamma \vdash \exists x\psi$ (**PS-depar**)

   Furthermore, $M \models_h \exists\vec{p}(\bigwedge\Gamma' \wedge \forall\vec{v}(\exists x\gamma' \leftrightarrow \exists x\gamma))$, as required.

5. If ϕ is $\forall x\psi$ and $M \models_{\gamma(h)} \phi$, then there exists a tuple $\vec{p}$ of parameter variables not in the domain of h, a tuple $\vec{m}$ of elements of the model and a formula γ' such that $M \models_{\gamma'(h')} \psi$ and $M \models_{h'} \forall\vec{v}(\gamma' \leftrightarrow \exists x\gamma)$, where

[5] To be entirely formal, this passage consists of $|\vec{p}|$ distinct applications of **PS-depar**, all of which are correct because none of the parameters in $\vec{p}$ appear in γ.

$h' = h[\vec{m}/\vec{p}]$. By induction hypothesis, we can then find a Γ' such that $\Gamma' \mid \gamma' \vdash \psi$ is provable and $M \models_{h'} \Gamma'$.

Then the following is a valid proof:

(a) $\Gamma' \mid \gamma' \vdash \psi$ (Derived before)

(b) $\Gamma', \forall \vec{v}(\gamma' \leftrightarrow \exists x\gamma) \mid \gamma \vdash \forall x\psi$ (**PS-**$\forall$)

(c) $\exists \vec{p}(\bigwedge \Gamma' \wedge \forall \vec{v}(\gamma' \leftrightarrow \exists x\gamma)) \mid \gamma \vdash \forall x\psi$ (**PS-depar**)

And, once again, the assignment h satisfies the antecedent of the last sequent, as required.

□

The completeness of our proof system follows from the above lemma and from the compactness and the Löwenheim-Skolem theorem for First Order Logic:

Theorem 5.3.9 (Completeness)**.** *Suppose that* $\Gamma \mid \gamma \vdash \phi$ *is valid, where* Γ *is finite. Then it is provable.*

Proof. Since $\Gamma \mid \gamma \vdash \phi$ is valid, for any first order model M over the signature of Γ, γ and ϕ and for all h such that $M \models_h \Gamma$ we have that $M \models_{\gamma(h)} \phi$, and hence by the lemma that $M \models_h \Gamma_{M,h}$ for some finite $\Gamma_{M,h}$ such that $\Gamma_{M,h} \mid \gamma \vdash \phi$ is provable.

Then consider the first order, countable[6] theory

$$T = \{\bigwedge \Gamma\} \cup \{\neg \bigwedge \Gamma_{M,h} : M \text{ is a countable model, } h \text{ is an assignment s.t. } M \models_h \Gamma\}.$$

This theory is unsatisfiable. Indeed, suppose that M_0 is a model that satisfies $\bigwedge \Gamma$ under the assignment h_0: then, by the Löwenheim-Skolem theorem, there exists a countable elementary submodel (M_0', h_0') of (M_0, h_0).

Now, $M_0' \models_{h_0'} \Gamma$ and M_0' is countable, and hence by definition $M_0' \models_{h_0'} \Gamma_{M_0',h_0'}$.

But then $M_0 \models_{h_0} \Gamma_{M_0',h_0'}$ too, and therefore M_0 is not a model of T.

By the compactness theorem, this implies that there exists a finite subset $T_0 = \{\neg \bigwedge \Gamma_{M_1,h_1}, \ldots, \neg \bigwedge \Gamma_{M_n,h_n}\}$ of T such that $\{\bigwedge \Gamma\} \cup T_0$ is unsatisfiable,

[6]The fact that it is countable follows at once from the fact that it is a first order theory over a countable vocabulary.

that is, such that

$$\Gamma \models (\bigwedge \Gamma_{M_1,h_1}) \vee \ldots \vee (\bigwedge \Gamma_{M_n,h_n}).$$

Now, for each i, $\Gamma_{M_i,h_i} \mid \gamma \vdash \phi$ can be proved. Therefore, by rule **PS-split**, we have that $(\bigwedge \Gamma_{M_1,s_1}) \vee \ldots \vee (\bigwedge \Gamma_{M_n,s_n}) \mid \gamma \vdash \phi$ is also provable; and finally, by rule **PS-ent** we can prove that $\Gamma \mid \gamma \vdash \phi$, as required. □

Thus, we succeeded in designing a proof system which is sound and complete with respect to our semantics; and as we saw, with respect to finite models our semantics is identical to the standard one, and furthermore even with respect to infinite models it is a natural generalization of Team Semantics.

Using essentially the same method, it is also possible to prove a "compactness" result for our semantics:

Theorem 5.3.10. *Suppose that $\Gamma \mid \gamma \vdash \phi$ is valid. Then there exists a finite $\Gamma_0 \subseteq \Gamma$ such that $\Gamma_0 \mid \gamma \vdash \phi$ is provable (and valid).*

Proof. Let $\kappa = \max(|\Gamma|, \aleph_0)$, and consider the theory

$$T = \Gamma \cup \{\neg \bigwedge \Gamma_{M,h} : |M| \leq \kappa, M \models_h \Gamma\}$$

where, as in the previous proof, $\Gamma_{M,h}$ is a finite theory such that $M \models_h \Gamma_{M,h}$ and such that $\Gamma_{M,h} \mid \gamma \vdash \phi$ is provable in our system.

Then T is unsatisfiable: indeed, if T had a model then it would have a model (M, h) of cardinality at most κ, and since that model would satisfy Γ it would satisfy $\Gamma_{M,h}$ too, which contradicts our hypothesis.

Hence, by the compactness theorem, there exists a finite set $\{\bigwedge \Gamma_{M_1,h_1}, \ldots, \bigwedge \Gamma_{M_n,h_n}\}$ and a finite $\Gamma_0 \subseteq \Gamma$ such that

$$\Gamma_0 \models \bigwedge \Gamma_{M_1,h_1} \vee \ldots \vee \bigwedge \Gamma_{M_n,h_n}.$$

But by rule **PS-split**, we have that $\bigwedge \Gamma_{M_1,h_1} \vee \ldots \vee \bigwedge \Gamma_{M_n,h_n} \mid \gamma \vdash \phi$ is provable, and hence by rule **PS-ent** $\Gamma_0 \mid \gamma \vdash \phi$ is also provable, as required. □

5.4 Adding More Teams

The proof system that we developed in the previous section is, as we saw, sound and complete with respect to its intended semantics. However, this semantics is perhaps rather weak: all that we know is that the teams which correspond to parametrized first order formulas belong in our general models.

Rather than adding more and more axioms to our proof system in order to guarantee the existence of more teams, in this section we will attempt to separate our assumptions about team existence from our main proof system. This will allow us to make our formalism *modular*: depending on our needs, we may want to assume the existence of more or of less teams in our general model.

The natural language for describing assertions about the existence of relations is of course, existential second order logic. The following definitions show how it can be used for our purposes:

Definition 5.4.1. A *relation existence theory* Θ is a set of existential second order sentences of the form $\exists \vec{R} \phi(\vec{R})$, where ϕ is first order.

Definition 5.4.2. Let $(M, \mathcal{G})$ be a general model, and let Θ be a relation existence theory. Then $(M, \mathcal{G})$ is Θ-*closed* if and only if for all $\exists \vec{R} \phi(\vec{R})$ in Θ there exists a tuple of teams $\vec{X} \in \mathcal{G}$ such that $M \models \phi[\vec{\texttt{Rel}}(\vec{X})/\vec{R}]$.

Definition 5.4.3. Let $\Gamma \mid \gamma \vdash \phi$ be a sequent and let Θ be a relation existence theory. Then $\Gamma|\gamma \vdash \phi$ is *valid* if and only if for all Θ-closed models $(M.\mathcal{G})$ and all parameter assignments h with domain $\texttt{Free}_P(\Gamma) \cup \texttt{Free}_P(\gamma)$ such that $M \models_h \Gamma$ it holds that

$$(M, \mathcal{G}) \models_{\|\gamma\|_h} \phi.$$

Our proof system for Θ-closed general models can then be obtained by adding the following rule to our system:

PS-Θ: If $\Gamma_1(\vec{S}), \Gamma_2 \mid \gamma \vdash \phi$ is provable, where the relation symbols $\vec{S}$ do not occur in Γ_2, in γ or in ϕ, and $\exists \vec{R} \bigwedge \Gamma_1(\vec{R})$ is in Θ for some $\vec{R}$ then $\Gamma_2 \mid \gamma \vdash \phi$ is provable.

Theorem 5.4.4 (Soundness)**.** *Let $\Gamma \mid \gamma \vdash \phi$ be a sequent which is provable in our proof system plus **PS**-Θ. Then it is Θ-valid.*

Proof. The proof is by induction on the length of the proof, and follows very closely the one given already. Hence, we only examine the case in which the last rule used in the proof is **PS-Θ**. Then, by induction hypothesis, we have that $\Gamma_1(\vec{S}), \Gamma \mid \gamma \vdash \phi$ is Θ-valid for some Γ_1 and some $\vec{S}$ which does not occur in Γ, in γ or in ϕ, and moreover $\exists \vec{R} \bigwedge \Gamma_1(\vec{R})$ is in Θ.

Now, let $(M, \mathcal{G})$ be any Θ-closed general model, and let us assume without loss of generality that the relation symbols in $\vec{S}$ are not part of its signature. Furthermore, let h be a parameter assignment (with domain $\texttt{Free}(\Gamma) \cup \texttt{Free}(\gamma)$)

such that $M \models_h \Gamma$. By definition, there exists a tuple of teams $\vec{X} \in \mathcal{G}$ such that $M \models \bigwedge \Gamma_1[\vec{\mathtt{Rel}}(\vec{X})/\vec{S}]$. Now let M' be $M[\vec{\mathtt{Rel}}(\vec{X})/\vec{S}]$: since $\vec{X}$ is in $\mathcal{G}$, it is not difficult to see that $(M', \mathcal{G})$ is a general model. Furthermore, it is Θ-closed, $M' \models \Gamma_1$, and $M' \models_h \Gamma$. Hence, $(M', \mathcal{G}) \models_{\|\gamma\|_h} \phi$; but since the relation symbols $\vec{S}$ do not occur in γ or in ϕ, this implies that $(M, \mathcal{G}) \models_{\|\gamma\|_h} \phi$. □

In order to prove completeness, we first need a definition and a simple lemma.

Definition 5.4.5. Let Θ be a relation existence theory. Then Θ^{FO} is the theory $\{\theta_i[\vec{S}_i/\vec{R}] : \exists \vec{R} \theta_i(\vec{R}) \in \Theta\}$, where the tuples of symbols $\vec{S}_i$ are all disjoint and otherwise unused.

Lemma 5.4.6. *Let Θ be a relation existence theory and let M be a model such that $M \models \Theta^{FO}$. Then the least general model $(M, \mathcal{L})$ over it is Θ-closed.*

Proof. Consider any $\exists \vec{R} \theta(\vec{R}) \in \Theta$. Then $M \models \theta(\vec{S}_i)$, for some tuple of relation symbols $\vec{S}_i$ in the signature of M. Then, the teams $\vec{X}$ associated to the corresponding relations are in $\mathcal{L}$, and for these teams we have that $M \models \theta[\vec{\mathtt{Rel}}(\vec{X})/\vec{R}]$, as required. □

Theorem 5.4.7 (Completeness)**.** *Suppose that $\Gamma \mid \gamma \vdash \phi$ is Θ-valid. Then it is provable in our proof system plus **PS**-Θ.*

Proof. Let M be any first order model satisfying Θ^{FO}, where we assume that the relation symbols used in the construction of Θ^{FO} do not occur in Γ, in γ or in ϕ. Then, by the lemma, $(M, \mathcal{L})$ is Θ-closed, and this implies that, for all assignments h such that $M \models_h \Gamma$, $M \models_{\|\gamma\|_h} \phi$.

Therefore, $\Theta^{FO}, \Gamma \mid \gamma \vdash \phi$ is valid; and hence, for some finite $\Delta \subseteq \Theta^{FO}$ it holds that $\Delta, \Gamma \mid \gamma \vdash \phi$ is provable. Now we can get rid of Δ through repeated applications of rule **PS-**Θ and, therefore, prove that $\Gamma \mid \gamma \vdash \phi$, as required. □

5.5 Conclusions

We began this chapter by defining a general semantics for a logic of imperfect information. Then we proved that – owing to the relationships between it and existential second order logic – in order to study validity with respect to this semantics it suffices to examine *least* general models. We then showed that, because of the correspondence between teams in least general models and first order formulas with parameters, we could limit ourselves to study *entailments* between first-order team-defining formulas and independence logic formulas. Finally, we developed a sound and complete proof system for this semantics, and we showed that this system can easily be strengthened by assuming the existence of more teams.

As we said, the correspondence between our logic and *existential* second order logic is of essential importance for the construction which we described: extending our approach to such logics as team logic or intuitionistic dependence logic promises to be nontrivial, although certainly not impossible. The relationship between our approach and the one developed by Kontinen and Väänänen in [52] is also certainly worth investigating.

Furthermore, Entailment Semantics – the key ingredient of our construction, and our "bridge" between General Model Semantics and the proof system – is, as we wrote, of independent interest for a more syntactic approach to the study of dependence and independence, and more in general to the study of this interesting family of logics.

Chapter 6

Transition Dynamics

In this chapter, we will extend the mutual embedding relation between Dynamic Game Logic and First Order Logic proved by van Benthem (and presented here in Section 6.1) to a relation between Dependence Logic and a suitable imperfect-information, player-versus-Nature variant of Dynamic Game Logic, which we will call *Transition Logic* (Section 6.2).

This will allow us to reinterpret Dependence Logic as a logic for modeling *decision problems* under imperfect information; and in Section 6.3, we will exploit this intuition to develop a *dynamic* version of Dependence Logic in which formulas are interpreted in terms of *transitions* from information states (teams) to information states.

6.1 On Dynamic Game Logic and First Order Logic

6.1.1 Dynamic Game Logic

Game logics are logical formalisms which contain two different kinds of expressions:

1. *Game terms*, which are descriptions of games in terms of compositions of *atomic games*;

2. *Formulas*, which, in general, correspond to assertions about the abilities of players in games.

In this subsection, we are going to summarize the definition of a variant of Dynamic Game Logic [59].[1] Then, in the next subsection, we will discuss a remarkable connection between First-Order Logic and Dynamic Game Logic discovered by Johan van Benthem in [69].

One of the fundamental semantic concepts of Dynamic Game Logic is the notion of *forcing relation:*

Definition 6.1.1. Let S be a nonempty set of *states*. A *forcing relation* over S is a set $\rho \subseteq S \times \mathtt{Parts}(S)$.

In brief, a forcing relation specifies the abilities of a player in a perfect-information game: $(s, X) \in \rho$ if and only if the player has a strategy that guarantees that, whenever the initial position of the game is s, the terminal position of the game will be in X.

A (two-player) *game* is then defined as a pair of forcing relations satisfying some axioms:

Definition 6.1.2. Let S be a nonempty set of states. A *game* over S is a pair (ρ^E, ρ^A) of forcing relations over S satisfying the following conditions for all $i \in \{E, A\}$, all $s \in S$ and all $X, Y \subseteq S$:

Monotonicity: If $(s, X) \in \rho^i$ and $X \subseteq Y$ then $(s, Y) \in \rho^i$;

Consistency: If $(s, X) \in \rho^E$ and $(s, Y) \in \rho^A$ then $X \cap Y \neq \emptyset$;

Non-triviality: $(s, \emptyset) \notin \rho^i$.

Determinacy: If $(s, X) \notin \rho^i$ then $(s, S \backslash X) \in \rho^j$, where $j \in \{E, A\} \backslash \{i\}$.

Definition 6.1.3. Let S be a nonempty set of states, let Φ be a nonempty set of *atomic propositions* and let Γ be a nonempty set of *atomic game symbols*. Then a *game model* over S, Φ and Γ is a triple $(S, \{(\rho_g^E, \rho_g^A) : g \in \Gamma\}, V)$, where (ρ_g^E, ρ_g^A) is a game over S for all $g \in \Gamma$ and where V is a valutation function associating each $p \in \Phi$ to a subset $V(p) \subseteq S$.

The language of Dynamic Game Logic, as we already mentioned, consists of *game terms*, built up from atomic games, and of *formulas*, built up from atomic proposition. The connection between these two parts of the language is given by the *test* operation $\phi?$, which turns any formula ϕ into a test game, and the *diamond* operation, which combines a game term γ and a formula ϕ into a new

[1] The main difference between this version and the one of Parikh's original paper lies in the absence of the iteration operator γ^* from our formalism. In this, we follow [69, 71].

formula $\langle \gamma, i \rangle \phi$ which asserts that agent i can guarantee that the game γ will end in a state satisfying ϕ.

Definition 6.1.4. Let Φ be a nonempty set of *atomic propositions* and let Γ be a nonempty set of *atomic game formulas*. Then the sets of all game terms γ and formulas ϕ are defined as

$$\begin{array}{rcl} \gamma & ::= & g \mid \phi? \mid \gamma;\gamma \mid \gamma \cup \gamma \mid \gamma^d \\ \phi & ::= & \bot \mid p \mid \neg\phi \mid \phi \vee \phi \mid \langle \gamma, i \rangle \phi \end{array}$$

for p ranging over Φ, g ranging over Γ, and i ranging over $\{E, A\}$.

We already mentioned the intended interpretations of the test connective $\phi?$ and of the diamond connective $\langle \gamma, i \rangle \phi$. The interpretations of the other game connectives should be clear: γ^d is obtained by swapping the roles of the players in γ, $\gamma_1 \cup \gamma_2$ is a game in which the existential player E chooses whether to play γ_1 or γ_2, and $\gamma_1;\gamma_2$ is the *concatenation* of the two games corresponding to γ_1 and γ_2 respectively.

Definition 6.1.5. Let $G = (S, \{(\rho_g^E, \rho_g^A) : g \in \Gamma, V)$ be a game model over S, Γ and Φ. Then for all game terms γ and all formulas ϕ of Dynamic Game Logic over Γ and Φ we define a game $\|\gamma\|_G$ and a set $\|\phi\|_G \subseteq S$ as follows:

DGL-atomic-game: For all $g \in G$, $\|g\|_G = (\rho_g^E, \rho_g^A)$;

DGL-test: For all formulas ϕ, $\|\phi?\|_G = (\rho^E, \rho^A)$, where

- $s\rho^E X$ iff $s \in \|\phi\|_G$ and $s \in X$;

- $s\rho^A X$ iff $s \notin \|\phi\|_G$ or $s \in X$

for all $s \in S$ and all X with $\emptyset \neq X \subseteq S$;

DGL-concat: For all game terms γ_1 and γ_2, $\|\gamma_1;\gamma_2\|_G = (\rho^E, \rho^A)$, where, for all $i \in \{E, A\}$ and for $\|\gamma_1\|_G = (\rho_1^E, \rho_1^A)$, $\|\gamma_2\|_G = (\rho_2^E, \rho_2^A)$,

- $s\rho^i X$ if and only if there exists a Z such that $s\rho_1^i Z$ and for each $z \in Z$ there exists a set X_z satisfying $z\rho_2^i X_1$ such that

$$X = \bigcup_{z \in Z} X_z;$$

DGL-$\cup$: For all game terms γ_1 and γ_2, $\|\gamma_1 \cup \gamma_2\|_G = (\rho^E, \rho^A)$, where

- $s\rho^E X$ if and only if $s\rho_1^E X$ or $s\rho_2^E X$, and
- $s\rho^A X$ if and only if $s\rho_1^A X$ and $s\rho_2^A X$

where, as before, $\|\gamma_1\|_G = (\rho_1^E, \rho_1^A)$ and $\|\gamma_2\|_G = (\rho_2^E, \rho_2^A)$;[2]

DGL-dual: If $\|\gamma\|_G = (\rho^E, \rho^A)$ then $\|\gamma^d\|_G = (\rho^A, \rho^E)$;

DGL-$\bot$: $\|\bot\|_G = \emptyset$;

DGL-atomic-pr: $\|p\|_G = V(p)$;

DGL-$\neg$: $\|\neg\phi\|_G = S \backslash \|\phi\|_G$;

DGL-$\vee$: $\|\phi_1 \vee \phi_2\|_G = \|\phi_1\|_G \cup \|\phi_2\|_G$;

DGL-$\diamond$: If $\|\gamma\|_G = (\rho^E, \rho^A)$ then for all ϕ,

$$\|\langle\gamma, i\rangle\phi\|_G = \{s \in S : \exists X_s \subseteq \|\phi\|_G \text{ s.t. } s\rho^i X_s\}.$$

If $s \in \|\phi\|_G$, we say that ϕ is *satisfied* by s in G and we write $M \models_s \phi$.

We will not discuss here the properties of this logic, or the vast amount of variants and extensions of it which have been developed and studied. It is worth pointing out, however, that [71] introduced a *Concurrent Dynamic Game Logic* that can be considered one of the main sources of inspiration for the Transition Logic that we will develop in Subsection 6.2.3.

6.1.2 The Representation Theorem

In this subsection, we will briefly recall a remarkable result from [69] which establishes a connection between Dynamic Game Logic and First-Order Logic.

In brief, as the following two theorems demonstrate, either of these logics can be seen as a special case of the other, in the sense that models and formulas of the one can be uniformly translated into models the other in a way which preserves satisfiability and truth:

Theorem 6.1.6. *Let $G = (S, \{(\rho_g^E, \rho_g^A) : g \in \Gamma\}, V)$ be any game model, let ϕ be any game formula for the same language, and let $s \in S$. Then it is possible*

[2] [71] gives the following alternative condition for the powers of the universal player:

- $s\rho^A X$ if and only if $X = Z_1 \cup Z_2$ for two Z_1 and Z_2 such that $s\rho_1^A Z_1$ and $s\rho_2^A Z_2$.

It is trivial to see that, if our games satisfy the monotonicity condition, this rules is equivalent to the one we presented.

to uniformly construct a first-order model G^{FO}, a first-order formula ϕ^{FO} and an assignment s^{FO} of G^{FO} such that

$$G \models_s \phi \Leftrightarrow G^{FO} \models_{s^{FO}} \phi^{FO}.$$

Theorem 6.1.7. *Let M be any first order model, let ϕ be any first-order formula for the signature of M, and let s be an assignment of M. Then it is possible to uniformly construct a game model G^{DGL}, a game formula ϕ^{DGL} and a state s^{DGL} such that*

$$M \models_s \phi \Leftrightarrow G^{DGL} \models_{s^{DGL}} \phi^{DGL}.$$

We will not discuss here the proofs of these two results. Their *significance*, however, is something about which is necessary to spend a few words. In brief, what this back-and-forth representation between First Order Logic and Dynamic Game Logic tells us is that it is possible to understand First Order Logic as a *logic for reasoning about determined games*!

In the next sections, we will attempt to develop a similar result for the case of Dependence Logic.

6.2 Transition Logic

6.2.1 A Logic for Imperfect Information Games Against Nature

We will now define a variant of Dynamic Game Logic, which we will call *Transition Logic*. It deviates from the basic framework of Dynamic Game Logic in two fundamental ways:

1. It considers *one-player* games against Nature, instead of *two-player games* as is usual in Dynamic Game Logic;

2. It allows for *uncertainty* about the initial position of the game.

Hence, Transition Logic can be seen as a *decision-theoretic logic*, rather than a *game-theoretic* one: Transition Logic formulas, as we will see, correspond to assertions about the abilities of a single agent acting under uncertainty, instead of assertions about the abilities of agents interacting which each other.

In principle, it is certainly possible to generalize the approach discussed here to multiple agents acting in situations of imperfect information, and doing so might cause interesting phenomena to surface; but for the time being, we will content ourselves with developing this formalism and proving the analogue of van Benthem's above mentioned results.

Our first definition is a fairly straightforward generalization of the concept of forcing relation:

Definition 6.2.1. Let S be a nonempty set of *states*. A *transition system* over S is a nonempty relation $\theta \subseteq \texttt{Parts}(S) \times \texttt{Parts}(S)$ satisfying the following requirements:

Downwards Closure: If $(X, Y) \in \theta$ and $X' \subseteq X$ then $(X', Y) \in \theta$;

Monotonicity: If $(X, Y) \in \theta$ and $Y \subseteq Y'$ then $(X, Y') \in \theta$;

Non-creation: $(\emptyset, Y) \in \theta$ for all $Y \subseteq S$;

Non-triviality: If $X \neq \emptyset$ then $(X, \emptyset) \notin \theta$.

Informally speaking, a transition system specifies the abilities of an agent: for all $X, Y \subseteq S$ such that $(X, Y) \in \theta$, the agent has a strategy which guarantees that the output of the transition will be in Y whenever the input of the transition is in X.

The four axioms which we gave capture precisely this intended meaning, as we will see:

Definition 6.2.2. A *decision game* is a triple $\Gamma = (S, E, O)$, where S is a nonempty set of *states*, E is a nonempty set of *strategies* and O is an *outcome function* from $S \times E$ to $\texttt{Parts}(S)$.

If $s' \in O(s, e)$, we say that s' is a *possible outcome* of s under e; if $O(s, e) = \emptyset$, we say that e *fails* on input s.

Definition 6.2.3. Let $\Gamma = (S, E, O)$ be a decision game, and let $X, Y \subseteq S$. Then we say that Γ *allows* the transition $X \to Y$, and we write $\Gamma : X \to Y$, if and only if there exists a $e \in E$ such that $\emptyset \neq O(s, e) \subseteq Y$ for all $s \in X$.[3]

Theorem 6.2.4 (Transition Systems and Abilities)**.** *A set $\theta \subseteq \texttt{Parts}(S) \times \texttt{Parts}(S)$ is a transition system if and only if there exists a decision game $\Gamma = (S, E, O)$ such that*

$$(X, Y) \in \theta \Leftrightarrow \Gamma : X \to Y.$$

[3] That is, if and only if our agent has a strategy which guarantees that the outcome will be in Y whenever the input is in X.

Proof. Let $\theta = \{(X_i, Y_i) : i \in I)\}$ be a transition system, and let $\Gamma = (S, I, O)$ for

$$O(s,i) = \begin{cases} Y_i & \text{if } s \in X_i; \\ \emptyset & \text{otherwise.} \end{cases}$$

Suppose that $(X,Y) \in \theta$. If $X = \emptyset$, then $\Gamma : X \to Y$ follows at once by definition. If instead $X \neq \emptyset$, by **non-triviality** we have that Y is nonempty too, and furthermore $(X,Y) = (X_i, Y_i)$ for some $i \in I$. Then $O(s,i) = Y_i \neq \emptyset$ for all $s \in X_i$, as required.

Now suppose that $\Gamma : X \to Y$. Then there exists a $i \in I$ such that $\emptyset \neq O(s,i) \subseteq Y$ for all $s \in X$. If $X \neq \emptyset$, this implies that $X \subseteq X_i$ and $Y_i \subseteq Y$. Hence, by **monotonicity** and **downwards closure**, $(X,Y) \in \theta$, as required. If instead $X = \emptyset$, then by **non-creation** we have again that $(X,Y) \in \theta$.

Conversely, consider a decision game $\Gamma = (S, E, O)$. Then the set of its abilities satisfies our four axioms:

Downwards Closure: Suppose that $\Gamma : X \to Y$ and that $X' \subseteq X$. By definition, there exists a $e \in E$ such that $\emptyset \neq O(s,e) \subseteq Y$ for all $s \in X$. But then the same holds for all $s \in X'$, and hence $\Gamma : X' \to Y$.

Monotonicity: Suppose that $\Gamma : X \to Y$ and that $Y \subseteq Y'$. By definition, there exists a $e \in E$ such that $\emptyset \neq O(s,e) \subseteq Y$ for all $s \in X$. But then, for all such s, $O(s,e) \subseteq Y'$ too, and hence $\Gamma : X \to Y'$.

Non-creation: Let $Y \subseteq S$ and let $e \in E$ be any strategy. Then trivially $\emptyset \neq O(s,e) \subseteq Y$ for all $s \in \emptyset$, and hence $\Gamma : \emptyset \to Y$.

Non-triviality: Let $s_0 \in X$, and suppose that $\Gamma : X \to Y$. Then there exists a e such that $\emptyset \neq O(s,e) \subseteq Y$ for all $s \in X$, and hence in particular $\emptyset \neq O(s_0, e) \subseteq Y$. Therefore, Y is nonempty.

□

Definition 6.2.5. Let S be a nonempty set of states. A *trump* over S is a nonempty, downwards closed family of subsets of S.

Whereas a transition system describes the abilities of an agent to transition from a set of possible initial states to a set of possible terminal states, a trump describes the agent's abilities to reach *some* terminal state from a set of possible initial states:[4]

[4]The term "trump" is taken from [42], where it is used to describe the set of all teams which satisfy a given formula.

Proposition 6.2.6. *Let θ be a transition system and let $Y \subseteq S \neq \emptyset$. Then* $\texttt{reach}(\theta, Y) = \{X \mid (X, Y) \in \theta\}$ *forms a trump. Conversely, for any trump $\mathcal{X}$ over S there exists a transition system θ such that $\mathcal{X} = \texttt{reach}(\theta, Y)$ for any nonempty $Y \subseteq S$.*

Proof. Let θ be a transition system. Then if $(X, Y) \in \theta$ and $X' \subseteq X$, by downwards closure we have at once that $(X', Y) \in \theta$. Furthermore, $(\emptyset, Y) \in \theta$ for any Y. Hence, $\texttt{reach}(\theta, Y)$ is a trump, as required.

Conversely, let $\mathcal{X} = \{X_i : i \in I\}$ be a trump. Then define θ as

$$\theta = \{(A, B) : \emptyset \neq B \subseteq S, \exists i \in I \text{ s.t. } A \subseteq X_i\} \cup \{(\emptyset, \emptyset)\}$$

It is easy to see that θ is a transition system; and by construction, for $Y \neq \emptyset$ we have that $(A, Y) \in \theta \Leftrightarrow \exists i$ s.t. $A \subseteq X_i \Leftrightarrow A \in \mathcal{X}$, where we used the fact that $\mathcal{X}$ is downwards closed. □

We can now define the syntax and semantics of Transition Logic:

Definition 6.2.7. Let Φ be a set of *atomic propositional symbols* and let Θ be a set of *atomic transition symbols*. Then a *transition model* is a tuple $T = (S, \{\theta_t : t \in \Theta\}, V)$, where S is a nonempty set of states, θ_t is a transition system over S for any $t \in \Theta$, and V is a function sending each $p \in \Phi$ into a trump of S.

Definition 6.2.8. Let Φ be a set of atomic propositions and let Θ be a set of atomic transitions. Then the *transition terms* and *formulas* of our language are defined respectively as

$$\begin{array}{rcl} \tau & ::= & t \mid \phi? \mid \tau \otimes \tau \mid \tau \cap \tau \mid \tau ; \tau \\ \phi & ::= & \top \mid p \mid \phi \vee \phi \mid \phi \wedge \phi \mid \langle \tau \rangle \phi \end{array}$$

where t ranges over Θ and p ranges over Φ.

Definition 6.2.9. Let $T = (S, \{\theta_t : t \in \Theta), V)$ be a transition model, let τ be a transition term, and let $X, Y \subseteq S$. Then we say that τ *allows* the transition from X to Y, and we write $T \models_{X \to Y} \tau$, if and only if

TL-atomic-tr: $\tau = t$ for some $t \in \Theta$ and $(X, Y) \in \theta_t$;

TL-test: $\tau = \phi?$ for some transition formula ϕ such that $T \models_X \phi$, and $X \subseteq Y$;

TL-$\otimes$: $\tau = \tau_1 \otimes \tau_2$, and $X = X_1 \cup X_2$ for two X_1 and X_2 such that $T \models_{X_1 \to Y} \tau_1$ and $T \models_{X_2 \to Y} \tau_2$;

TL-$\cap$: $\tau = \tau_1 \cap \tau_2$, $T \models_{X \to Y} \tau_1$ and $T \models_{X \to Y} \tau_2$;

TL-concat: $\tau = \tau_1; \tau_2$ and there exists a $Z \subseteq S$ such that $T \models_{X \to Z} \tau_1$ and $T \models_{Z \to Y} \tau_2$.

Analogously, let ϕ be a transition formula, and let $X \subseteq S$. Then we say that X *satisfies* ϕ, and we write $T \models_X \phi$, if and only if

TL-$\top$: $\phi = \top$;

TL-atomic-pr: $\phi = p$ for some $p \in \Phi$ and $X \in V(p)$;

TL-$\vee$: $\phi = \psi_1 \vee \psi_2$ and $T \models_X \psi_1$ or $T \models_X \psi_2$;

TL-$\wedge$: $\phi = \psi_1 \wedge \psi_2$, $T \models_X \psi_1$ and $T \models_X \psi_2$;

TL-$\diamond$: $\phi = \langle \tau \rangle \psi$ and there exists a Y such that $T \models_{X \to Y} \tau$ and $T \models_Y \psi$.

Proposition 6.2.10. *For any transition model T, transition term τ and transition formula ϕ, the set*

$$\|\tau\|_T = \{(X, Y) : T \models_{X \to Y} \tau\}$$

is a transition system and the set

$$\|\phi\|_T = \{X : T \models_X \phi\}$$

is a trump.

Proof. By induction. □

We end this subsection with a few simple observations about this logic.

First of all, we did not take the negation as one of the primitive connectives. Indeed, Transition Logic, much like Dependence Logic, has an intrinsically *existential* character: it can be used to reason about which sets of possible states an agent *may* reach, but not to reason about which ones such an agent *must* reach. There is of course no reason, in principle, why a negation could not be added to the language, just as there is no reason why a negation cannot be added to Dependence Logic, thus obtaining the far more powerful *Team Logic* [66, 49]: however, this possible extension will not be studied in this work.

The connectives of Transition Logic are, for the most part, very similar to those of Dynamic Game Logic, and their interpretation should pose no difficulties. The exception is the *tensor operator* $\tau_1 \otimes \tau_2$, which replaces the game union operator $\gamma_1 \cup \gamma_2$ and which, while sharing roughly the same informal meaning, behaves in a very different way from the semantic point of view (for example, it is not in general idempotent!)

The decision game corresponding to $\tau_1 \otimes \tau_2$ can be described as follows: first the agent chooses an index $i \in \{1, 2\}$, then he or she picks a strategy for τ_i and plays accordingly. However, the choice of i may be a function of the initial state: hence, the agent can guarantee that the output state will be in Y whenever the input state is in X only if he or she can split X into two subsets X_1 and X_2 and guarantee that the state in Y will be reached from any state in X_1 when τ_1 is played, and from any state in X_2 when τ_2 is played.

It is also of course possible to introduce a "true" choice operator $\tau_1 \cup \tau_2$, with semantical condition

TL-$\cup$: $T \models_{X \to Y} \tau_1 \cup \tau_2$ iff $T \models_{X \to Y} \tau_1$ or $T \models_{X \to Y} \tau_2$;

but we will not explore this possibility any further in this work, nor we will consider any other possible connectives such as, for example, the iteration operator

TL-$*$: $T \models_{X \to Y} \tau^*$ iff there exist $n \in \mathbb{N}$ and $Z_0 \ldots Z_n$ such that $Z_0 = X$, $Z_n = Y$ and $T \models_{Z_i \to Z_{i+1}} \tau$ for all $i \in 1 \ldots n-1$.

6.2.2 A Representation Theorem for Dependence Logic

This subsection contains the central result of this chapter, that is, an analogue of van Benthem's results (Theorems 6.1.6 and 6.1.7 here) for Dependence Logic and Transition Logic.

Representing Dependence Logic models and formulas in Transition Logic is fairly simple:

Definition 6.2.11. Let M be a first-order model. Then $M^{TL} = (S, \{\theta_{\exists v}, \theta_{\forall v} : v \in \texttt{Var}\}, V)$ is the transition model such that

- S is the set of all teams over M;
- For any variable v, $\theta_{\exists v} = \{(X, Y) : \exists F \text{ s.t. } X[F/v] \subseteq Y\}$ and $\theta_{\forall v} = \{(X, Y) : X[M/v] \subseteq Y\}$;
- For any first-order literal or dependence atom α, $V(\alpha) = \{X : M \models_X \phi\}$.

Definition 6.2.12. Let ϕ be a Dependence Logic formula. Then ϕ^{TL} is the transition term defined as follows:

1. If ϕ is a literal or a dependence atom, $\phi^{TL} = \phi?$;

2. If $\phi = \psi_1 \vee \psi_2$, $\phi^{TL} = (\psi_1)^{TL} \otimes (\psi_2)^{TL}$;

3. If $\phi = \psi_1 \wedge \psi_2$, $\phi^{TL} = (\psi_1)^{TL} \wedge (\psi_2)^{TL}$;

4. If $\phi = \exists v \psi$, $\phi^{TL} = \exists v; (\psi)^{TL}$;

5. If $\phi = \forall v \psi$, $\phi^{TL} = \forall v; (\psi)^{TL}$.

Theorem 6.2.13. *For all first-order models M, teams X and formulas ϕ, the following are equivalent:*

- $M \models_X \phi$;
- $\exists Y$ *s.t.* $M^{TL} \models_{X \to Y} \phi^{TL}$;
- $M^{TL} \models_X \langle \phi^{TL} \rangle \top$;
- $M^{TL} \models_{X \to S} \phi^{TL}$.

Proof. We show, by structural induction on ϕ, that the first condition is equivalent to the last one. The equivalences between the last one and the second and third ones are then trivial.

1. If ϕ is a literal or a dependence atom, $M^{TL} \models_{X \to S} \phi?$ if and only if $X \in V(\phi)$, that is, if and only if $M \models_X \phi$;

2. $M^{TL} \models_{X \to S} (\psi_1)^{TL} \otimes (\psi_2)^{TL}$ if and only if $X = X_1 \cup X_2$ for two $X_1, X_2 \subseteq S$ such that $M^{TL} \models_{X_1 \to S} (\psi_1)^{TL}$ and $M^{TL} \models_{X_2 \to S} (\psi_2)^{TL}$. By induction hypothesis, this can be the case if and only if $M \models_{X_1} \psi_1$ and $M \models_{X_2} \psi_2$, that is, if and only if $M \models_X \psi_1 \vee \psi_2$.

3. $M^{TL} \models_{X \to S} (\psi_1)^{TL} \wedge (\psi_2)^{TL}$ if and only if $M^{TL} \models_{X \to S} (\psi_1)^{TL}$ and $M^{TL} \models_{X \to S} (\psi_2)^{TL}$, that is, by induction hypothesis, if and only if $M \models_X \psi_1 \wedge \psi_2$.

4. $M^{TL} \models_{X \to S} \exists v; (\psi)^{TL}$ if and only if there exists a Y such that $Y \supseteq X[F/v]$ for some F and $M^{TL} \models_{Y \to S} \psi$. By induction hypothesis and downwards closure, this can be the case if and only if $M \models_{X[F/v]} \psi$ for some F, that is, if and only if $M \models_X \exists v \psi$;

5. $M^{TL} \models_{X \to S} \forall v; (\psi)^{TL}$ if and only if $M^{TL} \models_{Y \to S} (\psi)^{TL}$ for some $Y \supseteq X[M/v]$, that is, if and only if $M \models_{X[M/v]} \psi$, that is, if and only if $M \models_X \forall v \psi$.

□

Representing Transition Models, game terms and formulas in Dependence Logic is somewhat more complex:

Definition 6.2.14. Let $T = (S, (\theta_t : t \in \Theta), V)$ be a transition model. Furthermore, for any $t \in \Theta$, let $\theta_t = \{(X_i, Y_i) : i \in I_t\}$, and, for any $p \in \Phi$, let $V(p) = \{X_j : j \in J_p\}$. Then T^{DL} is the first-order model with domain[5] $S \uplus \biguplus\{I_t : t \in \Theta\} \uplus \biguplus\{J_p : p \in \Phi\}$ whose signature contains

- For every $t \in \Theta$, a ternary relation R_t whose interpretation is $\{(i, x, y) : i \in I_t, x \in X_i, y \in Y_i\}$;
- For every $p \in \Phi$, a binary relation V_p whose interpretation is $\{(j, x) : j \in J_p, x \in X_j\}$.

Definition 6.2.15. For any formula ϕ, transition term τ, variable x and unary relation symbol P the Dependence Logic formulas ϕ_x^{DL} and $\tau_x^{DL}(P)$ are defined as follows:

1. $\top_x^{DL}$ is $\top$;
2. For all $p \in \Phi$, p_x^{DL} is $\exists j(=(j) \wedge V_p(j, x))$;
3. $(\psi_1 \vee \psi_2)_x^{DL}$ is $(\psi_1)_x^{DL} \sqcup (\psi_2)^{DL}$, where $\sqcup$ is the classical disjunction introduced in Definition 2.2.4;
4. $(\psi_1 \wedge \psi_2)_x^{DL}$ is $(\psi_1)_x^{DL} \wedge (\psi_2)_x^{DL}$;
5. $(\langle\tau\rangle\psi)_x^{DL}$ is $\exists P((\tau)_x^{DL}(P) \wedge \forall y(\neg Py \vee (\psi)_y^{DL}))$, where the second-order existential quantifier is a shorthand for the construction described in Definition 2.2.16 and y is a new and unused variable;

1. For all $t \in \Theta$, $t_x^{DL}(P)$ is $\exists i(=(i) \wedge \exists y(R_t(i, x, y)) \wedge \forall y(\neg R_t(i, x, y) \vee Py))$;
2. For all formulas ϕ, $(\phi?)_x^{DL}(P)$ is $\phi_x^{DL} \wedge Px$;
3. $(\tau_1 \otimes \tau_2)_x^{DL}(P) = (\tau_1)_x^{DL}(P) \vee (\tau_2)_x^{DL}(P)$;
4. $(\tau_1 \cap \tau_2)_x^{DL}(P) = (\tau_1)_x^{DL}(P) \wedge (\tau_2)_x^{DL}(P)$;
5. $(\tau_1; \tau_2)_x^{DL}(P) = \exists Q((\tau_1)_x^{DL}(Q) \wedge \forall y(\neg Qy \vee (\tau_2)_y^{DL}(P)))$, where y a new and unused variable.

[5]Here we write $A \uplus B$ for the *disjoint union* of the sets A and B.

Theorem 6.2.16. *For all transition models $T = (S, (\theta_t : t \in \Theta), V)$, transition terms τ, transition formulas ϕ, variables x, sets $P \subseteq S$ and teams X over T^{DL},*

$$T^{DL} \models_X \phi_x^{DL} \Leftrightarrow T \models_{X(x)} \phi$$

and

$$T^{DL} \models_X \tau_x^{DL}(P) \Leftrightarrow T \models_{X(x) \to P} \tau.$$

Proof. The proof is by structural induction on terms and formulas.

Let us first consider the cases corresponding to formulas:

1. For all teams X, $T^{DL} \models_X \top$ and $T \models_{X(x)} \top$, as required;

2. Suppose that $T^{DL} \models_X \exists j(=(j) \wedge V_p(j, x))$. Then there exists a $m \in \mathtt{Dom}(T^{DL})$ such that $T^{DL} \models_{X[m/j]} V_p(j, x)$. Hence, we have that $X(x) \subseteq X_m \in V(p)$; and, by downwards closure, this implies that $X(x) \in V(p)$, and hence that $T \models_{X(x)} p$ as required.

   Conversely, suppose that $T \models_{X(x)} p$. Then $X(x) \in V(p)$, and hence $X(x) = X_m$ for some $m \in J_p$. Then we have by definition that $T^{DL} \models_{X[m/j]} V_p(j, x)$, and finally that $T^{DL} \models_X T_x(p)$.

3. By Proposition 2.2.5, $T^{DL} \models_X (\psi_1 \vee \psi_2)_x^{DL}$ if and only if $T^{DL} \models_X (\psi_1)_x^{DL}$ or $T^{DL} \models_X (\psi_2)_x^{DL}$. By induction hypothesis, this is the case if and only if $T \models_{X(x)} \psi_1$ or $T \models_{X(x)} \psi_2$, that is, if and only if $T \models_{X(x)} \psi_1 \vee \psi_2$.

4. $T^{DL} \models_X (\psi_1 \wedge \psi_2)_x^{DL}$ if and only if $T^{DL} \models_X (\psi_1)_x^{DL}$ and $T^{DL} \models_x (\psi_2)_x^{DL}$, that is, by induction hypothesis, if and only if $T \models_X \psi_1 \wedge \psi_2$.

5. $T^{DL} \models_X (\langle \tau \rangle \psi)_x^{DL}$ if and only if there exists a P such that $T^{DL} \models_X (\tau)_x^{DL}(P)$ and $T^{DL} \models_{X[T^{DL}/y]} \neg Py \vee (\psi)_y^{DL}$. By induction hypothesis, the first condition holds if and only if $T \models_{X(x) \to P} \tau$. As for the second one, it holds if and only if $X[T^{DL}/y] = Y_1 \cup Y_2$ for two Y_1, Y_2 such that $T^{DL} \models_{Y_1} \neg Py$ and $T^{DL} \models_{Y_2} \tau_y(\psi)$. But then we must have that $T \models_{Y_2(y)} \psi$ and that $P \subseteq Y_2(y)$; therefore, by downwards closure, $T \models_P \psi$ and finally $T \models_{X(x)} \langle \tau \rangle \psi$.

   Conversely, suppose that there exists a P such that $T \models_{X(x) \to P} \tau$ and $T \models_P \psi$; then by induction hypothesis we have that $T^{DL} \models_X (\tau)_x^{DL}(P)$ and that $T^{DL} \models_{X[T^{DL}/y]} \neg Py \vee (\psi)_x^{DL}$, and hence $T^{DL} \models_X (\langle \tau \rangle \psi)_x^{DL}$.

Now let us consider the cases corresponding to transition terms:

1. Suppose that $T^{DL} \models_X \exists i(=(i) \wedge \exists y(R_t(i,x,y)) \wedge \forall y(\neg R_t(i,x,y) \vee Py))$. If $X = \emptyset$ then $X(x) = \emptyset$, and hence by **non-creation** we have that $(X(x), P) = (\emptyset, P) \in \theta_t$, as required.

   Let us assume instead that $X \neq \emptyset$. Then, by hypothesis, there exists a $m \in \mathtt{Dom}(T^{DL})$ such that

   - There exists a F such that $T^{DL} \models_{X[m/i][F/y]} R_t(i,x,y)$;
   - $T^{DL} \models_{X[m/i][T^{DL}/y]} \neg R_t(i,x,y) \vee Py$.

   From the first condition it follows that for every $p \in X(x)$ there exists a q such that $R_t(m,p,q)$: therefore, by the definition of R_t, every such p must be in X_m.

   From the second condition it follows that whenever $R_t(m,p,q)$ and $p \in X(x) \subseteq X_m$, $q \in P$; and, since $X(x) \neq \emptyset$, this implies that $Y_m \subseteq P$ by the definition of R_t.

   Hence, by **monotonicity** and **downwards closure**, $(X(x), P) \in \theta_t$ and $T \models_{X(x) \to P} t$, as required.

   Conversely, suppose that $(X(x), P) = (X_m, Y_m) \in \theta_t$ for some $m \in I_t$. If $X(x) = \emptyset$ then $X = \emptyset$, and hence by Proposition 2.2.6 we have that $T^{DL} \models_X t_x^{DL}(P)$, as required. Otherwise, by **non-triviality**, $P = Y_m \neq \emptyset$. Let now $p \in P$ be any of its elements and let $F(s) = p$ for all $p \in X[m/i]$: then $M \models_{X[m/i][F/y]} R_t(i,x,y)$, as any assignment of this team sends x to some element of X_m and y to $p \in Y_m$. Furthermore, let $s \in X(x) = X_m$, and let q be such that $R_t(m, s(x), q)$: then $q \in Y_m = P$, and hence $M \models_{X[m/i][T^{DL}/y]} \neg R_t(i,x,y) \vee Py$. So, in conclusion, $M \models_X t_x^{DL}(P)$, as required.

2. $T^{DL} \models_X \phi_x^{DL} \wedge Px$ if and only if $T \models_{X(x)} \phi$ and $X(x) \subseteq P$, that is, if and only if $T \models_{X(x) \to P} \phi?$.

3. $T^{DL} \models_X (\tau_1)_x^{DL}(P) \vee (\tau_2)_x^{DL}(P)$ if and only if $X = X_1 \cup X_2$ for two X_1, X_2 such that

   - $X = X_1 \cup X_2$, and therefore $X(x) = X_1(x) \cup X_2(x)$;
   - $T^{DL} \models_{X_1} (\tau_1)_x^{DL}(P)$, that is, by induction hypothesis, $T \models_{X_1(x) \to P} \tau_1$;
   - $T^{DL} \models_{X_2} (\tau_2)_x^{DL}(P)$, that is, by induction hypothesis, $T \models_{X_2(x) \to P} \tau_2$;

Hence, if $T^{DL} \models_X (\tau_1 \otimes \tau_2)^{DL}_x(P)$ then $T \models_{X(x)\to P} \tau_1 \otimes \tau_2$.

Conversely, if $X(x) = A \cup B$ for two A, B such that $T \models_{A\to P} \tau_1$ and $T \models_{B\to P} \tau_2$, let

$$\begin{aligned} X_1 &= \{s \in X : s(x) \in A\} \\ X_2 &= \{s \in X : s(x) \in B\}. \end{aligned}$$

Clearly $X = X_1 \cup X_2$, and furthermore by induction hypothesis $T^{DL} \models_{X_1} (\tau_1)^{DL}_x(P)$ and $T^{DL} \models_{X_2} (\tau_2)^{DL}_x(P)$. Hence, $T^{DL} \models_X (\tau_1 \otimes \tau_2)^{DL}_x(P)$, as required.

4. $T^{DL} \models_X (\tau_1 \cap \tau_2)^{DL}_x(P)$ if and only if $T^{DL} \models_X (\tau_1)^{DL}_x(P)$ and $T^{DL} \models_X (\tau_2)^{DL}_x(P)$, that is, by induction hypothesis, if and only if $T \models_{X(x)\to P} \tau_1 \cap \tau_2$.

5. $T^{DL} \models_X \exists Q((\tau_1)^{DL}_x(Q) \wedge \forall y(\neg Qy \vee (\tau_2)^{DL}_y(P)))$ if and only if there exists a Q such that $T \models_{X(x)\to Q} \tau_1$ and there exists a $Q' \supseteq Q$ such that $T \models_{Q'\to P} \tau_2$. By downwards closure, if this is the case then $T \models_{Q\to P} \tau_2$ too, and hence $T \models_{X(x)\to P} \tau_1; \tau_2$, as required.

   Conversely, suppose that there exists a Q such that $T \models_{X(x)\to Q} \tau_1$ and $T \models_{Q\to P} \tau_2$. Then, by induction hypothesis $T^{DL} \models_X (\tau_1)^{DL}_x(Q)$; and furthermore, $X[T^{DL}/y]$ can be split into

$$Z_1 = \{s \in X[T^{DL}/y] : s(y) \not\in Q\}$$

   and

$$Z_2 = \{s \in X[T^{DL}/y] : s(y) \in Q\}$$

   It is trivial to see that $T^{DL} \models_{Z_1} \neg Qy$; and furthermore, since $Z_2(y) = Q$ and $T \models_{Q\to P} \tau_2$, by induction hypothesis we have that $T^{DL} \models_{Z_2} (\tau_2)^{DL}_y$. Thus $T^{DL} \models_{X[T^{DL}/y]} \forall y(\neg Qy \vee (\tau_2)^{DL}_y(P))$ and finally $T^{DL} \models_X (\tau_1; \tau_2)^{DL}_x(P)$, and this concludes the proof.

□

The significance of the results of this subsection is comparable to that of the corresponding ones about First Order Logic which we recalled in Subsection 6.1.2. In brief, what Theorems 6.2.13 and 6.2.16 tell us is that it is possible to understand Dependence Logic as a language for reasoning about *imperfect information decision problems*!

In the rest of this chapter, we will examine how this insight may be used in order to further the study of Dependence Logic and its variants.

6.2.3 Transition Dependence Logic

Just as van Benthem's theorems, which we recalled in Subsection 6.1.2, allows one to reinterpret First Order Logic as a logic of perfect information two-player games, Theorems 6.2.13 and 6.2.16 of Subsection 6.2.2 permit us to understand Dependence Logic as a logic of imperfect-information decision problems.

However, the language of Dependence Logic, in itself, does very little to support this interpretation. We may certainly associate a Dependence Logic sentence such as, for example, $\forall x \exists y (=(y, f(x)) \wedge Pxy)$, to a certain game of imperfect information, and then establish a correspondence between the truth of the sentence and certain abilities of an agent in this game; but this interpretation - legitimate though it may be from a semantic perspective - does not appear to arise entirely naturally from the syntactical structure of the sentence.

But by exploiting of the representation of Dependence Logic inside of Transition Logic of Definitions 6.2.11 and 6.2.12, it is not difficult to define a variant of Dependence Logic in which this interpretation is manifested at the syntactical level itself:

Definition 6.2.17. Let Σ be a first-order signature. Then the sets of all *transition terms* and of all *formulas* of Dependence Transition Logic are given by the rules

$$\begin{array}{rcl} \tau & ::= & \exists v \mid \forall v \mid \phi? \mid \tau \otimes \tau \mid \tau \cap \tau \mid \tau ; \tau \\ \phi & ::= & R\vec{t} \mid \neg R\vec{t} \mid \ =(t_1 \dots t_n) \mid \phi \vee \phi \mid \phi \wedge \phi \mid \langle \tau \rangle \phi. \end{array}$$

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

Definition 6.2.18. Let M be a first-order model, let τ be a first-order transition term of the same signature, and let X and Y be teams over M. Then we say that the transition $X \to Y$ is *allowed* by τ in M, and we write $M \models_{X \to Y} \tau$, if and only if

TDL-$\exists$: τ is of the form $\exists v$ for some $v \in \mathtt{Var}$ and there exists a F such that $X[F/v] \subseteq Y$;

TDL-$\forall$: τ is of the form $\forall v$ for some $v \in \mathtt{Var}$ and $X[M/v] \subseteq Y$;

TDL-test: τ is of the form $\phi?$, $M \models_X \phi$, and $X \subseteq Y$;

TDL-$\otimes$: τ is of the form $\tau_1 \otimes \tau_2$ and $X = X_1 \cup X_2$ for some X_1 and X_2 such that $M \models_{X_1 \to Y} \tau_1$ and $M \models_{X_2 \to Y} \tau_2$;

TDL-$\cap$: τ is of the form $\tau_1 \cap \tau_2$, $M \models_{X \to Y} \tau_1$ and $M \models_{X \to Y} \tau_2$;

TDL-concat: τ is of the form $\tau_1; \tau_2$ and there exists a team Z such that $M \models_{X \to Z} \tau_1$ and $M \models_{Z \to Y} \tau_2$.

Similarly, if ϕ is a formula and X is a team with domain `Var`. Then we say that X *satisfies* ϕ in M, and we write $M \models_X \phi$, if and only if

TDL-lit: ϕ is a first-order literal and $M \models_s \phi$ in the usual first-order sense for all $s \in X$;

TDL-dep: ϕ is a dependence atom $=\!(t_1 \ldots t_n)$ and any two $s, s' \in X$ which assign the same values to $t_1 \ldots t_{n-1}$ also assign the same value to t_n;

TDL-$\vee$: ϕ is of the form $\phi_1 \vee \phi_2$ and $M \models_X \phi_1$ or $M \models_X \phi_2$;

TDL-$\wedge$: ϕ is of the form $\phi_1 \wedge \phi_2$, $M \models_X \phi_1$ and $M \models_X \phi_2$;

TDL-$\diamond$: ϕ is of the form $\langle\tau\rangle\psi$ and there exists a Y such that $M \models_{X \to Y} \tau$ and $M \models_Y \psi$.

It is not difficult to see, on the basis of the results of the previous section, that this new variant of Dependence Logic is equivalent to the usual one:

Theorem 6.2.19. *For every Dependence Logic formula ϕ there exists a Transition Dependence Logic transition term τ_ϕ such that*

$$M \models_X \phi \Leftrightarrow \exists Y \text{ s.t. } M \models_{X \to Y} \tau_\phi \Leftrightarrow M \models_X \langle\tau_\phi\rangle\top$$

for all first-order models M and teams X.

Proof. τ_ϕ is defined by structural induction on ϕ, as follows:

1. If ϕ is a first-order literal or a dependence atom then $\tau_\phi = \phi?$;
2. If ϕ is $\phi_1 \vee \phi_2$ then $\tau_\phi = \tau_{\phi_1} \otimes \tau_{\phi_2}$;
3. If ϕ is $\phi_1 \wedge \phi_2$ then $\tau_\phi = \tau_{\phi_1} \cap \tau_{\phi_2}$;
4. If ϕ is $\exists v \psi$ then $\tau_\phi = \exists v; \tau_\psi$;
5. If ϕ is $\forall v \psi$ then $\tau_\phi = \forall v; \tau_\psi$.

It is then trivial to verify, again by induction on ϕ, that $M \models_X \phi$ if and only if $M \models_X \langle\tau_\phi\rangle\top$, as required. □

Theorem 6.2.20. *For every Transition Dependence Logic formula ϕ there exists a Dependence Logic formula ϕ' such that*

$$M \models_X \phi \Leftrightarrow M \models_X \phi'$$

for all first-order models M and teams X.

Proof. **(Sketch)**
Translate ϕ into Σ_1^1, and then apply Theorem 2.2.14. □

However, in a sense, Transition Dependence Logic allows one to consider subtler distinctions than Dependence Logic does. The formula $\forall x \exists y (=(y, f(x)) \wedge Pxy)$, for example, could be translated as any of

- $\langle \forall x; \exists y \rangle (=(y, f(x)) \wedge Pxy)$;
- $\langle \forall x; \exists y \rangle \langle =(y, f(x))? \rangle Pxy$;
- $\langle \forall x; \exists y \rangle \langle Pxy? \rangle =(y, f(x))$;
- $\langle \forall x; \exists y \rangle \langle (Pxy?) \cap (=(y, f(x))?) \rangle \top$.

The intended interpretations of these formulas are rather different, even though they happen to be satisfied by the same teams: and for this reason, Transition Dependence Logic may be thought of as a *refinement* of Dependence Logic proper, even though it has exactly the same expressive power.

6.3 Dynamic Semantics

6.3.1 Dynamic Predicate Logic

Dynamic Semantics is an approach to the formal semantics of natural language which can be summarized by the following motto, from [34]:

> *The meaning of a sentence does not lie in its truth conditions, but rather in the way it changes (the representation of) the information of the interpreter.*

Whereas truth-theoretic semantics takes as its primary object of investigation the conditions under which a hypothetical listener would be willing to accept a statement as truthful, dynamic semantics takes as its fundamental semantic objects the *informational changes* that the utterance of a sentence has on the *contexts* under which further statements will be interpreted, as well as on the states of mind of hypothetical listeners.

This second approach has some advantages in formal linguistics, for example with respect to the interpretation of anaphora and "Donkey Sentences"; but as this work is not concerned with linguistic applications, and – more importantly – because the author is no linguist, we will not discuss the contribution of dynamic semantics to formal linguistics any further.

One thing worth pointing out, however, is that the dynamic approach and the truth-theoretic approach to semantics are not in competition. Rather, they are complementary: given a dynamic semantics, it is possible to recover the truth conditions by examining under which circumstances the interpretation of the formula leads to a state which accepts it, and, conversely, given a truth-theoretic semantics one can recover the dynamics hidden in it by comparing the truth conditions of an expression with the ones of its components.

We refer to van Benthem's book [68], to Dekker's paper [14] and to van Eijk's summary [72] for a more thorough introduction to this interesting approach to semantics. Here we will just present, as an example of a dynamic semantics, the *Dynamic Predicate Logic* introduced in [34]:

Definition 6.3.1. Let M be a first order model, let ϕ be a first order formula over its signature, and let s and s' be two assignments. Then we say that the transition from s to s' is *allowed* by ϕ in M, and we write $M \models_{s \to s'} \phi$, if and only if

DPL-atom: ϕ is an atomic formula, $s = s'$ and $M \models_s \phi$ in the usual sense;

DPL-$\neg$: ϕ is of the form $\neg\psi$, $s = s'$ and for all assignments h, $M \not\models_{s \to h} \psi$;

DPL-$\wedge$: ϕ is of the form $\psi_1 \wedge \psi_2$ and there exists a h such that $M \models_{s \to h} \psi_1$ and $M \models_{h \to s'} \psi_2$;

DPL-$\vee$: ϕ is of the form $\psi_1 \vee \psi_2$, $s = s'$ and there exists a h such that $M \models_{s \to h} \psi_1$ or $M \models_{s \to h} \psi_2$;

DPL-$\rightarrow$: ϕ is of the form $\psi_1 \rightarrow \psi_2$, $s = s'$ and for all h it holds that

$$M \models_{s \to h} \psi_1 \Rightarrow \exists h' \text{ s.t. } M \models_{h \to h'} \psi_2;$$

DPL-$\exists$: ϕ is of the form $\exists x \psi$ and there exists an element $m \in \mathtt{Dom}(M)$ such that $M \models_{s[m/x] \to s'} \psi$;

DPL-$\forall$: ϕ is of the form $\forall x \psi$, $s = s'$ and for all elements $m \in \mathtt{Dom}(M)$ there exists a h such that $M \models_{s[m/x] \to h} \psi$.

A formula ϕ is *satisfied* by an assignment s if and only if there exists an assignment s' such that $M \models_{s \to s'} \phi$; in this case, we will write $M \models_s \phi$.

We will not examine in any detail this semantics or its applications, as this of little relevance for our purposes here. However, what is worth pointing out is that according to it, formulas are interpreted not as sets of assignments, as in the case of Tarski's semantics for First Order Logic, but rather as sets of *transitions* from assignments to assignments. For example, an atomic formula $P\vec{t}$ is interpreted as a *test*, which allows a transition (s, s) if and only if the assignment s satisfies $P\vec{t}$; and instead, an existential quantification $\exists x\psi$ corresponds to a transition in which first we change the value of the variable x, and then we execute ψ. Of special interest is the conjunction $\psi_1 \wedge \psi_2$, which is interpreted as a *concatenation* of transitions: this, combined with the semantics for existential quantification, makes it so that $(\exists x\psi) \wedge \theta$ is logically equivalent to $\exists x(\psi \wedge \theta)$, differently from the case of standard First Order Logic.[6]

Furthermore, satisfaction in this semantics is a *derived* property, to be understood in terms of *reachability*: an assignment s satisfies a formula ϕ if and only if there exists some s' such that ϕ allows the transition from s to s', that is, if and only if s is not a "dead end" for ϕ.

Even more interestingly, it is not difficult to see that in this approach to semantics, it is possible to interpret the existential quantifier as an *atomic formula*! Indeed, if we define

DPL-∃-atom $M \models_{s \to s'} \exists x$ if and only if there exists a $m \in \texttt{Dom}(M)$ such that $s' = s[m/x]$

then it is easy to verify that $\exists x\phi$ is equivalent to $\exists x \wedge \phi$. In other words, we can isolate the semantic contribution of the existential quantifier itself, much as we did for the case of Transition Dependence Logic!

The same cannot be said, however, for the universal quantifier: it is easy to see that there exists no semantics for $\forall x$ in this framework which makes $\forall x\psi$ equivalent to $\forall x \wedge \psi$. The problem, of course, is that in order to verify the truth of $\forall x\psi$, we need to examine ψ with respect to multiple assignments, while, according to the above rules, a conjunction $\forall x \wedge \psi$ allows a transition

[6] As an aside, the fact that in Dynamic Predicate Logic existential quantifiers can have an effect even beyond their syntactic scope was one of the main reasons why this semantics can be used to interpret natural language statements in which pronouns refer to nouns which lie beyond their apparent scopes, as in the famous example

(A man)$_1$ walks in the park. (He)$_1$ whistles.

We refer to [34] for further details.

from s to s' if and only if there exists *at least one* s'' such that $\forall x$ allows the transition from s to s'' and ψ allows the transition from s'' to s'.

The similarity between this semantics and our semantics for transition terms should be evident. Hence, it seems natural to ask whether we can adopt, for a suitable variant of Dependence Logic, the following variant of Groenendijk and Stokhof's motto:

> *The meaning of a formula does not lie in its satisfaction conditions, but rather in the team transitions it allows.*

From this point of view, *transition terms* are the fundamental objects of our syntax, and formulas can be removed altogether from the language - although, of course, the tests corresponding to literals and dependence formulas should still be available. As in Groenendijk and Stokhof's logic, satisfaction becomes then a derived concept: in brief, a team X can be said to satisfy a term τ if and only if there exists a Y such that τ allows the transition from X to Y, or, in other words, if and only if *some* set of non-losing outcomes can be reached from set the initial positions X in the game corresponding to τ.

In the next section, we will make use of these intuitions to develop another, terser version of Dependence Logic; and finally, in Subsection 6.3.3 we will come full circle by showing how the semantics of this logic can be interpreted in terms of *reachability conditions* in a suitable variant of the Game Theoretic Semantics for standard Dependence Logic.

6.3.2 Dynamic Dependence Logic

We will now develop a formula-free variant of Transition Dependence Logic, along the lines of Groenendijk and Stokhof's Dynamic Predicate Logic.

Apart from Dynamic Predicate Logic, our treatment will be also inspired by Abramsky's Game Semantics for multiagent logics of imperfect information [1, 2]: this will be particularly evident in the next subsection, in which we will develop a Game Theoretic Semantics for our logic.

Definition 6.3.2. Let Σ be a first-order signature. The set of all *transition formulas* of Dynamic Dependence Logic over Σ is given by the rules

$$\tau ::= R\vec{t} \mid \neg R\vec{t} \mid \; =(t_1 \ldots t_n) \mid \exists v \mid \forall v \mid \tau \otimes \tau \mid \tau \cap \tau \mid \tau ; \tau$$

where, as usual, R ranges over all relation symbols of our signature, $\vec{t}$ ranges over all tuples of terms of the required lengths, n ranges over $\mathbb{N}$, $t_1 \ldots t_n$ range over all terms, and v ranges over `Var`.

The semantical rules associated to this language are precisely as one would expect:

Definition 6.3.3. Let M be a first-order model, let τ be a transition formula of Dynamic Dependence Logic over the signature of M, and let X and Y be two teams over M with domain `Var`. Then we say that τ *allows* the transition $X \to Y$ in M, and we write $M \models_{X \to Y} \tau$, if and only if

DDL-lit: τ is a first-order literal, $M \models_s \tau$ in the usual first-order sense for all $s \in X$, and $X \subseteq Y$;

DDL-dep: τ is a dependence atom $=\!(t_1 \ldots t_n)$, $X \subseteq Y$, and any two assignments $s, s' \in X$ which coincide over $t_1 \ldots t_{n-1}$ also coincide over t_n;

DDL-∃: τ is of the form $\exists v$ for some $v \in \texttt{Var}$, and $X[F/v] \subseteq Y$ for some $F : X \to \texttt{Dom}(M)$;

DDL-∀: τ is of the form $\forall v$ for some $v \in \texttt{Var}$, and $X[M/v] \subseteq Y$;

DDL-⊗: τ is of the form $\tau_1 \otimes \tau_2$ and $X = X_1 \cup X_2$ for two teams X_1 and X_2 such that $M \models_{X_1 \to Y} \tau_1$ and $M \models_{X_2 \to Y} \tau_2$;

DDL-∩: τ is of the form $\tau_1 \cap \tau_2$, $M \models_{X \to Y} \tau_1$ and $M \models_{X \to Y} \tau_2$;

DDL-concat: τ is of the form $\tau_1; \tau_2$, and there exists a Z such that $M \models_{X \to Z} \tau_1$ and $M \models_{Z \to Y} \tau_2$.

A formula τ is said to be *satisfied* by a team X in a model M if and only if there exists a Y such that $M \models_{X \to Y} \tau$; and if this is the case, we will write $M \models_X \tau$.

It is not difficult to see that Dynamic Dependence Logic is equivalent to Transition Dependence Logic (and, therefore, to Dependence Logic).

Proposition 6.3.4. *Let ϕ be a Dependence Logic formula. Then there exists a term ϕ' of Dynamic Dependence Logic which is equivalent to it, in the sense that*

$$M \models_X \phi \Leftrightarrow M \models_X \phi' \Leftrightarrow \exists Y \text{ s.t. } M \models_{X \to Y} \phi'$$

for all suitable teams X and models M

Proof. We build ϕ' by structural induction:

1. If ϕ is a literal or a dependence atom then $\phi' = \phi$;
2. If ϕ is $\psi_1 \vee \psi_2$ then $\phi' = \psi_1' \otimes \psi_2'$;

3. If ϕ is $\psi_1 \wedge \psi_2$ then $\phi' = \psi'_1 \cap \psi'_2$;

4. If ϕ is $\exists x\psi$ then $\phi' = \exists x; \psi'$;

5. If ϕ is $\forall x\psi$ then $\phi' = \forall x; \psi'$.

□

Proposition 6.3.5. *Let τ be a Dynamic Dependence Logic term. Then there exists a Transition Dependence Logic term τ' such that*

$$M \models_{X \to Y} \tau \Leftrightarrow M \models_{X \to Y} \tau'$$

for all suitable X, Y and M, and such that hence

$$M \models_X \tau \Leftrightarrow M \models_X \langle \tau' \rangle \top.$$

Proof. Build τ' by structural induction:

1. If τ is a literal or dependence atom then $\tau' = \tau?$;

2. If τ is of the form $\exists v$ or $\forall v$ then $\tau' = \tau$;

3. If τ is of the form $\tau_1 \otimes \tau_2$ then $\tau' = \tau'_1 \otimes \tau'_2$;

4. If τ is of the form $\tau_1 \cap \tau_2$ then $\tau' = \tau'_1 \cap \tau'_2$;

5. If τ is of the form $\tau_1; \tau_2$ then $\tau' = \tau'_1; \tau'_2$.

□

Corollary 6.3.6. *Dynamic Dependence Logic is equivalent to Transition Dependence Logic and to Dependence Logic*

Proof. Follows from the two previous results and from the equivalence between Dependence Logic and Transition Dependence Logic. □

6.3.3 Game Theoretic Semantics for Dynamic Dependence Logic

In this subsection, we will adapt the Game Theoretic Semantics of Subsection 2.2.3 to the case of Dynamic Dependence Logic.

Definition 6.3.7. Let τ be any Dynamic Dependence Logic formula. Then $\texttt{Player}(\tau) \in \{\texttt{E}, \texttt{A}\}$ is defined as follows:

1. If τ is a first-order literal or a dependence atom, $\texttt{Player}(\tau) = \texttt{E}$;

2. If τ is of the form $\tau_1 \otimes \tau_2$ or $\exists v$ then $\texttt{Player}(\tau) = \texttt{E}$;

3. If τ is of the form $\tau_1 \cap \tau_2$ or $\forall v$ then $\texttt{Player}(\tau) = \texttt{A}$;

4. If τ is of the form $\tau_1; \tau_2$ then $\texttt{Player}(\tau) = \texttt{Player}(\tau_1)$.

Positions of our game will be pairs (τ, s), where τ is a transition term and s is an assignment. The *successors* of a given position are defined as follows:

Definition 6.3.8. Let M be a first order model, let τ be a transition term and let s be an assignment over M. Then the set $\texttt{Succ}_M(\tau, s)$ of the *successors* of the position (τ, s) is defined as follows:

1. If τ is a first order literal ϕ then

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

   where λ stands for the empty string;

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

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

4. If τ is of the form $\tau_1 \otimes \tau_2$ or $\tau_1 \cap \tau_2$ then $\texttt{Succ}_M(\tau, s) = \{(\tau_1, s), (\tau_2, s)\}$;

5. If τ is of the form $\tau_1; \tau_2$ then

$$\texttt{Succ}_M(\tau, s) = \{(\tau'; \tau_2, s') : (\tau', s') \in \texttt{Succ}_M(\tau_1, s)\}$$

   where, with an abuse of notation, we assume that $\lambda; \tau_2$ is equal to τ_2.

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

Definition 6.3.9. Let M be a first-order model, let τ be a Dynamic Dependence Logic formula, and let X and Y be teams. Then the game $G^M_{X \to Y}(\tau)$ is defined as follows:

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

Definition 6.3.10. Let $G^M_{X\to Y}(\tau)$ be as in the above definition. Then a *play* of this game is a finite sequence $\vec{p} = p_1 \ldots p_n$ of positions of the game such that

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

If furthermore $\mathtt{Succ}_M(p_n) = \emptyset$, 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. Similarly to the case of Definition 2.2.21, such conditions are made to correspond as *uniformity conditions* over sets of plays:

Definition 6.3.11. Let $G^M_{X\to Y}(\tau)$ 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, such that p_i is of the form[7] $(((=(t_1 \ldots t_n); \tau_1); \ldots \tau_k), s)$ and q_j is of the form $(((=(t_1 \ldots t_n); \tau_1); \ldots \tau_k), s')$ for the same instance of the dependence atom $=(t_1 \ldots t_n)$ it holds that

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

where, with an abuse of notation, we identify $=(t_1 \ldots t_n); \lambda$ with $=(t_1 \ldots t_n)$.

As always, we will only consider *positional strategies*, that is, strategies that depend only on the current position.

Definition 6.3.12. Let $G^M_{X\to Y}(\tau)$ be as above, and let τ' be any expression such that (τ', s') is a possible position of the game for some s'. Then a *local strategy* for τ' is a function $f_{\tau'}$ sending each s' into a $(\tau'', s'') \in \mathtt{Succ}_M(\tau', s')$.

Definition 6.3.13. Let $G^M_{X\to Y}(\tau)$ be as above, let $\vec{p} = p_1 \ldots p_n$ be a play in it, and let $f_{\tau'}$ be a local strategy for some τ'. Then $\vec{p}$ is said to *follow* $f_{\tau'}$ if and only if for all $i \in 1 \ldots n-1$ and all s',

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

Definition 6.3.14. Let $G^M_{X\to Y}(\tau)$ be as above. Then a *global strategy* (for $\mathtt{E}$) in this game is a function f associating to each expression τ' occurring in some nonterminal position of the game and such that $\mathtt{Player}(\tau') = \mathtt{E}$ with some local strategy $f_{\tau'}$ for τ'.

[7] As a limit case for $k = 0$, this condition also applies to $p_i = (=(t_1 \ldots t_k), s)$ and $q_j = (=(t_1 \ldots t_k), s')$.

Definition 6.3.15. A play $\vec{p}$ of a game $G^M_{X\to Y}(\tau)$ is said to *follow* a global strategy f if and only if it follows $f_{\tau'}$ for all τ'.

Definition 6.3.16. A global strategy f for a game $G^M_{X\to Y}(\tau)$ is said to be *winning* if and only if all complete plays which follow f are winning.

Definition 6.3.17. A global strategy f for a game $G^M_{X\to Y}(\tau)$ is said to be *uniform* if and only if the set of all complete plays which follow f respects the uniformity condition of Definition 6.3.11.

The following result then connects the Game Theoretic Semantics we just defined and the Team Transition Semantics for Dynamic Dependence Logic:

Theorem 6.3.18. *Let M be a first-order model, let X and Y be teams, and let τ be any Dynamic Dependence Logic transition term. Then $M \models_{X\to Y} \tau$ if and only if the existential player* $\mathtt{E}$ *has a uniform winning strategy for $G^M_{X\to Y}(\tau)$.*

Proof. The proof is by structural induction on τ:

1. If τ is a first-order literal and $M \models_{X\to Y} \tau$, then $X \subseteq Y$ and $M \models_s \tau$ in the usual first-order sense for all $s \in X$. Then there exists only one strategy f for $\mathtt{E}$ in $G^M_{X\to Y}(\tau)$, and for this strategy we have that $f_\tau(s) = (\lambda, s)$ for all $s \in X$. Since $X \subseteq Y$, this strategy is winning; and furthermore, it is trivially uniform. Hence, $\mathtt{E}$ has a uniform winning strategy in $G^M_{X\to Y}(\tau)$.

   Conversely, suppose that $\mathtt{E}$ has an uniform winning strategy f in $G^M_{X\to Y}(\tau)$. If $M \not\models_s \tau$ for some $s \in X$, then the position (τ, s) is terminal in this game and it is not winning, which contradicts our hypothesis. Hence $M \models_s \tau$ for all $s \in X$, and furthermore - since $f_\tau(s) = (\lambda, s)$ for all $s \in X$ - we have that $X \subseteq Y$. Thus, $M \models_{X\to Y} \tau$, as required.

2. If τ is a dependence atom $=\!(t_1 \ldots t_n)$ and $M \models_{X\to Y} \tau$, then $X \subseteq Y$ and X satisfies the dependency condition associated with the atom. But then the only strategy f available to player $\mathtt{E}$ is winning, as the set of its terminal position is $\{(\lambda, s) : s \in X\}$ and $X \subseteq Y$, and it is also uniform, since the set of all possible plays of the game is $\{(=\!(t_1 \ldots t_n, s))(\lambda, s) : s \in X\}$.

   Conversely, suppose that the only strategy f available to $\mathtt{E}$ in $G^M_{X\to Y}(\tau)$ is uniform and winning. Since it is uniform, it follows at once that X satisfies the dependency condition; and since it is winning and the set of all terminal positions is $\{(\lambda, s) : s \in X\}$, we have that $X \subseteq Y$, and hence that $M \models_{X\to Y} \tau$.

3. If τ is $\exists v$ for some variable v and $M \models_{X \to Y} \tau$, then $X[F/v] \subseteq Y$ for some $F : X \to \mathtt{Dom}(M)$. Now let f be the strategy for E in $G^M_{X \to Y}(\tau)$ such that $f_\tau(s) = s[F(s)/v]$: this strategy is uniform, and the set of its terminal positions is $\{(\lambda, s[F(s)/v]) : s \in X\} = \{(\lambda, s') : s' \in X[F/v]\}$. Hence, f_τ is also winning, as required.

   Conversely, let f be any uniform winning strategy for E in $G^M_{X \to Y}(\tau)$, and define $F : X \to \mathtt{Dom}(M)$ so that

   $$f_\tau(s) = (\lambda, s[F(s)/v])$$

   for all $s \in X$.

   Since f is winning, $f_\tau(s)$ is a winning position for all $s \in X$, and hence $X[F/v] = \{s[F(s)/v] : s \in X\} \subseteq Y$. Hence, $M \models_{X \to Y} \tau$, as required.

4. If τ is $\forall v$ for some variable v and $M \models_{X \to Y} \tau$, then $X[M/v] \subseteq Y$. There exists only one (trivial, and trivially uniform) strategy for E in the game $G^M_{X \to Y}(\tau)$, as the universal player A moves in all non-terminal positions; and the set of all possible outcomes of the game for all the initial positions is $\{(\lambda, s[m/v]) : s \in X, m \in \mathtt{Dom}(M)\} = \{(\lambda, s') : s' \in X[M/v]\}$. Hence this strategy is winning, as required.

   Conversely, suppose that the unique strategy for E in $G^M_{X \to Y}(\tau)$ is winning for this game. Then, as the set of all possible outcomes is $\{(\lambda, s') : s' \in X[M/v]\}$, we have that $X[M/v] \subseteq Y$, and hence that $M \models_{X \to Y} \tau$.

5. If τ is $\tau_1 \otimes \tau_2$ for two transition terms τ_1 and τ_2 and $M \models_{X \to Y} \tau_1 \otimes \tau_2$, then $X = X_1 \cup X_2$ for two X_1 and X_2 such that $M \models_{X_1 \to Y} \tau_1$ and $M \models_{X_2 \to Y} \tau_2$. By induction hypothesis, this implies that there exist two strategies f_1 and f_2 for E which are uniform and winning in $G^M_{X_1 \to Y}(\tau_1)$ and $G^M_{X_2 \to Y}(\tau_2)$ respectively. Now define a strategy f for E in $G^M_{X \to Y}(\tau_1 \otimes \tau_2)$ as follows:

   - If τ' is part of τ_1 then $f_{\tau'} = (f_1)_{\tau'}$;
   - If τ' is part of τ_2 then $f_{\tau'} = (f_2)_{\tau'}$;
   - If τ' is $\tau_1 \otimes \tau_2$ then $f_{\tau'}(s) = \begin{cases} (\tau_1, s) & \text{if } s \in X_1; \\ (\tau_2, s) & \text{if } s \in X_2 \backslash X_1. \end{cases}$

   This strategy is clearly uniform, as f_1 and f_2 are uniform. Furthermore, it is winning: indeed, any play of $G^M_{X \to Y}(\tau_1 \otimes \tau_2)$ in which E follows it strictly contains a play of $G^M_{X_1 \to Y}(\tau)$ in which E follows f_1 or a play of $G^M_{X_2 \to Y}(\tau)$ in which E follows f_2, and in either case the game ends in a winning position in Y.

Conversely, suppose that f is a uniform winning strategy for $\mathtt{E}$ in $G^M_{X\to Y}(\tau)$. Now let $X_1 = \{s \in X : f_\tau(s) = (\tau_1, s)\}$, let $X_2 = \{s \in X : f_\tau(s) = (\tau_2, s)\}$, and let f_1 and f_2 be the restrictions of f to the subgames corresponding to τ_1 and τ_2 respectively. Then f_1 and f_2 are uniform and winning for $G^M_{X_1\to Y}(\tau_1)$ and $G^M_{X_2\to Y}(\tau_2)$ respectively, and hence by induction hypothesis $M \models_{X_1\to Y} \tau_1$ and $M \models_{X_2\to Y} \tau_2$. But $X = X_1 \cup X_2$, and hence this implies that $M \models_{X\to Y} \tau$.

6. If τ is $\tau_1 \cap \tau_2$ for some τ_1 and τ_2 and $M \models_{X\to Y} \tau_1 \cap \tau_2$, then $M \models_{X\to Y} \tau_1$ and $M \models_{X\to Y} \tau_2$. By induction hypothesis, this implies that $\mathtt{E}$ has two uniform winning strategies f_1 and f_2 for $G^M_{X\to Y}(\tau_1)$ and $G^M_{X\to Y}(\tau_2)$ respectively. Now let f be the strategy for $G^M_{X\to Y}(\tau_1 \cap \tau_2)$ which behaves like f_1 over the subgame corresponding to τ_1 and like f_2 over the subgame corresponding to τ_2 (it is not up to $\mathtt{E}$ to choose the successors of the initial positions $(\tau_1 \cap \tau_2, s)$, so she needs not specify a strategy for those). This strategy is winning and uniform, as required, because τ_1 and τ_2 are so.

   Conversely, suppose that $\mathtt{E}$ has a uniform winning strategy f for $G^M_{X\to Y}(\tau_1 \cap \tau_2)$. Since the opponent $\mathtt{A}$ chooses the successor of the initial positions $\{(\tau_1 \cap \tau_2, s) : s \in X\}$, any element of $\{(\tau_1, s) : s \in X\}$ and of $\{(\tau_2, s) : s \in X\}$ 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 τ_1 and τ_2 respectively: then f_1 and f_2 are uniform, because f is so, and they are winning for $G^M_{X\to Y}(\tau_1)$ and $G^M_{X\to Y}(\tau_2)$ respectively, because every play of these games in which $\mathtt{E}$ follows f_1 (resp f_2) starting from a position (τ_1, s) (resp. (τ_2, s)) for $s \in X$ can be transformed into a play of $G^M_{X\to Y}(\tau_1 \cap \tau_2)$ in which $\mathtt{E}$ follows f simply by appending the initial position $(\tau_1 \cap \tau_2, s)$ at the beginning.

7. If τ is $\tau_1; \tau_2$ for some τ_1 and τ_2 and $M \models_{X\to Y} \tau_1; \tau_2$, then there exists a Z such that $M \models_{X\to Z} \tau_1$ and $M \models_{Z\to Y} \tau_2$. By induction hypothesis, this implies that there exist two strategies f_1 and f_2 which are winning for $\mathtt{E}$ in $G^M_{X\to Z}(\tau_1)$ and in $G^M_{Z\to Y}(\tau_2)$ respectively. Now define a strategy f for $\mathtt{E}$ in $G^M_{X\to Y}(\tau_1; \tau_2)$ as

   - If $(f_1)_{\tau'}(s') = (\tau'', s'')$ then $f_{\tau';\tau_2}(s') = (\tau''; \tau_2, s'')$;
   - If τ' is part of τ_2 then $f_{\tau_2} = (f_2)_{\tau_2}$.

   We need to prove that this f is uniform and winning. Now, let us consider the set of all plays in which $\mathtt{E}$ follows f: it is easy to see that they will be played exactly as a game of $G^M_{X\to Z}(\tau_1)$ until a position of the form

(τ_2, s') is reached for some $s' \in Z$, and then they will be played exactly as a game of $G^M_{Z\to Y}(\tau_2)$ until a position of the form (λ, s'') is reached for some $s'' \in Y$. Hence, the strategy is winning, as it will always end in a winning position for Y, and it is uniform, because any violation of uniformity would also be a violation for f_1 or f_2.

Conversely, let f be a uniform winning strategy for $\mathtt{E}$ in $G^M_{X\to Y}(\tau_1; \tau_2)$, and let Z be the set of all assignments s such that the position (τ_2, s) occurs as part of some play of $G^M_{X\to Y}(\tau_1; \tau_2)$ in which $\mathtt{E}$ follows f. Furthermore, let the two strategies f_1 and f_2 for $\mathtt{E}$ in $G^M_{X\to Z}(\tau_1)$ and $G^M_{Z\to Y}(\tau_2)$ respectively be defined as

- If $f_{\tau';\tau_2}(s') = (\tau''; \tau_2, s'')$ then $(f_1)_{\tau'}(s) = (\tau'', s'')$;
- If τ' is part of τ_2 then $(f_2)_{\tau_2} = f_{\tau_2}$.

By construction and definition, it follows at once that τ_1 and τ_2 are uniform winning strategies for $G^M_{X\to Z}(\tau_1)$ and $G^M_{Z\to Y}(\tau_2)$ respectively. By induction hypothesis, this implies that $M \models_{X\to Z} \tau_1$ and that $M \models_{Z\to Y} \tau_2$, and finally that $M \models_{X\to Y} \tau_1; \tau_2$.

□

Theorem 6.3.18 shows that Dynamic Dependence Logic can be interpreted in terms of *reachability*: $M \models_{X\to Y} \tau$ if and only if, in the game corresponding to τ, the existential player can guarantee that the final assignment will be in Y whenever the initial assignment is in X. This corresponds exactly to the intuitions behind the notion of *transition system* which we introduced in Subsection 6.2.1, and further confirms that Dependence Logic and its variants are suitable frameworks for exploring decision-theoretic reasoning under imperfect information in a first-order setting.

Chapter 7

The Doxastic Interpretation

In this chapter, we will re-examine many of the connectives which we studied so far and consider their possible interpretation in terms of beliefs and belief updates. The framework which we will come to gradually develop in this chapter is, in practice, little more than a notational variant of Team Logic; but in the process of constructing it, we will develop doxastic interpretations for the operators and atoms of Dependence Logic and of its variants.

7.1 Belief Models

Let M be a first order model with at least two elements in its domain, let $V \subseteq \mathtt{Var}$ be a set of *state variables*, and let us consider the set of all first-order *assignments* over $\mathtt{Dom}(M)$ with domain V.

In Tarski's semantics for first order logic, such an assignment s represents a possible *state of things*: in other words, once the model M is fixed the truth value of a first order formula in an assignment depends on the elements of the model that the assignment associates to all the free variables of the formula, *and on nothing else*.

Even disregarding First Order Logic, first order assignments are very natural objects for representing states of things. For example, let us suppose that our model's domain contains all the participants to a given contest, and that our states represent the possible outcomes of the contest – and, in particular, the players who obtained the first three positions in the final ranking.

Such an outcome can be represented, in an obvious way, as an assignment s over M with domain $\{w_1, w_2, w_3\}$: here, $s(w_1)$ would be the identity of the winner, and so on.

For this particular example, of course, we would need to add a constraint requiring that no one can be placed in two different positions in the final ranking: this can be represented easily enough as the first order axiom

$$\phi := \neg\exists x((w_1 = x \wedge w_2 = x) \vee (w_2 = x \wedge w_3 = x) \vee (w_1 = x \wedge w_3 = x)) \quad (7.1)$$

as a condition that must hold for *all* possible states of things s, in the sense that s is an acceptable outcome if and only if $M \models_s \phi$.

Once we have added this axiom, there is not much left to do: as long as the domain of the model is the set of all participants to the contest, any assignment which satisfies the above formula represents a possible contest outcome.

A special case of this which is of no small interest is when the domain of the model M consists of only two elements 0 and 1, or "False" and "True": then an assignment is easily seen to be equivalent to a *possible world* in the sense of Kripke's Semantics for Modal Logic.

Now, let us return to our example, and let us consider an agent A who has some – not necessarily true, nor complete – belief about who will reach the first three places of our tournament. How can we represent this belief?

There are many possible choices here: for example, we could consider a probability distribution over states, or a possibility distribution [75], or even a Dempster-Shafer distribution [63, 15].

But let us limit ourselves to a very simple idea, and consider the set $X_A \subseteq (V \mapsto \texttt{Dom}(M))$ of all possible states of things (assignments) which our agent believes to be possible. This idea of representing beliefs as "sets of possible states" is fairly common in knowledge representation theory, and – even though other approaches, such as the ones described above, are certainly more sophisticated – it is a reasonable starting point.

Furthermore, this approach plays on the analogy between our framework and modal logic: indeed, it is easy to see that, at least for the case of Boolean models, such a belief set is exactly a set of *possible worlds* which an agent can see from the "actual" world. An important difference between our framework and modal logic, however, is that in our case the agents can reason only about *outcomes*, and not about their beliefs or about the beliefs of other agents.

Now, what can we do with beliefs? To begin with, we can describe their properties in some suitable logical formalism; and, as Section 7.2. will show, many primitive formulas considered in logics of imperfect information have a very natural interpretation in these terms.

But we can also *update* beliefs. In general, a (unary) *update operation* will be a function O from belief sets, or, to use the terminology in common use for logics of imperfect information, from *teams*, to sets of belief sets. We will not require these updates to be deterministic; and we will write $O(X) \mapsto Y$ as a shorthand for $Y \in O(X)$, that is, for the statement that Y is a possible outcome of updating X according to the rule O.

Binary update operators are defined analogously, as functions $\diamond$ mapping each pair of teams X and Y to a set $X \diamond Y$ of possible resulting teams; and again, we will write $X \diamond Y \mapsto Z$ for $Z \in (X \diamond Y)$, that is, for stating that the belief set Z is a possible outcome of updating X with Y according to the rule $\diamond$.

Ternary or n-ary operators can also be defined in the same way, but we will not need to consider any of them in the present work.

Once we have belief operators and a language for describing properties of belief sets we can ask a number of new questions, such as

1. Can a certain belief set be seen as the result of a certain update being applied between belief sets satisfying certain properties?

2. If we update a belief set under a certain rule, and the other belief sets used for the update (if any) satisfy certain properties, can we guarantee that the resulting belief set will satisfy certain other properties?

This kind of question is of clear practical importance in Artificial Intelligence: if some intelligent system's belief state respects a condition ϕ and our system interacts with some other system whose belief state respects another condition ψ, can we guarantee that the resulting belief states will respect some further condition θ?

The whole discipline of *belief revision* ([32]), for example, can be understood as a special case of this, as the fundamental problem of belief revision is to study and compare the ways of updating a knowledge base K if a new statement ϕ is learned which is contradictory to it.

Our framework, in itself, is vastly more general – and, of course, vastly more computationally expensive – than any system of belief revision; but on the other hand, the update operations that we will discuss here are all much simpler than those considered in belief revision. What, in the opinion of the author, logics of imperfect information can provide to the field of knowledge updating is a very general logical framework for defining update operations and reasoning about their properties.

But enough chatter. Beginning with the next section, we will define primitives and operators for a very general logic of imperfect information, containing most of the connectives which have been studied so far in the context of logics of imperfect information, and we will discuss their relevance to this kind of research program.

7.2 Atoms and First Order Formulas

In this section, we will gradually develop a logical formalism – basically, a fragment of Team Logic – and use it to state properties of belief sets.

Now, what can be said about the beliefs of an agent A, or, to be more precise, about his belief set X_A?

To begin with, we can ask, given a first order[1] formula ϕ, whether our agent A *believes* that ϕ holds. This justifies the following semantic rule:

DI-bel If ϕ is first order, $M \models_X B(\phi)$ if and only if $M \models_s \phi$ for all $s \in X$

where the expression $M \models_s \phi$ means that the assignment s satisfies ϕ in M according to the usual Tarski semantics.

As an example, let us consider again the scenario described in the previous section and the formula ϕ of Equation (7.1). Then $M \models_{X_A} B(\phi)$ is a sanity condition for our agent, corresponding to the statement that he believes that no player will get two distinct positions in the final rankings.

As another, perhaps quite unnecessary, example, suppose that our agent A believes that the winner of the contest will be female; then, for all $s \in X_A$ we will have that $M \models_s \text{Female}(w_1)$, and hence that $M \models_{X_A} B(\text{Female}(w_1))$.

In most logics of imperfect information, one would just write $M \models_X \phi$ for what we would write here as $M \models_X B(\phi)$. Furthermore, the above condition would be given just for first-order literals, and we would rely on the connectives of our logic in order to build expressions equivalent to $B(\phi)$ for complex first-order formulas ϕ, as per Proposition 2.2.9.

Here, however, we will not do so, for three different reasons. First of all, our objective in the present chapter is emphatically *not* to develop a terse formalism in which to express everything that can be expressed in a logic of imperfect information: instead, we want to illustrate a possible interpretation of logics of imperfect information, and hence it will be useful to examine many doxastically significant conditions and operators. Furthermore, it is vital for our purposes to distinguish between the first-order level of our language, which allows us to summarize the properties of all the assignments of the team, the more sophisticated kinds of atoms that we will describe later in this section,

[1]We could also use here any extension or variant of First Order Logic which admits a Tarski-style semantics, such as Transitive Closure Logic or First Order Logic augmented with the Härtig quantifier.

and the update connectives that we will introduce in Sections 7.3-7.6. The fact that first-order formulas can be decomposed in terms of first-order literals and update operators will then be, from this point of view, an interesting *theorem*, not something built in our definitions. Finally, and perhaps more practically, we want to be able to express another kind of "first-order" assertion in our language, and we need to distinguish it from belief statements as those just considered.

This new kind of first-order assertion is a *possibility* assertion, which corresponds to our agent believing some first order condition to be *possibly the case* in the "true" assignment. We can introduce this kind of assertion as follows:

DI-pos: If ϕ is first order, $M \models_X P(\phi)$ if and only if $M \models_s \phi$ for some $s \in X$

Dependence Logic and IF-Logic, the two most studied logics of imperfect information, are downwards closed and hence incapable of expressing this sort of statement. The most known formalism capable of that is Team Logic, where $P(\phi)$ corresponds to $\sim \neg\phi$, where $\neg$ is the dual negation and $\sim$ is the contradictory one. But possibility statements of this kind can also be constructed in Inclusion Logic: if the free variables of ϕ are $x_1 \ldots x_n$ then it is easy to see that $P(\phi)$ can be written as $\exists w_1 \ldots w_n(\phi[w_1 \ldots w_n/x_1 \ldots x_n] \wedge (w_1 \ldots w_n \subseteq x_1 \ldots x_n))$. By Theorem 4.3.12, this implies that Independence Logic is also capable of representing this statement.

From the point of view of the present chapter, possibility statements are very natural: for example, using them we can express that our agent A considers it *possible* that the winner will be female, that is, that $M \models_{X_A} P(\text{Female}(w_1))$.

We could also give and justify along similar lines further "first-order" conditions over belief sets: for example, we could state that an expression of the form $\text{Most}(\phi)$ holds in a team X if and only if most of the assignments[2] in X satisfy ϕ. But let us now move to conditions which cannot be verified by examining the truth value of a first order formula in all assignments of our belief set.

In the example which we are considering, what else could our agent A believe that we cannot express already? Well, to begin with, our agent could believe that he knows the identity of the winner.

It is easy enough to assert, using what we already have, that the agent knows that the winner will be a_0 for some $a_0 \in \texttt{Dom}(M)$: indeed, this is precisely the condition corresponding to $M \models_{X_A} B(w_1 = a_0)$. But what we are asking now is different: we want a formula that specifies that the agent believes that he knows the winner, but does not specify who this winner will be.

[2] In order to preserve locality (Proposition 2.2.8), it would be preferable to define an operator $\text{Most}_{\vec{x}}(\phi)$ for some tuple of variables $\vec{x}$ containing all free variables of ϕ, and have it hold in X if and only if $\text{Most}(\phi)$ holds in the restriction of X to $\vec{x}$.

An existential quantifier will not do the trick: writing $M \models_{X_A} B(\exists x(w_1 = x))$ corresponds only to asserting that our agent knows that *someone* will win the contest, not that he knows the identity of this someone! For example, if Tom, Bob and Jack are three participants to the tournament, it is easy to see that for the belief set

$X_A =$

	w_1	w_2	w_3
s_0	Tom	Bob	Jack
s_1	Bob	Tom	Jack

corresponding to the belief state in which our agent A knows that Jack will get third place, but is unsure about who between Tom and Bob will get the second place and who the first one, satisfies the formula $B(\exists x(w_1 = x))$ but it does not respect the condition we are talking about.

In fact, *no* expression of the form $B(\phi)$ or $P(\phi)$ will allow us to express our intended condition: indeed, those of the former sort are *flat* in the sense of [65], and hence hold in a team if and only if they hold in all singleton subteams, and those of the latter one are *upwards closed*, in the sense that if $M \models_X P(\phi)$ and $X \subseteq Y$ then $M \models_Y P(\phi)$.

What we seem to need is some way of saying that the value of w_1 is *the same* for all assignments in our team. This is precisely the semantics for *constancy atoms* of Dependence Logic.[3]

DI-con: For all terms t, $M \models_X =\!(t)$ if and only if, for all $s, s' \in X$, $t\langle s\rangle = t\langle s'\rangle$.

Given this definition, $=\!(w_1)$ characterizes precisely the condition of our example; and, more in general, it is easy to see that $=\!(t)$ is satisfied by a team X_A if and only if the corresponding agent A believes that he knows the value of t.

What if our agent instead believes that he knows who will be the first two placed players, but is not sure of their order? Then he would be able to guess the name of the winner from the names of the second placed participant. This corresponds nicely to the *dependence atom* $=\!(w_2, w_1)$, where the rule for dependence atoms is

DI-dep: For all $n \in \mathbb{N}$ and all terms $t_1 \ldots t_n$, $M \models_X =\!(t_1 \ldots t_n)$ if and only if, for all $s, s' \in X$ such that $t_i\langle s\rangle = t_i\langle s'\rangle$ for all $i = 1 \ldots n-1$, $t_n\langle s\rangle = t_n\langle s'\rangle$ too.

[3] Another reasonable approach could be to define an "external" existential quantifier $\exists$, and model the intended condition as $\exists x B(x = w_1)$. Later in this section, we will briefly explore this idea; but we can anticipate that this external existential quantifier is precisely the $\exists^1$ operator of [50], which we briefly mentioned in Chapter 3.

What else can we say about our agent A's beliefs? For example, he might think that everybody who has a chance to make it to first place has also a chance to make it to second place: then, for all $s \in X_A$ there exists a $s' \in X_A$ such that $s(w_1) = s'(w_2)$. This is represented by the *inclusion atom* $w_1 \subseteq w_2$, where

DI-inc: For all tuples of terms $\vec{t_1}$ and $\vec{t_2}$, of the same length, $M \models_X \vec{t_1} \subseteq \vec{t_2}$ if and only if for every $s \in X$ there exists a $s' \in X$ such that $\vec{t_1}\langle s\rangle = \vec{t_2}\langle s'\rangle$.

The meaning of the expression $w_1 \subseteq w_2$ is of course different from that of $B(w_1 = w_2)$, which would instead state that the agent believes that the first and second placed players will be the same.

Again, what else? Well, our agent could also think that no one who has some chance to take first place has also some chance to take third place – only first or second. This is represented by the *exclusion atom* $w_1 \mid w_3$, where

DI-exc: For all tuples of terms $\vec{t_1}$ and $\vec{t_2}$ of the same length, $M \models_X \vec{t_1} \mid \vec{t_2}$ if and only if for all $s, s' \in X$, $\vec{t_1}\langle s\rangle \neq \vec{t_2}\langle s'\rangle$.

Of course, this is different, and stronger, than $B(w_1 \neq w_3)$, which would only state that the agent believes that the winner and the third placed player will not be the same.

For the last pair of atoms that we will describe here, we need to modify slightly our example. Let us suppose that w_1, w_2 and w_3 represent the winners of three *different* tournaments in three successive years, so that the same player could conceivably win more than one of them. Then a possible situation might be that, in the opinion of the agent, learning the winner of the first year would not tell him anything about who the winner of the third year that he does not know already – or, in other words, that the set of all possible third year winners is the same for each fixed first year winner. This corresponds to the *independence atoms* $w_1 \perp w_3$, where

DI-ind$_C$: Let $\vec{t_1}$ and $\vec{t_2}$ be two tuples of terms, not necessarily of the same length. Then $M \models_X \vec{t_1} \perp \vec{t_2}$ if and only if for all $s, s' \in X$ there exists a $s'' \in X$ with $\vec{t_1}\langle s''\rangle = \vec{t_1}\langle s\rangle$ and $\vec{t_2}\langle s''\rangle = \vec{t_2}\langle s'\rangle$.

As pointed out in [33], $t \perp t$ is logically equivalent to $=\!(t)$. The reason for this is clear: in our interpretation, $t \perp t$ means that our agent thinks that he would not learn anything new about t by being told the value of t, and this is possible only if he believes that he knows it already.

Finally, our agent may think that, for the purpose of learning who will be the winner in the third year, learning who won in the first and second year and knowing who won in the second year alone makes no difference. Perhaps, according to this agent, learning who won in the second year *would* give him valuable information, and so would learning who won in the first year; but once he learned who won in the second year, learning who won in the first year too would be quite irrelevant.

As an example of this situation, let us consider the following belief set:

$X_A =$

	w_1	w_2	w_3
s_0	Bob	Tom	Tom
s_1	Tom	Bob	Bob
s_2	Tom	Bob	Jack
s_3	Jack	Bob	Bob
s_4	Jack	Bob	Jack

Here our agent believes that if Tom won the second year then he will win the third year too, and if that if Bob won the second year then either he or Jack will win the third year. So, learning who won the second year would allow him to infer something about who will win the third year. Also, he believes that if Tom won the first year then one of Bob or Jack will win the third year, that if Jack won the first year then one of Bob or Jack will win the third year and that if Bob won the first year then Tom will certainly win the third year. So, learning who won the first year would allow him to infer something about who will win the third year. However, suppose that Tom is told who won the second year. Then learning also who won in the first year would tell him nothing new about who will win the third year: if the winner of the second year is Tom, then he is also the winner of the third year, and if the winner of the second year is Bob then, no matter who won the first year, both Bob and Jack are possible winners for the third year competition.

This is modeled by the following, more general *independence atom* from [33], which we recalled in Subsection 2.4.1:

DI-ind: Let $\vec{t}_1$, $\vec{t}_2$ and $\vec{t}_3$ be two tuples of terms, not necessarily of the same length. Then $M \models_X \vec{t}_2 \perp_{\vec{t}_1} \vec{t}_3$ if and only if for all $s, s' \in X$ with $\vec{t}_1\langle s\rangle = \vec{t}_1\langle s'\rangle$ there exists a $s'' \in X$ with $\vec{t}_1\vec{t}_2\langle s''\rangle = \vec{t}_1\vec{t}_2\langle s\rangle$ and $\vec{t}_1\vec{t}_3\langle s''\rangle = \vec{t}_1\vec{t}_3\langle s'\rangle$.

In particular, it is not difficult to see that the above team satisfies $w_1 \perp_{w_2} w_3$, as required.[4]

It might be of course possible to consider other notions of dependence or independence, and justify them along similar lines. And indeed, in the opinion

[4] As observed in [33], we then have that $=(x, y)$ is equivalent to $y \perp_x y$, and so on. This can be justified along the same lines in which it was justified that $=(x)$ is equivalent to $x \perp x$.

of the author at least, one of the main future research directions in the field of logics of imperfect information will be the search of new, doxastically significant notions of atom and the study of the relationships between the corresponding logics of imperfect information.

However, we already have more than enough basic material for the purposes of this chapter.

Let us now add the classical *conjunction* and *disjunction* to our language:

DI-or: $M \models_X \psi \vee \theta$ if and only if $M \models_X \phi$ or $M \models_X \theta$.

DI-and: $M \models_X \psi \wedge \theta$ if and only if $M \models_X \psi$ and $M \models_X \theta$.

There is not much to say here about these two connectives, as they are simply a way to join together basic belief constraints into more complex ones.

Can we also add the classical quantifiers at this point? Nicely enough, this has been already done by Kontinen and Väänänen in [50] by defining the $\exists^1$ and $\forall^1$ quantifiers. These quantifiers have properties which are quite different from the ones of the "usual" existential and universal quantifiers for logics of imperfect information, which will be interpreted in Section 7.6 in terms of update operations. Here we will write $\exists$ and $\forall$ for the $\exists^1$ and $\forall^1$ of [50]. Their truth conditions are

DI-exists: $M \models_X \exists x \psi(x)$ if and only if there exists a $m \in \mathtt{Dom}(M)$ such that $M \models_X \psi(m)$;

DI-forall: $M \models_X \forall x \psi(x)$ if and only if for all $m \in \mathtt{Dom}(M)$, $M \models_X \psi(m)$

where $\psi(m)$ stands for the formula obtained by substituting m (or, to be more precise, a new constant whose interpretation is the element m) for x in $\psi(x)$.

These new connectives actually allow us to do without our many non first-order atoms: for example, it is not difficult to see at this point that $=\!(x, y)$ is equivalent to $\forall u \exists v B(u \neq x \vee v = y)$, $x \subseteq y$ is equivalent to $\forall u(B(u \neq x) \vee P(u = y))$, $x \mid y$ is equivalent to $\forall u(B(u \neq x) \vee B(u \neq y))$ and $y \perp_x z$ is equivalent to $\forall u_1 u_2 u_3 (B(u_1 \neq x \wedge u_2 \neq y) \vee B(u_1 \neq x \wedge u_3 \neq z) \vee B(u_1 = x \wedge u_2 = y \wedge u_3 = z))$.

Examining logics of imperfect information in terms of these quantifiers, rather than in terms of some classes of non first-order atom, may actually be a promising – and, at the moment, largely unexplored – avenue of research; but in this work, we thought it better to begin by describing the atoms that have been studied so far, and only show their translations in term of basic quantifiers at a second occasion.

Another connective that we are missing is a negation. We can certainly negate a first-order formula inside a belief or possibility statement; but can we also consider an "external" negation, to go with our external quantifiers and our classical conjunction and disjunctions?

It turns out that we can, of course, and that such an operator is precisely the *contradictory negation* $\sim \phi$ of Team Semantics. Here we call it simply $\neg\phi$, as there is no "other" negation with which it may be confused. Its semantics is precisely the one that we would expect:

DI-not: $M \models_X \neg\psi$ if and only if $M \not\models_X \psi$.

Given such an operator, we can as usual rewrite $\phi \wedge \psi$ as $\neg(\neg\phi \vee \neg\psi)$ and $\forall x\psi$ as $\neg\exists x\neg\psi$.

Furthermore, as in the case of the "diamond" and "box" operators of Modal Logic, we can remove one of them from our list of primitives: for example, we could keep only $P(\phi)$, and define $B(\phi)$ as $\neg P(\neg\phi)$. Here, however, it must be noted that the roles of the internal and external negations are quite different: the former states that something is not true with respect to the whole team, while the negation of $\neg\phi$ states that ϕ is not true *in one specific assignment.* This is rather reminiscent, although not entirely identical, to the distinction between *contradictory negation* and *dual negation* of Team Logic.[5]

Given such a theory T in the language developed so far and a suitable model M, one can consider the set $\mathrm{Bel}_M(T)$ of all belief sets which satisfy T in M: formally, we can define

$$\mathrm{Bel}_M(T) := \{X : M \models_X T\}.$$

When the choice of the model M is known, we will write $\mathrm{Bel}(T)$ rather than $\mathrm{Bel}_M(T)$.

Given two theories T_1 and T_2 and a suitable model M, we write $T_1 \models_M T_2$ if $\mathrm{Bel}_M(T_1) \subseteq \mathrm{Bel}_M(T_2)$, and $T_1 \models T_2$ if $T_1 \models_M T_2$ for all suitable models M.

The significance of these expressions is clear: $T_1 \models_M T_2$ means that whenever our belief set over the model M satisfies all the conditions of T_1, it also describes all those described in T_2, and $T_1 \models T_2$ means that this is the case even if we do not know the underlying model M.

[5] In particular, the Team Logic expression $\sim \phi$ corresponds to our $\neg B(\phi)$, while the expression $\neg\phi$ corresponds to our $B(\neg\phi)$. However, the dual negation can operate over "external" expressions too, and turns for example a conjunction $\phi \wedge \psi$ into the "tensor" operator $\phi \otimes \psi$ that we will define in Section 7.3.

7.3 Belief Updates

Consider again our agent A with his belief set X_A, and suppose that he interacts with another agent B with a different belief set X_B. Then, our first agent could update his beliefs in many different ways, according to the degree up to which he trusts his own beliefs, to the degree up to which he trust the other agent's beliefs and on a number of other possible factors.

This can be represented by defining update operations $X_A \diamond X_B$ from pairs of teams to *sets* of teams; and, as stated in Section 7.1., we will write $X_A \diamond X_B \mapsto Y$ for $Y \in (X_A \diamond X_B)$, that is, for the assertion according to which Y is a possible outcome of a $\diamond$-interaction between two agents whose beliefs are represented by X_A and X_B respectively.

There are many possible choices of update operations of this sort. Here we will consider four of them which seem, at least at first sight, to be relatively reasonable choices:

Confident update $X_A \oplus X_B$: A trusts his beliefs concerning the true assignment s_0, but learns and trusts in the same way the beliefs of B too. Therefore, $X_A \oplus X_B \mapsto Y$ if and only if $Y = X_A \cap X_B$.

Credulous update $X_A \otimes X_B$: A is willing to entertain the possibility that he is wrong and the true state is one that B believes possible and he does not. Or B may be wrong and he may be right, he does not know. Hence, $X_A \otimes X_B \mapsto Y$ if and only if $Y = X_A \cup X_B$.

Skeptical update $X_A \ominus X_B$: A *might* trust B's beliefs, but only if B appears to know more than A – that is, if B does not consider possible anything that A considers impossible. Otherwise, A refuses to perform the update. Therefore, $X_A \ominus X_B \mapsto Y$ if and only if $X_B \subseteq X_A$ and $Y = X_B$.

Openminded update $X_A \odot X_B$: A might trust B's beliefs, but only if B appears to know *less* than A – that is, if he does not consider impossible anything that A considers possible. Otherwise, A refuses to perform the update. Therefore, $X_A \odot X_B \mapsto Y$ if and only if $X_A \subseteq X_B$ and $Y = X_B$.

Note that the skeptical update and the openminded one can *fail*, that is, may not lead to any possible outcome. This is a feature, not a bug: an update operation does not need to be specified for all possible belief states, and does not need to be deterministic either – none of the update operations considered here can lead to more than one possible outcome, but nothing prevents in principle the definition of such update operators.

Furthermore, these update operators satisfy the three following properties:

Idempotence: $X \diamond X \mapsto Y$ if and only if $Y = X$;

Associativity: If $X_1 \diamond X_2 \mapsto Y$ and $Y \diamond X_3 \mapsto Z$, then there exists a W such that $X_2 \diamond X_3 \mapsto W$ and $X_1 \diamond W \mapsto Z$;

Monotonicity: If $X \diamond Y \mapsto Z$ and $Z \diamond W \mapsto X$ then $X = Z$.

The interpretation of these properties, and the reason why they may be reasonable properties to require for an update operator, should be clear. An update $X \diamond Y$ represents an interaction between two agents whose beliefs correspond to X and Y: therefore, idempotence states that whenever the two agents have the exact same beliefs, the interaction does not modify these beliefs, associativity states that, in group interactions, the order of the individual interactions is irrelevant, and monotonicity means that an agent who changed idea cannot "return back" to his previous beliefs through another interaction of the same kind.

Let us verify that the update operators that we defined satisfy these properties. For the confident and credulous updates, this is obvious; hence, we will verify the case of the skeptical update, as the one of the openminded update is completely analogous.

Idempotence: Since $X \subseteq X$, it follows at once that $X \ominus X \mapsto Y$ if and only if $X = Y$.

Associativity: Suppose that $X_1 \ominus X_2 \mapsto Y$ and $Y \ominus X_3 \mapsto Z$. Then, by the definition of the skeptical update, we have that $X_1 \supseteq X_2 = Y \supseteq X_3 = Z$. But then we have that $X_2 \ominus X_3 \mapsto X_3$, and that $X_1 \ominus X_3 \mapsto X_3 = Z$, as required.

Monotonicity: If $X \ominus Y \mapsto Z$, then $Z = Y \subseteq X$. Furthermore, if $Z \ominus W \mapsto X$ then $X = W \subseteq Z$. Hence, $Z \subseteq X \subseteq Z$, and therefore $X = Z$.

Of course, these are only a possible selection of update properties: it may well be the case that in the future interesting update operators which do not respect them will be found, or that other, more important conditions will be explored.

After this intermezzo, let us reconsider the language of Section 7.2. As we saw, this language is already capable of describing a number of properties of belief states. But now we have something new, that is, some ways of *updating* beliefs. This allows us to can ask some very natural questions: for example, if A's belief state X_A satisfies ϕ, and if B's belief state X_B satisfies ψ, what can we say about belief state corresponding to some update of X_A and X_B?

To answer this, we need to add some way of talking about updates to our language. A possibility is to consider, for each update $\diamond$, a connective $\phi \diamond \psi$ such that $M \models_X \phi \diamond \psi$ if and only if the belief set X can be seen as the result of a $\diamond$-update between a belief set satisfying ϕ and a belief set satisfying ψ.

In other words, the semantics of $\phi \diamond \psi$ will be

DI-$\diamond$: $M \models_X \phi \diamond \psi$ if and only if there exist teams Y and Z such that $Y \diamond Z \mapsto X$, $M \models_Y \phi$ and $M \models_Z \psi$.

This gives us at once the following operations:

DI-$\oplus$: $M \models_X \phi \oplus \psi$ if and only if $X = Y \cap Z$ for some Y and Z such that $M \models_Y \phi$ and $M \models_Z \psi$;

DI-$\otimes$: $M \models_X \phi \otimes \psi$ if and only if $X = Y \cup Z$ for some Y and Z such that $M \models_Y \phi$ and $M \models_Z \psi$;

DI-$\ominus$: $M \models_X \phi \ominus \psi$ if and only if $M \models_X \psi$ and there exists a $Y \supseteq X$ such that $M \models_Y \phi$.

DI-$\odot$: $M \models_X \phi \odot \psi$ if and only if $M \models_X \psi$ and there exists a $Y \subseteq X$ such that $M \models_X \phi$.

The credulous update connective is exactly the tensor connective of Team Logic, or the disjunction of Dependence Logic. The other ones, to the knowledge of the author, have not been studied in depth yet, but they do hold some interest; it is worth noting, in particular, that for downwards closed logics (such as Dependence Logic or Intuitionistic Dependence Logic, for example) $\phi \oplus \psi$ and $\phi \ominus \psi$ are equivalent and correspond to the classical conjunction $\phi \wedge \psi$.

The intended interpretation of these connectives is better understood by considering expressions of the form $\phi \diamond \psi \models \theta$. According to what we just discussed, such an expression corresponds to the statement that

> *Any possible outcome of $\diamond$-update between two belief states satisfying ϕ and ψ respectively will satisfy θ.*

The significance of such a statement, and its relevance for the kind of framework that we are presenting, is then clear.

An expression of the form $\phi \models \psi \diamond \theta$ is perhaps a little more difficult to read, but its meaning is still intuitive enough: such an entailment holds if and only if any belief state which satisfies ϕ can be thought of as the result of a $\diamond$-update between a belief state such that ψ and a belief state such that θ.

This allows us to make sense of some properties of logics of imperfect information. Here we will describe only three examples relative to the credulous update:

1. It is easy to see that $(\phi\otimes\psi)\otimes\theta \models \phi\otimes(\psi\otimes\theta)$ for all ϕ, ψ and θ. This just means that the credulous update is *associative*: if a belief state can be the result of an agent, whose belief state satisfies ϕ, performing a credulous update with some agent whose belief set satisfies ψ and *then* with some other one whose belief set satisfies θ , then it can also be the result of an agent, whose belief state satisfies ϕ, performing a credulous update with some agent whose belief state *satisfied* ψ before *he* performed a credulous update with some agent whose belief state satisfied θ.

2. The credulous update is not idempotent: in general, $\phi\otimes\phi \not\models \phi$. This can be verified easily by letting ϕ be the constancy atom $=(x)$: if the value of the variable x is constant in Y and in Z, indeed, it does not necessarily follow that it is constant in $Y \cup Z$.

   The reason for this is clear: if an agent who believes that he knows the value of x performs a credulous update with another agent who also believes that he knows the value of x, and the two agents disagree on this value, then our first agent will become unsure about who, between him and the other agent, was in the right about x.

3. If ϕ, ψ and θ are downwards closed formulas – for example, if they are expressible in Intuitionistic Dependence Logic or in Exclusion Logic – then the following "distributivity property", first pointed out by Ville Nurmi, holds:

$$(\phi\otimes\psi)\wedge(\phi\otimes\theta) \models \phi\otimes\phi\otimes(\psi\wedge\theta).$$

   According to what we just discussed, this entailment can be read as follows:

   > *If a team X can be seen as the result of a credulous update between a teams such that ϕ and one such that ψ, and also as the result of a credulous update between a team such that ϕ and one such that θ, then it is can also be the result of a credulous update between two teams such that ϕ and one such that ψ and θ.*

   This is a nontrivial – and, at least in the opinion of the author, rather interesting – property concerning belief updates and their properties.

As an aside, the last property fails if we consider non-downwards closed formulas: for example, consider the team

$$X = \begin{array}{c|ccc} & x & y & z \\ \hline s_0 & 0 & 1 & 1 \\ s_1 & 1 & 0 & 0 \end{array}$$

in any model M with at least two elements. Then $M \models_X z = 1 \otimes x \subseteq y$: indeed, for $Y = \emptyset$ and $Z = X$ we have that $M \models_Y x = 1$, $M \models_Z x \subseteq y$, and $X = Y \cup Z$. Furthermore, $M \models_X z = 1 \otimes z = 0$, as can be easily verified by splitting X into the two subteams $\{s_0\}$ and $\{s_1\}$.

However, $M \not\models_X z = 1 \otimes z = 1 \otimes (x \subseteq y \wedge z = 0)$: indeed, otherwise we could split X into three subteams X_1, X_2 and X_3 such that $M \models_{X_1} z = 1$, $M \models_{X_2} z = 1$ and $M \models_{X_3} x \subseteq y \wedge z = 0$. Now, $s_1(z) = 0 \neq 1$, and therefore s_1 would necessarily be in X_3; but then, since $M \models_{X_3} x \subseteq y$, there should be another assignment $s \in X_3$ with $s(y) = s_1(x) = 1$. The only such assignment is s_0; but $s_0(z) = 1$, and therefore it would not be the case that $M \models_{X_3} z = 0$. This contradicts our hypothesis.

As this example shows, different fragments of our language may have different properties when it comes to the entailment relation. This may be worth exploring further in the future.

7.4 Adjoints

In the previous section we considered a few update operators, and for each one of them we defined a connective expressing that a given team X can be seen the result of an update between some teams Y and Z satisfying certain properties. This increased substantially the expressive power of our formalism, and, in fact, by now our language contains the propositional fragment of most logics of imperfect information.

However, this is not all that we can do with these update operations. In particular, it may be useful to be able to make conjectures about what *would* happen if we updated the team in a certain way. In particular, any update operator $\diamond$ induces a corresponding implication $\xrightarrow{\diamond}$ between sets of belief sets, defined as

DI-$\xrightarrow{\diamond}$: $M \models_X \phi \xrightarrow{\diamond} \psi$ if and only if for all Y s.t. $M \models_Y \phi$ and for all Z such that $X \diamond Y \mapsto Z$, $M \models_Z \psi$.

The intended interpretation of these new connectives is clear: $\phi \xrightarrow{\diamond} \psi$ corresponds to the statement asserting that *any* $\diamond$-update between the beliefs X of our agent and the beliefs Y of *any* other agent such that $M \models_Y \phi$ will *always* result in a belief Z such that ψ.

This easily implies that

$$\phi \diamond \psi \models_M \theta \Leftrightarrow \phi \models_M \psi \xrightarrow{\diamond} \theta$$

for all ϕ, ψ and θ and for all models M. In other words, the operator $\xrightarrow{\diamond}$ is the *right adjoint* of the operator $\diamond$.

This notion of *adjointness* is precisely the one studied in [3], in which it was one of the motivations given for the definitions of the intuitionistic and linear implications.

Hence, it should come to no surprise that now our framework will allow us to recover both these implications, plus two new ones. Indeed, by instantiating our definition of $\xrightarrow{\diamond}$ with the update operators of the previous section we obtain the following connectives:

Confident implication: $M \models_X \phi \xrightarrow{\oplus} \psi$ if and only if for all Y such that $M \models_Y \phi$ it holds that $M \models_{X\cap Y} \psi$;

Credulous implication: $M \models_X \phi \xrightarrow{\otimes} \psi$ if and only if for all Y such that $M \models_Y \phi$ it holds that $M \models_{X\cup Y} \psi$;

Skeptical implication: $M \models_X \phi \xrightarrow{\ominus} \psi$ if and only if for all $Y \subseteq X$ such that $M \models_Y \psi$ it holds that $M \models_Y \theta$;

Openminded implication: $M \models_X \phi \xrightarrow{\odot} \psi$ if and only if for all $Y \supseteq X$ such that $M \models_Y \psi$ it holds that $M \models_Y \theta$.

Skeptical implication and credulous implication are precisely the intuitionistic and linear implications of [3], which we recalled in Subsection 2.4.2. Moreover, it is not difficult to see that whenever the antecedent satisfies the downwards closure property, the confident and the skeptical implications are equivalent (as would the credulous and the openminded ones in the case of an upwards closed antecedent). This, in particular, implies that for downwards closed logics (such as, for example, Intuitionistic Dependence Logic) these two forms of implication are interchangeable.

However, in general the skeptical and the confident implications are not equivalent. For example, consider the team

$$X = \begin{array}{c|cc} & x & y \\ \hline s_0 & 0 & 0 \\ s_1 & 1 & 1 \end{array}$$

in a model M with two elements 0 and 1. Then $M \models_X x \perp y \xrightarrow{\ominus} =\!(x)$: indeed, the only subteams of X in which x is independent on y are $\{s_0\}$ and $\{s_1\}$, and in these subteams x is clearly constant.

But $M \not\models_X x \perp y \xrightarrow{\oplus} =\!(x)$: indeed, for

$$Y = \begin{array}{c|cc} & x & y \\ \hline s_0 & 0 & 0 \\ s_1 & 1 & 1 \\ s_2 & 0 & 1 \\ s_3 & 1 & 0 \end{array}$$

we have that $M \models_Y x \perp y$, but that in $X \cap Y = X$ the value of x is not constant.

So now we have a new class of formulas which describe beliefs in terms of how they *would* change if they interacted with other beliefs; and this is, of course, of potential significance for a number of practical applications. More in general, it seems that the problem of deciding, given two formulas ϕ and ψ of our language (or of a fragment thereof) and a fixed model M, whether $\phi \models_M \psi$, is of no small relevance for the field of knowledge updating; and that the same may also be said for the problem of whether ϕ entails ψ in all models.

The interpretation just discussed also clarifies the fact, already pointed out in [3], that $=\!(x, y)$ is logically equivalent to $=\!(x) \xrightarrow{\ominus} =\!(y)$ (or equivalently, since constancy atoms are downwards closed, to $=\!(x) \xrightarrow{\oplus} =\!(y)$), and that more in general dependence atoms can be decomposed in terms of constancy atoms and intuitionistic implication: indeed, a belief set X satisfies $=\!(x) \xrightarrow{\oplus} =\!(y)$ if and only if the corresponding agent, by trusting the beliefs Y of some agent who believes that he knows the value of x, will reach a new belief state $X \cap Y$ in which he believes to know the value of y too. This corresponds precisely to the doxastic interpretation of $=\!(x, y)$.

7.5 Minimal updates

Let $\diamond$ be an update operator satisfying the *idempotence*, *associativity* and *monotonicity* conditions described in Section 7.3. Then $\diamond$ defines a *partial order* over belief sets as follows:

$$X \leq^{\diamond} Y \Leftrightarrow \exists X' \text{ s.t. } X \diamond X' = Y.$$

Indeed, by idempotence we have that $X \leq^{\diamond} X$; by the associativity of the operator, we have that the $\leq^{\diamond}$ relation is transitive; and by the monotonicity of the operator, we have that if $X \leq^{\diamond} Y$ and $Y \leq^{\diamond} X$ then $X = Y$.

The interpretation of the $\leq^{\diamond}$ operator in our framework is the following: $X \leq^{\diamond} Y$ if and only if an agent, whose belief set is X, *may* reach the belief state Y through a sequence of $\diamond$-updates.

Different update operators, of course, may generate the same partial order. In particular, for the operators that we considered we have that

$$X \leq^{\oplus} Y \Leftrightarrow X \leq^{\ominus} Y \Leftrightarrow Y \subseteq X$$

and

$$X \leq^{\otimes} Y \Leftrightarrow X \leq^{\odot} Y \Leftrightarrow X \subseteq Y.$$

In other words, a belief state Y can be reached from a state X through a confident or a skeptical update if and only if Y represents a *stricter* belief than X does, and it can be reached through a credulous or an openminded statement if and only if it represents a *looser* belief than X does.

By the way, this allows us to give an alternative definition of the skeptical and openminded updates in terms of the confident and credulous ones as follows:

Skeptical update, v2: $X \ominus Y \mapsto Z$ if and only if $X \leq^{\oplus} Y$ and $X \oplus Y \mapsto Z$;

Openminded update, v2: $X \odot Y \mapsto Z$ if and only if $X \leq^{\otimes} Y$ and $X \otimes Y \mapsto Z$.

This seems to be an instance of a more general phenomenon: given an update operation $\diamond$ satisfying our three conditions, we can always generate a new operation $\diamond'$ as

$$X \diamond' Y \mapsto Z \Leftrightarrow X \leq^{\diamond} Y \text{ and } X \diamond Y = Z.$$

These new update operations $\diamond'$, in other words, are defined precisely as the older operations $\diamond$, except that now our agent – who believes that X – is willing to perform an update with Y if and only if Y itself is a belief state that he *could* possibly reach through a $\diamond$-update.

Now, let $\mathcal{V}$ be any family of belief sets, let X be a belief, and let us define $X \diamond \mathcal{V}$ as $\{Z : \exists Y \in \mathcal{V} \text{ s.t. } X \diamond Y \mapsto Z\}$. As usual, we will write $X \diamond \mathcal{V} \mapsto Z$

for $Z \in (X \diamond \mathcal{V})$: in other words, with $X \diamond \mathcal{V} \mapsto Z$ we mean that Z is a possible outcome of updating X with some $Y \in \mathcal{V}$.

Suppose now that our agent can choose which $Y \in \mathcal{V}$ to pick to update his beliefs, and also select the resulting Z if more than one exists: which strategy could he use?

A reasonable choice might be that our agent will attempt to make a $\diamond$-*minimal* update, that is, one that does not commit him any more than necessary: in particular, if he can reach both Z_1 and Z_2, and he could reach Z_2 from Z_1 through *another* $\diamond$-update, then he should pick Z_1 over Z_2. This can be defined formally as the notion of *minimal update*:

$X \ \square\ \mathcal{V} \mapsto Z$ if and only if there is a Y such that $X \diamond Y \mapsto Z$ and Z is $\leq^{\diamond}$-minimal in $X \diamond \mathcal{V}$.

Here, stating that Z is $\leq^{\diamond}$-minimal in $X \diamond \mathcal{V}$ means simply that there exists no $Z' \in X \diamond \mathcal{V}$ with $Z' <^{\diamond} Z$.

Substituting $\diamond$ with the four update operators considered so far, we get the following updates:

Minimal confident update: $X \boxplus \mathcal{V} \mapsto Z$ if and only if there exists a $Y \in \mathcal{V}$ such that $X \cap Y = Z$, and if for all $Z' \supsetneq Z$ and all $Y' \in \mathcal{V}$ it holds that $X \cap Y' \neq Z'$;

Minimal credulous update $X \boxtimes \mathcal{V} \mapsto Z$ if and only if there exists a $Y \in \mathcal{V}$ such that $X \cup Y = Z$, and for all $Z' \subsetneq Z$ and all $Y' \in \mathcal{V}$ it holds that $X \cup Y' \neq Z'$;

Minimal skeptical update: $X \boxminus \mathcal{V} \mapsto Z$ if and only $Z \subseteq X$, $Z \in \mathcal{V}$ and for all Z' with $Z \subsetneq Z' \subseteq X$ it holds that $Z' \notin \mathcal{V}$;

Minimal openminded update: $X \boxdot \mathcal{V} \mapsto Z$ if and only if $X \subseteq Z$, $Z \in \mathcal{V}$, and for all Z' with $X \subseteq Z' \subsetneq Z$ it holds that $Z' \notin \mathcal{V}$.

As before, we can at this point define connectives $\phi \ \square\ \psi$ for describing that a team X is a possible result of a minimal update of this kind, and connectives $\phi \overset{\square}{\longrightarrow} \psi$ for describing that whenever we perform a minimal update between X and a the family of teams satisfying ϕ, the result will satisfy ψ.

The formal definitions would then be

DI-$\square$: $M \models_X \phi \ \square\ \psi$ if and only if there exists a Y such that $M \models_Y \phi$ and $Y \ \square\ \mathrm{Bel}(\psi) \mapsto X$;

DI-$\overset{\square}{\longrightarrow}$: $M \models_X \phi \overset{\square}{\longrightarrow} \psi$ if and only if whenever $X \ \square\ \mathrm{Bel}(\phi) \mapsto Z$ it holds that $M \models_Z \psi$.

Here, as usual, Bel(ψ) represents the family of all teams which satisfy ψ in M.

Here we will not give the instantiations of these connectives for the four updates described above, nor will we discuss their properties.

All that we will point out is that the minimal skeptical implication connective $\phi \xrightarrow{\boxminus} \psi$ is precisely the *maximal implication* $\phi \hookrightarrow \psi$ mentioned in [49] and defined as

DI-maximp: $M \models_X \phi \hookrightarrow \psi$ if and only if for all $Y \subseteq X$ such that $M \models_Y \phi$ and $M \not\models_Z \phi$ for all Z with $Y \subsetneq Z \subseteq X$, $M \models_Y \psi$;

Thus, even this connective can be interpreted in this framework. This notion of minimal update appears to be rather natural, and it probably deserves further study; however, for the moment we will content ourselves with having defined it and shown how to recover the $\hookrightarrow$ implication through it.

As an aside, this implication allows us to decompose independence atoms: for example, it is not difficult to see that $y \perp_x z$ is equivalent to $=\!(x) \hookrightarrow y \perp z$.

7.6 Quantifiers

So far, our operators have treated assignments as if they were point-like possible worlds. This is not the case, of course: for example, our agent may be confident about the values of certain variables, but not about these of the others. Furthermore, we have no way so far of adding or removing variables from the domains of our teams. In this section, we will attempt to remedy this.

Let us begin with a *forgetting* operator ρx, where x is a variable, which has the effect of removing the variable x from the domain of our team: more precisely, for all belief states X we define $\rho x(X)$ as $X_{\backslash x}$, that is, as the team containing the restrictions of all assignments in X to $\texttt{Dom}(X)\backslash\{x\}$. In the case that x is not in the domain of X to begin with, this operator has no effect.

The doxastic meaning of this operator is the one suggested by its value: after performing the update (ρx), our agent forgets everything about the value of the variable x, and even the fact that this variable exists to begin with! This is not the same as our agent simply professing ignorance about the value of x – this would be another operator, that we will examine later – as here we are *really* erasing the variable from our domain.

As in the case of the binary update operators considered in the previous sections, this forgetting operator corresponds to *two* distinct connectives, which can be formally defined as

DI-forgotten: $M \models_X (\rho x)\phi$ if and only if there exists a team X' such that $X = (\rho x)X' = X'_{\backslash x}$ and $M \models_{X'} \phi$.

DI-forgetting: $M \models_X (\eta x)\phi$ if and only if $M \models_{X'} \phi$, where $X' = (\rho x)X = X_{\backslash x}$;

In other words, $(\eta x)\phi$ holds in a team X if, starting from X and forgetting the values of the variable x, we obtain a belief state which satisfies ϕ, and $(\rho x)\phi$ holds in a team X if this team can be obtained by starting from a team X' which satisfies ϕ and forgetting the value of x. As always, these two operators are adjoints, that is,

$$(\rho x)\phi \models \psi \Leftrightarrow \phi \models (\eta x)\psi.$$

A combination of these two operators which is of particular importance is the *disbelieving operator* $Dx\phi = (\eta x)(\rho x)\phi$. The intuition here is that $M \models_X Dx\phi$ if, apart from the value of the variable x, the team X could correspond to a belief set which satisfies ϕ. This is compatible with the corresponding semantic rule: indeed, by combining the semantic rules for the forgetting and remembering operators, one can see that $M \models_X Dx\phi$ if and only if there exists a X' such that $X_{\backslash x} = X'_{\backslash x}$ and such that $M \models_{X'} \phi$.

The condition $X_{\backslash x} = X'_{\backslash x}$ is easily seen to be equivalent to the existence of a function $H : X \to \mathcal{P}(\mathtt{Dom}(X))\backslash\{\emptyset\}$ such that $X' = X[H/x] = \{s[m/x] : s \in X, m \in H(s)\}$. Therefore, the $Dx\phi$ operator corresponds precisely to the lax existential quantifier rule **TS-∃-lax** mentioned in Subsection 2.2.1.

Just like all connectives, $Dx\phi$ can also be built directly from some belief update operator. This is our first true case of a *non-deterministic* belief update: more precisely, we can define it as

$$(Dx)X \mapsto Y \text{ if and only if } X_{\backslash x} = Y_{\backslash x}.$$

The significance of this operator in our framework should be easy to see: in brief, we have that $(Dx)X \mapsto Y$ if and only if an agent who starts from the belief X and, disbelieving his previous opinions about the possible values of the variable x, changes them *and nothing else*, can possibly reach the belief state represented by Y.

The quantifier $(Dx)\phi$ is then the unary equivalent of the $\phi \diamond \psi$ connectives considered in the previous sections: in brief, $M \models_X (Dx)\phi$ if and only if there exists a team Y with $M \models_Y \phi$ and $(Dx)Y \mapsto X$.

Of course, we also get another quantifier $(Rx)\phi$, which is satisfied by a belief state X if and only if for all Y such that $(Dx)X \mapsto Y$ we have that $M \models_Y \phi$: in other words, $M \models_X (Rx)\phi$ corresponds that our agent, whose belief state is

X, is confident that ϕ would hold even if his beliefs about the value of x were wrong. Hence, we may perhaps call it the *regardless* quantifier. Thus, we have obtained a new pair of unary connectives:

DI-disbelief: $M \models_X (Dx)\phi$ if and only if there exists a Y such that $Y_{\backslash x} = X_{\backslash x}$ and $M \models_Y \phi$;

DI-regardless: $M \models_X (Rx)\phi$ if and only if for all teams Y with $Y_{\backslash x} = X_{\backslash x}$ it holds that $M \models_Y \phi$.

The "regardless" operator is a lax version of the universal quantifier $\sim \exists x \sim \ldots$ of Team Semantics; and, as always, these two operators are adjoints, that is,

$$(Dx)\phi \models \psi \Leftrightarrow \phi \models (Rx)\psi.$$

Finally, let us consider the following scenario: our agent's belief is represented by the team X, which satisfies some property ϕ, but now the agent decides that he does not trust at all his own opinion about the value of some variable x. Then the new belief state is given by

$$¡x\phi = X[M/x] = \{s[m/x] : s \in X, m \in \mathtt{Dom}(M)\}$$

that is, the new belief state of our agent is the same as the old one, except that now our agent knows nothing at all about x. This, once again, represents a situation in which our agent doubts the validity of his beliefs about x; but where the disbelief operator Dx corresponds to the agent revising these beliefs in some arbitrary, nondeterministic way, this new operator $¡x$ has the agent taking an agnostic position about the possible values of x and moving to a belief state in which he knows nothing about it.

As in all previous cases, this allows us to develop two new connectives, that we will call the *doubted* and the *doubting* quantifiers:

DI-doubted: $M \models_X (¡x)\phi$ if and only if $X = Y[M/x]$ for some Y such that $M \models_Y \phi$.

DI-doubting: $M \models_X (!x)\phi$ if and only if $M \models_{X[M/x]} \phi$;

The $!x$ connective is exactly the $!x$ operator in Team Logic, or the one written as $\forall x$ in Dependence Logic or in many other logics of imperfect information. The other one is, to the knowledge of the author, new, but its interpretation is clear: $M \models_X (¡x)\phi$ if the belief X can be seen as the result of taking a belief state Y which satisfies ϕ, and doubting its guess about it. As always, we have that

$$¡x\phi \models \psi \Leftrightarrow \phi \models !x\psi.$$

This concludes the chapter. It was perhaps in many regards a bit informal, and we touched a number of issues without examining them in much depth; however, our purpose here was not to formulate and study a specific logical formalism, but rather to illustrate how formulas of logics of imperfect information have natural interpretations as statements about beliefs and belief updates.

We leave to the reader to decide whether we achieved this objective; but it is the hope of the author that this discussion, as well as the results presented in other parts of this thesis, may provide further incentive for the development and study of this interesting family of logics.

Chapter 8

Conclusions

In this work, we examined the properties of a number of doxastically inspired variants of Dependence Logic. The author's aims in pursuing this research program were

1. To *explore* the space of the possible variants of Dependence Logic;
2. To *achieve*, through the above mentioned analysis, a fuller understanding of the potential and the properties of Team Semantics;
3. To *analyze* the dynamics of information change which lies underneath this semantics;
4. To *argue* that first-order logics of imperfect information provide a natural framework for reasoning about beliefs and belief updates in a first order setting.

According to my current doxastic state, all of these objectives have been met in full.

Sadly, none of the logics considered in this work is capable of expressing statements about higher-order beliefs; and therefore, I find myself unable to formulate exactly my hope that the reader's belief state is now, if not necessarily in complete agreement with the above evaluation, at least not incompatible with a certain degree of satisfaction and of interest in the possibilities of Team Semantics and in its doxastic interpretation.

One research question which I left essentially untouched, and which is related to the issue of higher-order beliefs, consists in the relationship between our approach and dynamic modal logics of belief and knowledge [5, 60, 70]. We studiedly avoided such a comparison, even though many of the notions which we considered (the announcement operators of Chapter 3, for example, or the team update operations considered in Chapters 6 and 7) have clear parallels in such formalisms: indeed, an overeager attempt of establishing connections between Kripke Semantics and Team Semantics could have risked hiding some of the peculiarities and possibilities of the latter.[1] However, the doxastic interpretation of Team Semantics is now, in the opinion of the author, more than mature enough for such comparisons to be opportune: the groundwork for a more formal study of the connections between these two subjects is more than ready, and – in particular – the analysis of Chapter 7 appears to be a promising starting point for such an enterprise.

Another interesting possibility is to consider graded beliefs and finer variants of Team Semantics. Studying the Nash Equilibria of semantic games, after [61, 25, 62, 31], is surely an option; but another, perhaps more promising one is to take Team Semantics as primary, as we did in all of this work, and adapt it to "probabilistic teams" after the fashion of [25]. But of course, one must also keep in mind that probability theory is not the only available means for representing graded beliefs. Another, perhaps even more intriguing in our framework, one is to consider *fuzzy teams* (that is, fuzzy sets of assignments), thus developing an Team Semantics analogue of Cintula and Mayer's Game Theoretic Semantics for Fuzzy Logic [12].

Also, the classification of variants of Dependence Logic through generalized dependence atoms which we begun in Chapter 4 is far from complete. The most outstanding open problem, in the opinion of the author, consists in the characterization of the expressive power of Inclusion Logic; but more in general, it is clear that the space of all semantically interesting dependence notions is largely unexplored, and that its systematic study promises to hold many interesting results and surprises.

Finally, the formal properties of the Team Transition Semantics of Chapter 6 are far from entirely known, and definitely deserving of further analysis.

I conclude this work on this note. The field of Dependence Logic and Team Semantics is in a state of very rapid growth; and I can only express the hope that the results described in this thesis may be of some utility for the further development of this fascinating area of research.

[1] Also, from a practical point of view, it was simpler for the author to mostly focus on a single semantical framework and its variants.

Bibliography

[1] Samson Abramsky. Socially Responsive, Environmentally Friendly logic. In A. Tuomo and A-V. Pietarinen, editors, *Truth and Games: Essays in Honour of Gabriel Sandu*, pages 17–46. Acta Philosophica Fennica, Societas Philosophicas Fennica, 2006.

[2] Samson Abramsky. A compositional game semantics for multi-agent logics of partial information. In J. van Benthem, D. Gabbay, and B. Löwe, editors, *Interactive Logic*, volume 1 of *Texts in Logic and Games*, pages 11–48. Amsterdam University Press, 2007.

[3] Samson Abramsky and Jouko Väänänen. From IF to BI. *Synthese*, 167:207–230, 2009. 10.1007/s11229-008-9415-6.

[4] William W. Armstrong. Dependency Structures of Data Base Relationships. In *Proc. of IFIP World Computer Congress*, pages 580–583, 1974.

[5] Alexandru Baltag, Lawrence S. Moss, and Slawomir Solecki. The logic of public announcements, common knowledge, and private suspicions. In *Proceedings of the 7th conference on Theoretical aspects of rationality and knowledge*, TARK '98, pages 43–56, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[6] Julian Bradfield. Independence: Logics and concurrency. In Peter Clote and Helmut Schwichtenberg, editors, *Computer Science Logic*, volume 1862 of *Lecture Notes in Computer Science*, pages 247–261. Springer Berlin / Heidelberg, 2000.

[7] John P. Burgess. A remark on Henkin sentences and their contraries. *Notre Dame Journal of Formal Logic*, 3(44):185–188, 2003.

[8] Peter Cameron and Wilfrid Hodges. Some Combinatorics of Imperfect Information. *The Journal of Symbolic Logic*, 66(2):673–684, 2001.

[9] Marco A. Casanova, Ronald Fagin, and Christos H. Papadimitriou. Inclusion dependencies and their interaction with functional dependencies. In *Proceedings of the 1st ACM SIGACT-SIGMOD symposium on Principles of database systems*, PODS '82, pages 171–176, New York, NY, USA, 1982. ACM.

[10] Marco A. Casanova and Vânia M. P. Vidal. Towards a sound view integration methodology. In *Proceedings of the 2nd ACM SIGACT-SIGMOD symposium on Principles of database systems*, PODS '83, pages 36–47, New York, NY, USA, 1983. ACM.

[11] Ashok K. Chandra and Moshe Y. Vardi. The implication problem for functional and inclusion dependencies is undecidable. *SIAM Journal on Computing*, 14(3):671–677, 1985.

[12] Petr Cintula and Ondrej Majer. Towards evaluation games for fuzzy logics. In Ondrej Majer, Ahti-Veikko Pietarinen, Tero Tulenheimo, Olga Pombo, Juan Manuel Torres, John Symons, and Shahid Rahman, editors, *Games: Unifying Logic, Language, and Philosophy*, volume 15 of *Logic, Epistemology, and the Unity of Science*, pages 117–138. Springer Netherlands, 2009.

[13] Christopher J. Date. *Introduction to Database Systems.* Addison Wesley, 8th edition, 2003.

[14] Paul Dekker. A guide to dynamic semantics. *ILLC Publications*, (PP-2008-42), 2008.

[15] Arthur Dempster. Upper and lower probabilities induced by a multivalued mapping. In Roland Yager and Liping Liu, editors, *Classic Works of the Dempster-Shafer Theory of Belief Functions*, volume 219 of *Studies in Fuzziness and Soft Computing*, pages 57–72. Springer Berlin / Heidelberg, 2008.

[16] Jakub Dotlačil. Fastidious distributivity. In A. Chereches N. Ashton and D. Lutz, editors, *Proceedings of SALT 21*, pages 313–332, 2011.

[17] Arnaud Durand and Juha Kontinen. Hierarchies in dependence logic. *CoRR*, abs/1105.3324, 2011.

[18] Herbert B. Enderton. Finite partially-ordered quantifiers. *Mathematical Logic Quarterly*, 16(8):393–397, 1970.

[19] Fredrik Engström. Generalized quantifiers in dependence logic. *Journal of Logic, Language and Information (to appear)*, 2011.

[20] Ronald Fagin. Generalized first-order spectra and polynomial-time recognizable sets. In *Complexity of Computation, SIAM-AMS Proceedings,*, volume Vol. 7, pages 43–73, 1974.

[21] Ronald Fagin. Multivalued dependencies and a new normal form for relational databases. *ACM Transactions on Database Systems*, 2:262–278, September 1977.

[22] Ronald Fagin. A normal form for relational databases that is based on domains and keys. *ACM Transactions on Database Systems*, 6:387–415, September 1981.

[23] Ronald Fagin and Moshe Vardi. The theory of data dependencies An overview. In *Automata, Languages and Programming*, pages 1–22. Springer Berlin / Heidelberg, 1984.

[24] Thomas Forster. Deterministic and Nondeterministic Strategies for Hintikka games in First-order and Branching-quantifier logic. *Logique et Analyse*, 49(195), 2006.

[25] Pietro Galliani. Game Values and Equilibria for Undetermined Sentences of Dependence Logic. MSc Thesis. ILLC Publications, MoL–2008–08, 2008.

[26] Pietro Galliani. Epistemic operators and uniform definability in dependence logic. In Juha Kontinen and Jouko Väänänen, editors, *Proceedings of Dependence and Independence in Logic*, pages 4–29. ESSLLI 2010, 2010.

[27] Pietro Galliani. Epistemic operators and uniform definability in dependence logic. *Studia Logica*, 2010. To Appear.

[28] Pietro Galliani. Sensible semantics of imperfect information. In Mohua Banerjee and Anil Seth, editors, *Logic and Its Applications*, volume 6521 of *Lecture Notes in Computer Science*, pages 79–89. Springer Berlin / Heidelberg, 2011.

[29] Pietro Galliani. General models and entailment semantics for independence logic. *Notre Dame Journal of Formal Logic (to appear)*, 2012.

[30] Pietro Galliani. Inclusion and exclusion dependencies in team semantics: On some logics of imperfect information. *Annals of Pure and Applied Logic*, 163(1):68 – 84, 2012.

[31] Pietro Galliani and Allen L. Mann. Lottery semantics. In Juha Kontinen and Jouko Väänänen, editors, *Proceedings of Dependence and Independence in Logic*, pages 118–132. ESSLLI 2010, 2010.

[32] Peter Gardenfors, editor. *Belief Revision*. Cambridge University Press, New York, NY, USA, 1992.

[33] Erich Grädel and Jouko Väänänen. Dependence and Independence. *Studia Logica (to appear)*, 2010.

[34] Jeroen Groenendijk and Martin Stokhof. Dynamic Predicate Logic. *Linguistics and Philosophy*, 14(1):39–100, 1991.

[35] Leon Henkin. Completeness in the theory of types. *The Journal of Symbolic Logic*, 15:81–91, 1950.

[36] Leon Henkin. Some Remarks on Infinitely Long Formulas. In *Infinitistic Methods. Proc. Symposium on Foundations of Mathematics*, pages 167–183. Pergamon Press, 1961.

[37] Jaakko Hintikka. *The Principles of Mathematics Revisited*. Cambridge University Press, 1996.

[38] Jaakko Hintikka. No scope for scope? *Linguistics and Philosophy*, 20(5):515–544, 1997.

[39] Jaakko Hintikka and Gabriel Sandu. Informational independence as a semantic phenomenon. In J.E Fenstad, I.T Frolov, and R. Hilpinen, editors, *Logic, methodology and philosophy of science*, pages 571–589. Elsevier, 1989.

[40] Jaakko Hintikka and Gabriel Sandu. Game-Theoretical Semantics. In Johan van Benthem and Alice T. Meulen, editors, *Handbook of Logic and Language*, pages 361–410. Elsevier, 1997.

[41] Wilfrid Hodges. *A Shorter Model Theory*. Cambridge University Press, 1997.

[42] Wilfrid Hodges. Compositional Semantics for a Language of Imperfect Information. *Journal of the Interest Group in Pure and Applied Logics*, 5 (4):539–563, 1997.

[43] Wilfrid Hodges. Some strange quantifiers. In J. Mycielski, G. Rozenberg, and A. Salomaa, editors, *Structures in Logic and Computer Science*, volume

1261 of *Lecture Notes in Computer Science*, pages 51–65. Springer Berlin / Heidelberg, 1997.

[44] Wilfrid Hodges. Formal features of compositionality. *Journal of Logic, Language, and Information*, 10(1):7–28, 2001.

[45] Wilfrid Hodges. Logics of imperfect information: why sets of assignments? In J. van Benthem, D. Gabbay, and B. Löwe, editors, *Interactive Logic*, Texts in Logic and Games, pages 117–133. Amsterdam University Press, 2007.

[46] Theo Janssen. Compositionality. In Johan van Benthem and Alice ter Meulen, editors, *Handbook of Logic and Language*, pages 417–473. Elsevier, Amsterdam, 1996.

[47] Theo Janssen and Francien Dechesne. Signaling in IF-Games: A Tricky Business. In J. van Benthem, G. Heinzmann, M. Rebuschi, and H. Visser, editors, *The age of alternative logics*, pages 221–241. Springer, 2006.

[48] Jarmo Kontinen. Coherence and computational complexity of quantifier-free dependence logic formulas. In Juha Kontinen and Jouko Väänänen, editors, *Proceedings of Dependence and Independence in Logic*, pages 58–77. ESSLLI 2010, 2010.

[49] Juha Kontinen and Ville Nurmi. Team logic and second-order logic. In Hiroakira Ono, Makoto Kanazawa, and Ruy de Queiroz, editors, *Logic, Language, Information and Computation*, volume 5514 of *Lecture Notes in Computer Science*, pages 230–241. Springer Berlin / Heidelberg, 2009.

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

[51] Juha Kontinen and Jouko Väänänen. A Remark on Negation of Dependence Logic. *Notre Dame Journal of Formal Logic*, 52(1):55–65, 2011.

[52] Juha Kontinen and Jouko Väänänen. Axiomatizing first-order consequences in dependence logic. To Appear, 2011.

[53] Antti Kuusisto. Logics of imperfect information without identity. Research Note, 2011.

[54] Allen L. Mann, Gabriel Sandu, and Merlijn Sevenster. *Independence-Friendly Logic: A Game-Theoretic Approach*. Cambridge University Press, 2011.

[55] John C. Mitchell. The implication problem for functional and inclusion dependencies. *Information and Control*, 56(3):154–173, 1983.

[56] Richard Montague. Universal grammar. *Theoria*, 36:373–398, 1970.

[57] Ville Nurmi. *Dependence Logic: Investigations into Higher-Order Semantics Defined on Teams.* PhD thesis, University of Helsinki, 2009.

[58] Jan Paredaens, Paul De Bra, Marc Gyssens, and Dirk Van Gucht. *The structure of the relational database model.* Springer-Verlag New York, Inc., New York, NY, USA, 1989.

[59] Rohit Parikh. The logic of games and its applications. In *Selected papers of the international conference on "Foundations of computation theory" on Topics in the theory of computation*, pages 111–139, New York, NY, USA, 1985. Elsevier North-Holland, Inc.

[60] Jan Plaza. Logics of public communications. *Synthese*, 158:165–179, 2007.

[61] Merlijn Sevenster. Branches of imperfect information: logic, language, and computation. *PhD Thesis, Institute for Logic, Language and Computation, DS–2006–06*, 2006.

[62] Merlijn Sevenster and Gabriel Sandu. Equilibrium semantics of languages of imperfect information. *Annals of Pure and Applied Logic*, 161(5):618–631, 2010. The Third workshop on Games for Logic and Programming Languages (GaLoP), Galop 2008.

[63] Glenn Shafer. *A Mathematical Theory of Evidence.* Princeton University Press, 1976.

[64] Tero Tulenheimo. *Independence-Friendly Modal Logic.* PhD thesis, University of Helsinki, 2004.

[65] Jouko Väänänen. *Dependence Logic.* Cambridge University Press, 2007.

[66] Jouko Väänänen. Team Logic. In J. van Benthem, D. Gabbay, and B. Löwe, editors, *Interactive Logic. Selected Papers from the 7th Augustus de Morgan Workshop*, pages 281–302. Amsterdam University Press, 2007.

[67] Jouko Väänänen. Modal Dependence Logic. In Krzysztof R. Apt and Robert van Rooij, editors, *New Perspectives on Games and Interaction.* Amsterdam University Press, Amsterdam, 2008.

[68] Johan van Benthem. *Exploring logical dynamics.* Studies in Logic, Language and Information. CSLI Publications, Stanford, CA, USA, 1997.

[69] Johan van Benthem. Logic games are complete for game logics. *Studia Logica*, 75:183–203, 2003.

[70] Johan Van Benthem. Dynamic logic for belief revision. *Journal of Applied NonClassical Logics*, 17(2):129–155, 2007.

[71] Johan van Benthem, Sujata Ghosh, and Fenrong Liu. Modelling simultaneous games in dynamic logic. *Synthese*, 165:247–268, 2008.

[72] Jan van Eijck and Albert Visser. Dynamic semantics. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy.* Fall 2010 edition, 2010.

[73] Wilbur John Walkoe. Finite partially-ordered quantification. *The Journal of Symbolic Logic*, 35(4):pp. 535–555, 1970.

[74] Fan Yang. Expressing second-order sentences in intuitionistic dependence logic. In Juha Kontinen and Jouko Väänänen, editors, *Proceedings of Dependence and Independence in Logic*, pages 118–132. ESSLLI 2010, 2010.

[75] Lotfi A. Zadeh. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, 1(1):3 – 28, 1978.

[76] Ernst Zermelo. Über eine Anwendung der Mengenlehre auf die Theorie des Schachspiels. In *Proceedings of the Fifth Congress of Mathematicians*, volume 2, pages 501–504. Cambridge University Press, 1913.

Index

Samenvatting

Wij bestuderen de doxastisch geïnspireerde varianten en extensies van afhankelijkheids logica die voortvloeien uit de overweging van aankondigings operatoren en niet-functionele afhankelijkheids atomen. We lossen verscheidene open vragen in dit gebied op, waaronder de volgende twee:

1. De $\forall^1$ quantifier van (Kontinen and Vaananen, 2009) is niet uniform te definiëren in Dependence Logic;

2. Alle NP eigenschappen van teams zijn te definiëren in onafhankelijkheids logica,

Verder, generaliseren we het resultaat van Cameron and Hodges over de combinatorische eigenschappen van compositionele semantiek voor de logica van imperfecte informatie naar de oneindige casus, daardoor introduceren we een nieuw begrip van *verstandige semantiek*; en we ontwikkelen een "algemene" semantiek voor onafhankelijkheids logica (of logicas die hierin bevat zijn, zoals afhankelijkheids logica) als mede een bewijssysteem en we bewijzen correctheid en volledigheid.

Vervolgens onderzoeken we de dynamica van informatie-updates die onder de verschijning van team samantiek ligt, we breiden van Benthems wederzijdse inbeddings resultaat tussen eerste orde logica en dynamische spel logica (DGL) uit naar de casussen van afhankelijkheids logica en een niet perfecte informatie variant van DGL. Tot slot laten we zien dat veel van de operatoren en connectieven die gebruikt worden in team semantiek hebben natuurlijke interpretaties in termen van geloofs beschrijvingen en geloofs updates en we pleiten voor de *doxastische interpretatie* van dit semantische kader.

Abstract

We examine doxastically inspired variants and extensions of Dependence Logic which arise from the consideration of announcement operators and non-functional dependence atoms. We solve several open questions of the area, among them the following two:

1. The $\forall^1$ quantifier of (Kontinen and Väänänen, 2009) is not uniformly definable in Dependence Logic;

2. All NP properties of teams are definable in Independence Logic,

Furthermore, we generalize Cameron and Hodges' result about the combinatorial properties of compositional semantics for logics of imperfect information to the infinite case, thus introducing a new notion of *sensible semantics*; and we develop a "general" semantics for Independence Logic (or logics contained in it, such as Dependence Logic) as well as a proof system for which we prove soundness and completeness. We then examine the dynamics of information update which lies beneath the appearance of Team Semantics, extending van Benthem's mutual embedding result between First Order Logic and Dynamic Game Logic (DGL) to the cases of Dependence Logic and an imperfect-information variant of DGL. We use the insights arising from the embedding to develop dynamic variants of Dependence Logic and a Team Transition Semantics in which expressions are interpreted as transition systems over teams. Finally, we show that many of the operators and connectives considered in Team Semantics have natural interpretations in terms of belief descriptions and belief updates, and we argue in favor of the *doxastic interpretation* of this semantical framework.

Titles in the ILLC Dissertation Series:

ILLC DS-2006-01: **Troy Lee**
Kolmogorov complexity and formula size lower bounds

ILLC DS-2006-02: **Nick Bezhanishvili**
Lattices of intermediate and cylindric modal logics

ILLC DS-2006-03: **Clemens Kupke**
Finitary coalgebraic logics

ILLC DS-2006-04: **Robert Špalek**
Quantum Algorithms, Lower Bounds, and Time-Space Tradeoffs

ILLC DS-2006-05: **Aline Honingh**
The Origin and Well-Formedness of Tonal Pitch Structures

ILLC DS-2006-06: **Merlijn Sevenster**
Branches of imperfect information: logic, games, and computation

ILLC DS-2006-07: **Marie Nilsenova**
Rises and Falls. Studies in the Semantics and Pragmatics of Intonation

ILLC DS-2006-08: **Darko Sarenac**
Products of Topological Modal Logics

ILLC DS-2007-01: **Rudi Cilibrasi**
Statistical Inference Through Data Compression

ILLC DS-2007-02: **Neta Spiro**
What contributes to the perception of musical phrases in western classical music?

ILLC DS-2007-03: **Darrin Hindsill**
It's a Process and an Event: Perspectives in Event Semantics

ILLC DS-2007-04: **Katrin Schulz**
Minimal Models in Semantics and Pragmatics: Free Choice, Exhaustivity, and Conditionals

ILLC DS-2007-05: **Yoav Seginer**
Learning Syntactic Structure

ILLC DS-2008-01: **Stephanie Wehner**
Cryptography in a Quantum World

ILLC DS-2008-02: **Fenrong Liu**
Changing for the Better: Preference Dynamics and Agent Diversity

ILLC DS-2008-03: **Olivier Roy**
Thinking before Acting: Intentions, Logic, Rational Choice

ILLC DS-2008-04: **Patrick Girard**
Modal Logic for Belief and Preference Change

ILLC DS-2008-05: **Erik Rietveld**
Unreflective Action: A Philosophical Contribution to Integrative Neuroscience

ILLC DS-2008-06: **Falk Unger**
Noise in Quantum and Classical Computation and Non-locality

ILLC DS-2008-07: **Steven de Rooij**
Minimum Description Length Model Selection: Problems and Extensions

ILLC DS-2008-08: **Fabrice Nauze**
Modality in Typological Perspective

ILLC DS-2008-09: **Floris Roelofsen**
Anaphora Resolved

ILLC DS-2008-10: **Marian Counihan**
Looking for logic in all the wrong places: an investigation of language, literacy and logic in reasoning

ILLC DS-2009-01: **Jakub Szymanik**
Quantifiers in TIME and SPACE. Computational Complexity of Generalized Quantifiers in Natural Language

ILLC DS-2009-02: **Hartmut Fitz**
Neural Syntax

ILLC DS-2009-03: **Brian Thomas Semmes**
A Game for the Borel Functions

ILLC DS-2009-04: **Sara L. Uckelman**
Modalities in Medieval Logic

ILLC DS-2009-05: **Andreas Witzel**
Knowledge and Games: Theory and Implementation

ILLC DS-2009-06: **Chantal Bax**
Subjectivity after Wittgenstein. Wittgenstein's embodied and embedded subject and the debate about the death of man.

ILLC DS-2009-07: **Kata Balogh**
Theme with Variations. A Context-based Analysis of Focus

ILLC DS-2009-08: **Tomohiro Hoshi**
Epistemic Dynamics and Protocol Information

ILLC DS-2009-09: **Olivia Ladinig**
Temporal expectations and their violations

ILLC DS-2009-10: **Tikitu de Jager**
"Now that you mention it, I wonder...": Awareness, Attention, Assumption

ILLC DS-2009-11: **Michael Franke**
Signal to Act: Game Theory in Pragmatics

ILLC DS-2009-12: **Joel Uckelman**
More Than the Sum of Its Parts: Compact Preference Representation Over Combinatorial Domains

ILLC DS-2009-13: **Stefan Bold**
Cardinals as Ultrapowers. A Canonical Measure Analysis under the Axiom of Determinacy.

ILLC DS-2010-01: **Reut Tsarfaty**
Relational-Realizational Parsing

ILLC DS-2010-02: **Jonathan Zvesper**
Playing with Information

ILLC DS-2010-03: **Cédric Dégremont**
The Temporal Mind. Observations on the logic of belief change in interactive systems

ILLC DS-2010-04: **Daisuke Ikegami**
Games in Set Theory and Logic

ILLC DS-2010-05: **Jarmo Kontinen**
Coherence and Complexity in Fragments of Dependence Logic

ILLC DS-2010-06: **Yanjing Wang**
Epistemic Modelling and Protocol Dynamics

ILLC DS-2010-07: **Marc Staudacher**
Use theories of meaning between conventions and social norms

ILLC DS-2010-08: **Amélie Gheerbrant**
Fixed-Point Logics on Trees

ILLC DS-2010-09: **Gaëlle Fontaine**
Modal Fixpoint Logic: Some Model Theoretic Questions

ILLC DS-2010-10: **Jacob Vosmaer**
Logic, Algebra and Topology. Investigations into canonical extensions, duality theory and point-free topology.

ILLC DS-2010-11: **Nina Gierasimczuk**
Knowing One's Limits. Logical Analysis of Inductive Inference

ILLC DS-2010-12: **Martin Mose Bentzen**
Stit, Iit, and Deontic Logic for Action Types

ILLC DS-2011-01: **Wouter M. Koolen**
Combining Strategies Efficiently: High-Quality Decisions from Conflicting Advice

ILLC DS-2011-02: **Fernando Raymundo Velazquez-Quesada**
Small steps in dynamics of information

ILLC DS-2011-03: **Marijn Koolen**
The Meaning of Structure: the Value of Link Evidence for Information Retrieval

ILLC DS-2011-04: **Junte Zhang**
System Evaluation of Archival Description and Access

ILLC DS-2011-05: **Lauri Keskinen**
Characterizing All Models in Infinite Cardinalities

ILLC DS-2011-06: **Rianne Kaptein**
Effective Focused Retrieval by Exploiting Query Context and Document Structure

ILLC DS-2011-07: **Jop Briët**
Grothendieck Inequalities, Nonlocal Games and Optimization

ILLC DS-2011-08: **Stefan Minica**
Dynamic Logic of Questions

ILLC DS-2011-09: **Raul Andres Leal**
Modalities Through the Looking Glass: A study on coalgebraic modal logic and their applications

ILLC DS-2011-10: **Lena Kurzen**
Complexity in Interaction

ILLC DS-2011-11: **Gideon Borensztajn**
The neural basis of structure in language

ILLC DS-2012-01: **Federico Sangati**
Decomposing and Regenerating Syntactic Trees

ILLC DS-2012-02: **Markos Mylonakis**
Learning the Latent Structure of Translation

ILLC DS-2012-03: **Edgar José Andrade Lotero**
Models of Language: Towards a practice-based account of information in natural language

ILLC DS-2012-04: **Yurii Khomskii**
Regularity Properties and Definability in the Real Number Continuum: idealized forcing, polarized partitions, Hausdorff gaps and mad families in the projective hierarchy.

ILLC DS-2012-05: **David García Soriano**
Query-Efficient Computation in Property Testing and Learning Theory

ILLC DS-2012-06: **Dimitris Gakis**
Contextual Metaphilosophy - The Case of Wittgenstein

ILLC DS-2012-07: **Pietro Galliani**
The Dynamics of Imperfect Information