The dynamics of imperfect information

Galliani, P.

Citation for published version (APA):

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: https://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.
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:
Chapter 4. Dependencies in Team Semantics

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 $\text{Dom}(X)$. Then

$$TS-\text{const}: M \models_X = (t) \text{ if and only if, for all } s, s' \in X, t(s) = t(s').$$

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

Then $M \models_X \phi \iff M \models_X \exists z (= (z) \land \phi')$.

Proof. The proof is by induction on $\phi$.

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

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

$$M \models_X \exists z (= (z) \land z = t) \iff \exists m \in \text{Dom}(M) \text{ s.t. } M \models_{X[m/z]} z = t \iff$$

$$\exists m \in \text{Dom}(M) \text{ s.t. } t(s) = m \text{ for all } s \in X \iff M \models_X = (t)$$

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

3. If $\phi$ is $\psi_1 \lor \psi_2$, let us assume without loss of generality that the instance of $= (t)$ that we are considering is in $\psi_1$. Then $\psi'_2 = \psi_2$, and since $z$ does
4.1. Constancy Logic

not occur in $\psi_2$

$M \models X \exists z ((z) \land (\psi_1' \lor \psi_2)) \iff \exists m \text{ s.t. } M \models_{X[m/z]} \psi_1' \lor \psi_2 \iff$

$\exists m, X_1, X_2 \text{ s.t. } X_1 \cup X_2 = X, M \models_{X_1[m/z]} \psi_1' \land M \models_{X_2[\psi_2]} \iff$

$\exists m, X_1, X_2 \text{ s.t. } X_1 \cup X_2 = X, M \models_{X_1[m/z]} \exists z ((z) \land \psi_1') \land M \models_{X_2[\psi_2]} \iff$

$\exists X_1, X_2 \text{ s.t. } X_1 \cup X_2 = X, M \models_{X_1[\psi_1]} \land M \models_{X_2[\psi_2]} \iff$

$\iff M \models X \psi_1 \lor \psi_2$

as required.

4. If $\phi$ is $\psi_1 \land \psi_2$, let us assume again that the instance of $=t$ that we are considering is in $\psi_1$. Then $\psi_2' = \psi_2$, and

$M \models X \exists z ((z) \land (\psi_1' \land \psi_2)) \iff$

$\iff \exists m \text{ s.t. } M \models_{X[m/z]} \psi_1' \land M \models_{X[m/z]} \psi_2 \iff$

$\iff M \models X \exists z ((z) \land \psi_1') \land M \models \psi_2 \iff$

$\iff M \models X \psi_1 \land M \models \psi_2 \iff$

$\iff M \models X \psi_1 \lor \psi_2$

5. If $\phi$ is $\exists x \psi$, 

$M \models X \exists z ((z) \land \exists x \psi') \iff$

$\iff \exists m \text{ s.t. } M \models_{X[m/z]} \exists x \psi' \iff$

$\iff \exists m, \exists H : X[m/z] \rightarrow \text{Parts}(\text{Dom}(M)) \setminus \{\emptyset\} \text{ s.t. } M \models_{X[m/z]} H \psi' \iff$

$\iff \exists H' : X \rightarrow \text{Parts}(\text{Dom}(M)) \setminus \{\emptyset\}, \exists m \text{ s.t. } M \models_{X[H'/z][m/z]} \psi' \iff$

$\iff \exists H' : X \rightarrow \text{Parts}(\text{Dom}(M)) \setminus \{\emptyset\}, \text{ s.t. } M \models_{X[H'/z]} \exists z ((z) \land \psi') \iff$

$\iff \exists H' : X \rightarrow \text{Parts}(\text{Dom}(M)) \setminus \{\emptyset\}, \text{ s.t. } M \models_{X[H'/z]} \psi \iff$

$\iff M \models X \exists x \psi$. 
6. If $\phi$ is $\forall x\psi$,

$$M \models_X \exists z(=z ∧ ∀x\psi')$$

$\iff$

$$\exists m \text{ s.t. } M \models_{X[m/z]} ∀x\psi'$$

$\iff$

$$\exists m \text{ s.t. } M \models_{X[m/z]} M[x/z] \psi'$$

$\iff$

$$\exists m \text{ s.t. } M \models_{X[M/z]} \exists z(=z ∧ \psi')$$

$\iff$

$$M \models_{X[M/x]} \psi$$

$\iff$

$$M \models_X ∀x\psi.$$ 

\(\square\)

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

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

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

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

**Proof.** Repeatedly apply Proposition 4.1.2 to “push out” all constancy atoms from $\phi$, thus obtaining a formula, equivalent to it, of the form

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

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

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

as required. 

\(\square\)

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

---

1This 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 $\phi = \exists \vec{z} (\bigwedge_i (z_i = z) \land \psi(\vec{z}))$ be a Constancy Logic sentence in normal form.

Then $\phi$ 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 $\text{Dom}(M)$ 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 $\psi$ 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; 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]:

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 $m_1 \ldots m_n$. 
Chapter 4. Dependencies in Team Semantics

**TS-multidep**: \( M \models_X \vec{x} \rightarrow \vec{y} \) 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 \( \vec{x} \rightarrow \vec{y} \) 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” obtained 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 \( \Sigma_1 \), 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 \( \vec{z} = \text{Dom}(X) \setminus \{\vec{x},\vec{y}\} \) then it is not difficult to see that

\[
M \models_X \vec{x} \rightarrow \vec{y} \text{ if and only if } M \models_X \vec{y} \not\perp_{\vec{x}} \vec{z}.
\]

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 \( \vec{t}_2 \not\perp_{\vec{t}_1} \vec{t}_3 \) 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 \( \vec{t}_2 \not\perp_{\vec{t}_1} \vec{t}_3 \) 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

\[
\vec{t}_2 \not\perp_{\vec{t}_1} \vec{t}_3 \equiv \exists \vec{x}_1 \vec{x}_2 \vec{x}_3 (\vec{x}_1 \equiv \vec{t}_1 \wedge \vec{x}_2 \equiv \vec{t}_2 \wedge \vec{x}_3 \equiv \vec{t}_3 \not\perp_{\vec{x}_1} \vec{x}_3).
\]

Indeed, suppose that \( M \models_X \vec{t}_2 \not\perp_{\vec{t}_1} \vec{t}_3 \); then, choose the functions \( F_i \) so that \( F_i(s) = \{\vec{t}_i(s)\} \) and let \( Y = X[F_1 F_2 F_3/\vec{x}_1 \vec{x}_2 \vec{x}_3] \). 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 \( M \models_Y \vec{x}_2 \not\perp_{\vec{x}_1} \vec{x}_3 \), since \( M \models_X \vec{t}_2 \not\perp_{\vec{t}_1} \vec{t}_3 \).

\(^3\)For example, compare the values of \( x \rightarrow y \) in \( X = \{(x := 0, y := 0, z := 0), (x := 1, y := 1, z := 1)\} \) and in \( Y = \{(x := 0, y := 0, z := 0), (x := 1, y := 1, z := 0)\} \).
4.2. Multivalued Dependence Logic is Independence Logic

Conversely, suppose that $M \models X(F_1 F_2 F_3 / \vec{x}_1 \vec{x}_2 \vec{x}_3) (\vec{x}_1 = \vec{t}_1 \land \vec{x}_2 = \vec{t}_2 \land \vec{x}_3 = \vec{t}_3 \land \vec{x}_2 \perp_{\vec{t}_1} \vec{x}_3)$. Then, again for $Y = X(F_1 F_2 F_3 / \vec{x}_1 \vec{x}_2 \vec{x}_3)$ and all $i = 1 \ldots 3$, it must hold that $Y(\vec{x}_i) = \{t_i(s)\}$. But then, since $M \models \vec{t}_2 \perp_{\vec{t}_1} \vec{t}_3$, we have that $M \models \vec{t}_2 \perp_{\vec{t}_1} \vec{t}_3$ too. But all variables occurring in $\vec{t}_1 \vec{t}_2 \vec{t}_3$ are already in $\text{Dom}(X)$, and therefore

$$M \models X \vec{t}_2 \perp_{\vec{t}_1} \vec{t}_3$$

\[\square\]

**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 $\vec{w} = \text{Dom}(X) \setminus \{\vec{x}, \vec{y}, \vec{z}\}$. Then

$$M \models X \vec{y} \perp_{\vec{x}} \vec{z} \iff M \models X \forall \vec{w}(\vec{x} \rightarrow \vec{y}).$$

**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 $\vec{m}_1, \vec{m}_2 \in \text{Dom}(M)[\vec{w}]$ 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

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

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

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

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

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

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

Now, $\text{Dom}(X) \setminus \{\vec{x}\vec{y}\}$ is precisely $\vec{x}\vec{w}$: 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

$$s''(\vec{x}\vec{z}) = h''(\vec{x}\vec{z}) = h'(\vec{x}\vec{z}) = s'(\vec{x}\vec{z}).$$
Chapter 4. Dependencies in Team Semantics

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 \( \text{TS-} \exists \text{-lax} \) 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 \( \text{TS-} \exists \text{-lax} \) is to be preferred to Rule \( \text{TS-} \exists \) 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
4.3. Inclusion and Exclusion in Logic

considered in that field. Therefore, studying the effect of substituting the dependency 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

---

⁴As 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.

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

⁶Equivalently, 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

\[ \{ \text{Father} \subseteq \text{Person}, \text{Mother} \subseteq \text{Person}, \text{Child of Father} \subseteq \text{Person}, \text{Child of Mother} \subseteq \text{Person} \}. \]

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:

\[ \{ \text{Person} \rightarrow \text{Date of Birth}, \text{Child of Father} \rightarrow \text{Father}, \text{Child of Mother} \rightarrow \text{Mother} \}. \]

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 \( \bar{x} \), \( \bar{x} \subseteq \bar{x} \);
- **I2:** If \( |\bar{x}| = |\bar{y}| = n \) then, for all \( m \in \mathbb{N} \) and all \( \pi : 1 \ldots m \rightarrow 1 \ldots n \),
  \[ \bar{x} \subseteq \bar{y} \vdash x_{\pi(1)} \ldots x_{\pi(m)} \subseteq y_{\pi(1)} \ldots y_{\pi(m)}; \]
- **I3:** For all tuples of attributes of the same length \( \bar{x}, \bar{y}, \bar{z}, \bar{\bar{x}}, \bar{\bar{y}} \), and \( \bar{\bar{z}}, \)
  \( \bar{x} \subseteq \bar{y}, \bar{y} \subseteq \bar{z} \vdash \bar{x} \subseteq \bar{z}. \)

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

\[ \Gamma \vdash \bar{x} \subseteq \bar{y} \]

\footnote{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 \( \bar{x} \subseteq \bar{y} \), so does \( R \cup S \). Since functional dependencies as the above ones are clearly not closed under unions, they cannot be represented by inclusions.}
4.3. Inclusion and Exclusion in Logic

can be derived from the axioms $I_1$, $I_2$ and $I_3$ if and only if all relations which respect all dependencies of $\Gamma$ 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

$$(\text{Person, Date of birth}) \times (\text{Father, Child of Father}) \times (\text{Mother, Child of Mother})$$

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

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

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 $\text{Father} \mid \text{Mother}$.

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 $\pi : 1 \ldots m \rightarrow 1 \ldots n$,

   $$\vec{x} \subseteq \vec{y} \vdash x_{\pi(1)} \ldots x_{\pi(m)} \subseteq y_{\pi(1)} \ldots y_{\pi(m)};$$
Chapter 4. Dependencies in Team Semantics

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

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

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 $\pi : 1 \ldots m \rightarrow 1 \ldots n$,

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

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

3. Axioms for inclusion/exclusion interaction:

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

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

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

$$M \models X \quad \vec{t}_1 \subseteq \vec{t}_2 \text{ if and only if for every } s \in X \text{ there exists a } s' \in X \text{ such that } \vec{t}_1(s) = \vec{t}_2(s');$$

$$M \models X \quad \vec{t}_1 \mid \vec{t}_2 \text{ if and only if for all } s, s' \in X, \vec{t}_1(s) \neq \vec{t}_2(s').$$

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
Inclusion and Exclusion in Logic

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

\[
\text{Opponent}_\text{as White} \subseteq \text{Opponent}_\text{as Black}, \\
\text{Opponent}_\text{as Black} \subseteq \text{Opponent}_\text{as White}.
\]

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

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

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

\[
\text{Winner}_\text{Boxing} \mid \text{Winner}_\text{Chess}.
\]
This is a different, and stronger, condition than the first order expression \( \text{Winner}_{\text{Boxing}} \neq \text{Winner}_{\text{Chess}} \): 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

\[
X = \begin{array}{|c|c|}
\hline
s_0 & \text{Winner}_{\text{Boxing}} \\
T. \text{Dovramadjiev} & V. \text{Anand} \\
\hline
s_1 & \text{Winner}_{\text{Chess}} \\
T. \text{Woolgar} & T. \text{Dovramadjiev} \\
\hline
\end{array}
\]

### 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 \( \text{TS-inc} \) 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 \( \phi \) of Inclusion Logic such that \( M \models_X \exists x \phi \) according to the lax semantics of Rule \( \text{TS-3-Lax} \) but not according to the strict semantics of Rule \( \text{TS-3} \).

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

\[
X = \begin{array}{|c|c|c|}
\hline
s_0 & y & z \\
0 & 1 \\
\hline
\end{array}
\]

and let \( \phi \) be \( y \subseteq x \land z \subseteq x \).

---

\(^8\)Chessboxing is a hybrid sport, in which chess and boxing rounds are alternated.
4.3. Inclusion and Exclusion in Logic

- \( M \models_X \exists x \phi \) according to the lax semantics:
  Let \( H : X \to \text{Parts}(\text{Dom}(M)) \) be such that \( H(s_0) = \{0,1\} \).
  Then
  \[
  X[H/x] = \begin{array}{ccc}
  y & z & x \\
  s_0' & 0 & 1 & 0 \\
  s_1' & 0 & 1 & 1
  \end{array}
  \]
  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 \( \text{Dom}(M) \).
  Then
  \[
  X[F/x] = \begin{array}{ccc}
  y & z & x \\
  s_0' & 0 & 1 & F(s_0)
  \end{array}
  \]
  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 \( \xi \) such that \( M \models_X \exists x \xi \), but for \( X' = X|_{\text{Free}(\exists x \xi)} \) we have that \( M \not\models_{X'} \exists x \xi \) instead.

**Proof.** Let \( \text{Dom}(M) = \{0,1\} \), let \( \xi \) be \( y \subseteq x \land z \subseteq x \), and let

\[
X = \begin{array}{ccc}
  y & z & u \\
  s_0 & 0 & 1 & 0 \\
  s_1 & 0 & 1 & 1
  \end{array}
\]

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

\[
F(s_0) = 0; \\
F(s_1) = 1;
\]
we have that

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

and it is easy to check that this team satisfies \( \xi \). However, the restriction \( X' \) of \( X \) to \( \text{Free}(\exists x \xi) = \{y, z\} \) 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 \( \phi \) be any Inclusion Logic formula, and let \( V \) be a set of variables with \( \text{Free}(\phi) \subseteq V \). Then, for all suitable teams \( X \),

\[
M \models_X \phi \iff M \models_{X,\mathbf{v}} \phi
\]

with respect to the lax interpretation of existential quantification.

**Proof.** The proof is by structural induction on \( \phi \).

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 \( \phi \) 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 \( M \models_X \phi \) 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 \( \phi \).

1. If \( \phi \) is a first order literal, this is obvious.

2. Suppose that \( M \models_X \bar{t}_1 \subseteq \bar{t}_2 \) for all \( i \in I \). Then \( M \models_X \bar{t}_1 \subseteq \bar{t}_2 \). 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 \( s'(\bar{t}_2) = s(\bar{t}_1) \). Since \( X_i \subseteq X \) we then have that \( s' \in X \), as required.
3. Suppose that $M \models_{X_i} \psi \lor \theta$ 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 $M \models_X \psi \lor \theta$, as required.

4. Suppose that $M \models_{X_i} \psi \land \theta$ 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 $M \models_X \psi \land \theta$.

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 $H_i : X_i \to \text{Parts}(\text{Dom}(M)) \setminus \{\emptyset\}$ such that $M \models_{X_i[H_i/x]} \psi$. Then define the function $H : X \to \text{Parts}(\text{Dom}(M)) \setminus \{\emptyset\}$ 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 $\bigcup_{i \in I}(X_i[M/x]) = \bigcup_{i \in I} X_i[M/x] = X[M/x]$, by induction hypothesis $M \models_{X[M/x]} \psi$ and therefore $M \models_X \forall x \psi$, as required.

$$\square$$

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 $M \models_{X_0} \equiv(x)$ and $M \models_{X_1} \equiv(x)$, but $M \not\models_{X_0 \cup X_1} \equiv(x)$.

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 $\overline{t}_1 \subseteq \overline{t}_2$ is equivalent to the Independence Logic formula

$$\phi := \forall v_1 v_2 \exists (\overline{z} \neq \overline{t}_1 \land \overline{z} \neq \overline{t}_2) \lor (v_1 \neq v_2 \land \overline{z} \neq \overline{t}_2) \lor ((v_1 = v_2 \land \overline{z} = \overline{t}_2) \land \overline{z} \perp v_1 v_2)).$$
where \( v_1, v_2 \) and \( \bar{z} \) do not occur in \( \bar{t}_1 \) or \( \bar{t}_2 \) and where \( \bar{z} \perp v_1 v_2 \) is a shorthand for \( \bar{z} \bot_{v_1 v_2} \).

Proof. Suppose that \( M \models_X \bar{t}_1 \subseteq \bar{t}_2 \). Then split the team \( X' = X[M/v_1 v_2 \bar{z}] \) into three teams \( Y, Z \) and \( W \) as follows:

- \( Y = \{ s \in X' : s(\bar{z}) \neq \bar{t}_1(s) \text{ and } s(\bar{z}) \neq \bar{t}_2(s) \} \);
- \( Z = \{ s \in X' : s(v_1) \neq s(v_2) \text{ and } s(\bar{z}) \neq \bar{t}_2(s) \} \);
- \( W = X' \setminus (Y \cup Z) = \{ s \in X' : s(\bar{z}) = \bar{t}_2(s) \text{ or } (s(\bar{z}) = \bar{t}_1(s) \text{ and } s(v_1) = s(v_2)) \} \).

Clearly, \( X' = Y \cup Z \cup W \), \( M \models_Y \bar{z} \neq t_1 \land \bar{z} \neq t_2 \) and \( M \models_Z v_1 \neq v_2 \land \bar{z} \neq t_2 \); hence, if we can prove that

\[
M \models_W ((v_1 = v_2 \lor \bar{z} = \bar{t}_2)) \land \bar{z} \perp v_1 v_2
\]

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 \( s(\bar{z}) = \bar{t}_2 \), since otherwise we would have that \( s \in Z \) instead. Hence, the first conjunct \( v_1 = v_2 \lor \bar{z} = \bar{t}_2 \) 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 \( s''(\bar{z}) = s(\bar{z}) \) and \( s''(v_1 v_2) = s'(v_1 v_2) \). There are two distinct cases to examine:

1. If \( s(\bar{z}) = \bar{t}_2(s) \), consider the assignment

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

by construction, \( s'' \in X' \). Furthermore, since \( s''(\bar{z}) = \bar{t}_2(s) = \bar{t}_2(s'') \), \( s'' \) is neither in \( Y \) nor in \( Z \). Hence, it is in \( W \), as required.

2. If \( s(\bar{z}) \neq \bar{t}_2(s) \) and \( s \in W \), then necessarily \( s(\bar{z}) = \bar{t}_1(s) \) and \( s(v_1) = s(v_2) \).

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

\[
\bar{t}_1(o) = \bar{t}_1(s) = s(\bar{z});
\]

and since \( M \models_X \bar{t}_1 \subseteq \bar{t}_2 \), there also exist an assignment \( o' \in X \) such that

\[
\bar{t}_2(o') = \bar{t}_1(o) = s(\bar{z}).
\]
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

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

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 $\phi$, 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

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

Then consider the assignments

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

and

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

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

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

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

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

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

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

$$\vec{t}_2(s'') = \vec{t}_2(h'') = h''(\vec{z}) = h(\vec{z}) = \vec{t}_1(s).$$
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 into consideration 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
Suppose that \( M \models \psi(\vec{m}_1, \vec{m}_2) \) and \( \vec{m}_1 \) is 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 \( M \models \psi(\vec{m}_1, \vec{m}_2) \) and \( \vec{m}_2 \) is 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

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

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

Conversely, suppose that \( M \models \neg[\text{TC}_{\vec{x}, \vec{y}} \psi](\vec{a}, \vec{b}) \): 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. \( \square \)

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 \( M = \)
Chapter 4. Dependencies in Team Semantics

(\text{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

\[
M \models \exists z (0 \subseteq z \land z \neq e \land \forall w (w \neq S(z)) \lor w \subseteq z)
\]

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 \( X \),

\[
M \models_X \vec{t}_1 \bowtie \vec{t}_2 \iff M \models_X \vec{t}_1 \subseteq \vec{t}_2 \land \vec{t}_2 \subseteq \vec{t}_1.
\]

**Proof.** Obvious. \( \square \)

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

\[
M \models_X \forall u_1u_2 \exists \vec{z} (\vec{t}_2 \bowtie \vec{z} \land (u_1 \neq u_2 \lor \vec{z} = \vec{t}_1))
\]

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 \( X' = X[M/u_1u_2] \), and pick the tuple of functions \( \vec{H} \) used to choose \( \vec{z} \) so that

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

for all \( s \in X' \).

Then, for \( Y = X'[\vec{H}/\vec{z}] \), by definition we have that \( M \models_Y u_1 \neq u_2 \lor \vec{z} = \vec{t}_1 \), 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 \( \vec{t}_2(s) = \vec{z}(h) \).
Now let 0 and 1 be two distinct elements of \( \mathcal{M} \), and consider the assignment \( h' = s[0/u_1][1/u_2][\vec{H}/\vec{z}] \). By construction, \( h' \in \mathcal{Y} \); and furthermore, by the definition of \( \vec{H} \) we have that \( h'(\vec{z}) = \vec{t}_2(s) = \vec{t}_2(h) \), as required.

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

Conversely, suppose that \( \mathcal{M} \models_{\mathcal{X}} \forall u_1 u_2 \exists \vec{z} (\vec{t}_2 
Rightarrow \vec{z} \wedge (u_1 \neq u_2 \vee \vec{z} = \vec{t}_1)) \), and that therefore there exists a tuple of functions \( \vec{H} \) such that, for \( \mathcal{Y} = \mathcal{X}[\mathcal{M}/u_1 u_2][\vec{H}/\vec{z}] \), \( \mathcal{M} \models_{\mathcal{Y}} \vec{t}_2 \nRightarrow \vec{z} \wedge (u_1 \neq u_2 \vee \vec{z} = \vec{t}_1) \). Then consider any assignment \( s \in \mathcal{X} \), and let \( h = s[0/u_1][1/u_2][\vec{H}/\vec{z}] \). Now, \( h \in \mathcal{Y} \) and \( h(\vec{z}) = \vec{t}_1(s) \); but since \( \mathcal{M} \models_{\mathcal{Y}} \vec{t}_2 \nRightarrow \vec{z} \), this implies that there exists an assignment \( h' \in \mathcal{Y} \) such that \( \vec{t}_2(h') = h(\vec{z}) = \vec{t}_1(s) \). Finally, \( h' \) derives from some assignment \( s' \in \mathcal{X} \), and for this assignment we have that \( \vec{t}_2(s) = \vec{t}_2(h') = \vec{t}_1(s) \) 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 \( \vec{t}_1 \nmid \vec{t}_2 \), 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 \( \phi \) such that

\[
\mathcal{M} \models_{\mathcal{X}} \phi \iff \mathcal{M} \models_{\mathcal{X}} \vec{t}_1 \nmid \vec{t}_2
\]

for all suitable models \( \mathcal{M} \) and teams \( \mathcal{X} \).

**Proof.** This follows immediately from Theorem 2.2.14, since the satisfaction condition for the exclusion atom is downwards monotone and expressible in \( \Sigma^1_1 \).
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 \( \vec{\varepsilon} \) 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 \( M \models X \vec{t}_1 \models \vec{t}_2 \) if and only if

\[
M \models X \forall \vec{\varepsilon} \exists u_1 u_2 ((\vec{\varepsilon}, u_1) \land (\vec{\varepsilon}, u_2) \land ((u_1 = u_2 \land \vec{\varepsilon} \neq \vec{t}_1) \lor (u_1 \neq u_2 \land \vec{\varepsilon} \neq \vec{t}_2))).
\]

Indeed, suppose that \( M \models X \vec{t}_1 \models \vec{t}_2 \), let \( X' = X[M/\vec{\varepsilon}] \), and let 0, 1 be two distinct elements in \( \text{Dom}(M) \).

Then define the functions \( H_1 \) and \( H_2 \) as follows:

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

Then, for \( Y = X'[H_1 H_2/u_1 u_2] \), we have that \( M \models Y = (\vec{\varepsilon}, u_1) \) and that \( M \models Y = (\vec{\varepsilon}, u_2) \), 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 \( \vec{\varepsilon} \).

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 \( M \models Y_2 u_1 = u_2 \); hence, we only need to verify that \( M \models Y_1 \vec{\varepsilon} \neq \vec{t}_1 \) and that \( M \models Y_2 \vec{\varepsilon} \neq \vec{t}_2 \).

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

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

Conversely, suppose that

\[
M \models X \forall \vec{\varepsilon} \exists u_1 u_2 ((\vec{\varepsilon}, u_1) \land (\vec{\varepsilon}, u_2) \land ((u_1 = u_2 \land \vec{\varepsilon} \neq \vec{t}_1) \lor (u_1 \neq u_2 \land \vec{\varepsilon} \neq \vec{t}_2))).
\]

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

\[
M \models Y = (\vec{\varepsilon}, u_1) \land (\vec{\varepsilon}, u_2) \land ((u_1 = u_2 \land \vec{\varepsilon} \neq \vec{t}_1) \lor (u_1 \neq u_2 \land \vec{\varepsilon} \neq \vec{t}_2)).
\]
4.3. Inclusion and Exclusion in Logic

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 $\vec{t}_1(s_1) \neq \vec{t}_2(s_2)$. Suppose instead that $\vec{t}_1(s_1) = \vec{t}_2(s_2) = \vec{m}$ for some tuple of elements $\vec{m}$, and consider two assignments $h_1, h_2$ such that

$$h_1 \in \{s_1[\vec{m}/\vec{z}]\}[H_1/H_2/u_1u_2],$$

and

$$h_2 \in \{s_2[\vec{m}/\vec{z}]\}[H_1/H_2/u_1u_2].$$

Then $h_1, h_2 \in Y$; and furthermore, since $h_1(\vec{z}) = h_2(\vec{z})$ and $M \models (\vec{z}, u_1) \land = (\vec{z}, u_2)$, 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, $M \models_y (u_1 = u_2 \land \vec{z} \neq \vec{t}_1) \lor (u_1 \neq u_2 \land \vec{z} \neq \vec{t}_2)$, and therefore $Y$ can be split into two subteams $Y_1$ and $Y_2$ such that

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

and

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

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

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

and therefore $h_1$ cannot be in $Y_1$, and because

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

and therefore $h_2$ cannot be in $Y_2$.

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

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

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

for all suitable models $M$ and teams $X$.

**Proof.** Suppose that $M \models X = (t_1 \ldots t_n)$, and consider the team $X[M/z]$. Now, let $Y = \{s \in X[M/z] : s(z) = t_n(s)\}$ and let $Z = X[M/z] \setminus Y$.

---

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 
\( Z \models t_1 \ldots t_{n-1} z | t_1 \ldots t_{n-1} t_n \) we can conclude that \( M \models_X \phi \), as required.

Now, consider any two \( s, s' \in Z \), and suppose that \( t_i(s) = t_i(s') \) for all \( i = 1 \ldots n-1 \). But then \( s(z) \neq t_n(s') \); 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 \( t_n(s) = t_n(s') \).

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

So \( s(z) \neq t_n(s') \), and since this holds for all \( s \) and \( s' \) in \( Z \) which coincide over \( t_1 \ldots t_{n-1} \) we have that
\[
M \models_Z t_1 \ldots t_{n-1} z | t_1 \ldots t_{n-1} t_n,
\]
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 \( \phi \), \( X[M/z] \) can be split into two sub-teams \( Y \) and \( Z \) such that \( M \models_Y z = t_n \) and 
\[
M \models_Z (t_1 \ldots t_{n-1} z | t_1 \ldots t_{n-1} t_n).
\]

Now, suppose that \( t_n(s) = m \) and \( t_n(s') = m' \), 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 \( t_i(s) = t_i(s') \) for all \( i = 1 \ldots n-1 \),
\[
t_n(s') = m' = s[m'/z](z) \neq t_n(s'[m/z]) = t_n(s)
\]
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.
4.3. Inclusion and Exclusion in 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 \( \phi \) there exists an Independence Logic formula \( \phi^* \) such that

\[
M \models_X \phi \iff M \models_X \phi^*
\]

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 \( \phi \) be the formula

\[
\forall \vec{p}\vec{q}\vec{r} \exists u_1 u_2 u_3 u_4 \left( \bigwedge_{i=1}^4 = (\vec{p} \vec{q} \vec{r}, u_i) \land ((u_1 \neq u_2 \land (\vec{p} \vec{q} | \vec{t}_1 \vec{t}_2)) \lor \right)
\]

\[
\lor (u_1 = u_2 \land u_3 \neq u_4 \land (\vec{p} \vec{q} | \vec{t}_1 \vec{t}_2)) \lor (u_1 = u_2 \land u_3 = u_4 \land (\vec{p} \vec{q} \vec{r} \subseteq \vec{t}_1 \vec{t}_2 \vec{t}_3)))
\]

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 \( X \),

\[
M \models_X \vec{t}_2 \perp_{\vec{t}_1} \vec{t}_3 \iff M \models_X \phi.
\]

**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 \( s \in X'[H_1/u_1] \),

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

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

\[
H_4(s) = \begin{cases} 
\{0\} & \text{if there exists a } s' \in X \text{ such that } \vec{t}_1(s')\vec{t}_3(s') = s(\vec{p})s(\vec{r}); \\
\{1\} & \text{otherwise}.
\end{cases}
\]
Chapter 4. Dependencies in Team Semantics

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:

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

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

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

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

$$M \models_{Y_1} u_1 \neq u_2 \land (\vec{pq} \mid \vec{t}_1\vec{t}_2),$$

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

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

and therefore

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

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

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

and

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

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

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

Now, let $o$ be the assignment of $Y$ given by

$$o = h[\vec{t}_1(h)\vec{t}_2(h)\vec{t}_3(h)/\vec{pq}]\cdot[H_1 \ldots H_4/u_1 \ldots u_4]:$$
4.3. Inclusion and Exclusion in Logic

by the definitions of $H_1 \ldots H_4$ 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,

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

and hence

$$M \models \overline{t}_1\overline{t}_2\overline{t}_3$$

as required.

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

$$h = s[\overline{t}_1(s)/\overline{p}[\overline{t}_2(s)/\overline{q}][\overline{t}_3(s')/\overline{r}]
$$

and

$$h' = s'[\overline{t}_1(s)/\overline{p}[\overline{t}_2(s)/\overline{q}][\overline{t}_3(s')/\overline{r}].$$

Now, since $M \models X \phi$, there exist functions $H_1 \ldots H_4$, depending only on $\overline{p}, \overline{q}$ and $\overline{r}$, 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

$$M \models Y_1 (u_1 \neq u_2 \lor (\overline{p}\overline{q} \mid \overline{t}_1\overline{t}_2));$$

$$M \models Y_2 (u_1 = u_2 \land u_3 \neq u_4 \land (\overline{p}\overline{r} \mid \overline{t}_1\overline{t}_3));$$

$$M \models Y_3 (u_1 = u_2 \land u_3 = u_4 \land (\overline{p}\overline{q}\overline{r} \subseteq \overline{t}_1\overline{t}_2\overline{t}_3)).$$

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 $\overline{p}\overline{q}\overline{r}$ 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 $u_1 \ldots u_4$, and therefore that they belong to the same $Y_i$.

But they cannot be in $Y_1$ nor in $Y_2$, since

$$o(\overline{p})o(\overline{q}) = o'(\overline{p})o'(\overline{q}) = \overline{t}_1(s)\overline{t}_2(s) = \overline{t}_1(o)\overline{t}_2(o).$$
Chapter 4. Dependencies in Team Semantics

and

\[ o(\vec{p})o(\vec{r}) = o'(\vec{p})o'(\vec{r}) = \vec{t}_1(s')\vec{t}_3(s') = \vec{t}_1(o')\vec{t}_3(o'); \]

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

\[ \vec{t}_1(o'')\vec{t}_2(o'')\vec{t}_3(o'') = o(\vec{p})o(\vec{q})o(\vec{r}) = \vec{t}_1(s)\vec{t}_2(s)\vec{t}_3(s') \]

and, finally, there exists a \( s'' \in X \) such that \( \vec{t}_1(s'')\vec{t}_2(s'')\vec{t}_3(s'') = \vec{t}_1(s)\vec{t}_2(s)\vec{t}_3(s') \), 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 \( \phi \) be any I/E Logic formula and let \( V \) be a set of variables such that \( \text{Free}(\phi) \subseteq V \). Then, for all suitable teams \( X \),

\[ M \models_X \phi \iff M \models_{X|V} \phi \]

**Proof.** The proof is by structural induction on \( \phi \).

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

2. Let \( \phi \) be of the form \( \psi \lor \theta \), and suppose that \( M \models_X \psi \lor \theta \). 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, \( M \models_{Y|V} \psi \) and \( M \models_{Z|V} \theta \). But \( X|V = Y|V \cup Z|V \): indeed, \( s \in X \) if and only if \( s \in Y \) or \( s \in Z \), and hence \( s|V \in X|V \) if and only if it is in \( Y|V \) or \( Z|V \). Hence, \( M \models_{X|V} \psi \lor \theta \), as required.

Conversely, suppose that \( M \models_{X|V} \psi \lor \theta \), that is, that \( X|V = Y' \cup Z' \) 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.
4.3. Inclusion and Exclusion in Logic

First Order (Team) Logic

\[
\text{Inclusion Logic, Equiextension Logic, I/E Logic, Dependence Logic, Exclusion Logic, Constancy Logic, Independence Logic, Multivalued Dependence Logic, I/E Logic}
\]

Figure 4.2: Relations between logics of imperfect information (wrt formulas)
define $Y = \{s \in X : s|_V \in Y'\}$ and $Z = \{s \in X : s|_V \in Z'\}$. Now, $X = Y \cup Z$: indeed, if $s \in X$ then $s|_V$ is in $X|_V$, 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, $Y|_V = Y'$ and $Z|_V = Z'$,\(^{11}\) and hence by induction hypothesis $M \models_Y \psi$ and $M \models_Z \theta$, and finally $M \models_X \psi \lor \theta$.

3. Let $\phi$ be of the form $\psi \land \theta$. Then $M \models_X \psi \land \theta$ if and only if $M \models_X \psi$ and $M \models_X \theta$, that is, by induction hypothesis, if and only if $M \models_X s_1 \psi$ and $M \models_X s_2 \theta$. But this is the case if and only if $M \models_X s_1 \psi \land s_2 \theta$, as required.

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

Now consider the function $H' : X|_V \to \text{Parts}(\text{Dom}(M)) \setminus \{\emptyset\}$ which assigns to every $s' \in X|_V$ the set
\[
H'(s') = \bigcup \{H(s) : s \in X, s' = s|_V\}.
\]

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

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

5. For all suitable teams $X$, $X[M/x]|_{\cup(x)} = X|_V[M/x]$; and hence, $M \models_{X|_V} \forall x \psi \iff M \models_{X[M/x]|_{\cup(x)}} \psi \iff M \models_{X[M/x]} \psi \iff M \models_{X} \forall x \psi$, as required.

---

\(^{11}\)By definition, $Y|_V \subseteq Y'$ and $Z|_V \subseteq Z'$. On the other hand, if $s' \in Y'$ then $s' \in X|_V$, and hence $s'$ is of the form $s|_V$ for some $s \in X$, and therefore this $s$ is in $Y$ too, and finally $s' = s|_V \in Y|_V$. The same argument shows that $Z' \subseteq Z|_V$.

\(^{12}\)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 $s|_V \in X|_V$, and moreover $m \in H'(s|_V)$ by the definition of $H'$, and hence $s'|_{\cup(x)} = s|_V[m/x] \in X|_V[H'/x]$.

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

\(^{13}\)In brief, for all $s \in X$ and all $m \in \text{Dom}(M)$ we have that $m \in H'(s|_V)$ if and only if $m \in H(s)$, by definition. Hence, for all such $s$ and $m$, $s|_V[m/x] \in X|_V[H'/x]$ 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

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

The uniformity condition will be changed in the obvious way:

**Definition 4.4.1.** Let \( G^M_X(\phi) \) 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 \( \vec{t}_1(s) = \vec{t}_2(s') \);

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, \( \vec{t}_1(s) \neq \vec{t}_2(s') \).

The other modification which we need to make, in order to account for the \( \text{TS-}\exists\text{-lax} \) rule, is that we must now be able to consider nondeterministic strategies:

**Definition 4.4.2.** Let \( G^M_X(\phi) \) be a semantic game and let \( \psi \) be any expression such that \( (\psi, s') \) is a possible position for some \( s' \). Then a nondeterministic local strategy for \( \psi \) is a function \( f_\psi \) sending each \( s' \) into a nonempty subset of \( \text{Succ}_M(\psi, s') \).

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

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

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 $\phi$ 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 $\phi$, 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:

1. If $\phi$ is a disjunction $\psi_1 \lor \psi_2$ 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 $f_1$ 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 $G^M_X(\psi_1 \lor \psi_2)$ as follows:
   - If $\theta$ is part of $\psi_1$ then $f_\theta = (f_1)_\theta$;
   - If $\theta$ is part of $\psi_2$ then $f_\theta = (f_2)_\theta$;
   - If $\theta$ is the initial formula $\psi_1 \lor \psi_2$ then
     
     $f_\theta(s) = \begin{cases} 
     \{(\psi_1, s), (\psi_2, s)\} & \text{if } s \in X_1 \cap X_2; \\
     \{(\psi_1, s)\} & \text{if } s \in X_1 \setminus X_2; \\
     \{(\psi_2, s)\} & \text{if } s \in X_2 \setminus X_1;
     \end{cases}$

   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 $G^M_X(\psi_1 \lor \psi_2)$ 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 nondeterministic uniform winning strategy for $E$ in $G^M_X(\psi)$. Now let $X_1 = \{ s \in X : (\psi_1, s) \in f_\theta(s) \}$, let $X_2 = \{ s \in X : (\psi_2, s) \in f_\theta(s) \}$, and let $f_1$ and $f_2$ be the restrictions of $f$ to the subgames corresponding to $\psi_1$ and $\psi_2$ respectively. Then $f_1$ and $f_2$ are uniform and

\[ \text{Note that here it is vital that all possible plays of } G^M_{X_1}(\psi_1) \text{ in which } E \text{ follows } f_1 \text{ are part of some possible play of } G^M_X(\psi) \text{ in which } E \text{ follows } f. \] Otherwise, it would not be guaranteed that the uniformity conditions corresponding to inclusion atoms are respected.
4.4. Game Theoretic Semantics for I/E Logic

winning for $G^M_X(\psi_1)$ and $G^M_X(\psi_2)$ respectively, and hence by induction hypothesis $M \models_X \psi_1$ and $M \models_X \psi_2$. But $X = X_1 \cup X_2$, and hence this implies that $M \models_X \phi$.

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

- If $\theta$ is part of $\psi$ then $f_\theta = g_\theta$;
- $f_{\exists v \psi}(s) = \{(\psi, s[m/v]) : m \in H(s)\}$.

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

Conversely, suppose that $E$ has a nondeterministic uniform winning strategy $f$ for $G^M_X(\exists v \psi)$. Then define the function $H : X \to \text{Parts}(\text{Dom}(M)) \setminus \{\emptyset\}$ so that for all $s \in X$, $f_\psi(\exists v \psi, s) = \{(\psi, s[m/v]) : m \in H(s)\}$, and let $g$ be the restriction of $f$ to $\psi$. Then $g$ 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 descendants 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

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

for all suitable models $M$ and teams $X$.

**Proof.** The proof is an unproblematic induction over the formula $\phi$, 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.\textsuperscript{15}

**Theorem 4.5.2.** Let $\Phi(A)$ be a formula in $\Sigma_1^1$ such that $\text{Free}(\Phi) = \{A\}$, and let $\vec{v}$ be a tuple of distinct variables with $|\vec{v}| = \text{Arity}(A)$. Then there exists an

\textsuperscript{15}The details of this proof are similar to the ones of [50] and [49].
4.5. Definability in I/E Logic (and in Independence Logic)

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

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

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

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

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

$$\exists \vec{f}_1 \forall \vec{v}_1 \exists \vec{z}_1 \left( \left( \bigwedge_i = (\vec{w}_i, z_i) \land ((\vec{v}_i \subseteq \vec{x} \land z_1 = z_2) \lor (\vec{v}_i \land z_1 \neq z_2)) \land \psi'(\vec{x}, \vec{y}, \vec{z}) \right) \right),$$

where $\vec{f}_1 = f_1 f_2 \ldots f_n$, $\psi'(\vec{x}, \vec{y}, \vec{z})$ 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

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

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 $M \models_X \phi(\vec{v}) \iff M \models \Phi^*(\text{Rel}_\vec{v}(X))$:

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

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

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

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 $s \in X[M/\vec{x}y]$ with $s(\vec{w}_i) = \vec{m}$, and let $o$ be any assignment with domain $\vec{y}$. Thus, if we can prove that $M \models_o (\text{Rel}_\vec{v}(X)) \vec{x} \leftrightarrow f_1(\vec{x}) = f_2(\vec{x}) \land \psi(\vec{x}, \vec{y}, \vec{f})$ then the left-to-right direction of our proof is done.

First of all, suppose that $M \models_o (\text{Rel}_\vec{v}(X)) \vec{x}$, 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 $h(\vec{r}) = s(\vec{v}) = \vec{m} = h(\vec{x})$, 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 $\mathcal{M}, \not\models (\text{Rel}_{\vec{v}}(X)) \vec{x}$, 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 $h'(\vec{v}) = s(\vec{v}) = \vec{m}$; 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

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

for all assignments $o$ with domain $\vec{x}\vec{y}$, and we still need to verify that $\mathcal{M} \models_o \psi(\vec{x}, \vec{y}, f)$ 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 $\mathcal{M} \models X[M/\vec{x}\vec{y}][\vec{H}/\vec{z}] \psi'(\vec{x}, \vec{y}, \vec{z})$ and $\psi'(\vec{x}, \vec{y}, \vec{z})$ is first order, $\mathcal{M} \models (h) \psi'(\vec{x}, \vec{y}, \vec{z})$; but $\psi'(\vec{x}, \vec{y}, f(\vec{x}\vec{y}))$ is equivalent to $\psi(\vec{x}, \vec{y}, f)$ and $h(z) = f(h(\vec{w}_i)) = f(o(\vec{w}_i))$, and therefore

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

as required.

Conversely, suppose that $\mathcal{M} \models_s (\text{Rel}_{\vec{v}}(X)) \vec{x} \iff (f_1(\vec{x}) = f_2(\vec{x})) \land \psi(\vec{x}, \vec{y}, f)$

for all assignments $s$ with domain $\vec{x}\vec{y}$ and for some fixed choice of the tuple of functions $f$.

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

Clearly, $Y$ satisfies the dependency conditions; furthermore, it satisfies $\psi'(\vec{x}, \vec{y}, \vec{z})$, as required.
4.5. Definability in I/E Logic (and in Independence Logic)

because for every assignment $h \in Y$ and every $i \in 1 \ldots n$ we have that $H_i(h) = \{h(z_i)\} = \{f_i(h(\vec{\bar{w}}))\}$.

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 $M \models_{Y_1} \vec{z}_1 = \vec{z}_2$ and $M \models_{Y_2} \vec{z}_1 \neq \vec{z}_2$, 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 $o(z_1) = o(z_2)$, $f_1(o(\vec{v})) = f_2(o(\vec{v}))$.

This implies that $o(\vec{x}) \in \text{Rel}_{\vec{v}}(X)$, and hence that there exists an assignment $s' \in X$ with $s'(\vec{v}) = o(\vec{x})$.

Now consider the assignment

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

since in $Y$ the values of $\vec{z}$ depend only on the values of $\vec{x}$ 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, $o(\vec{x}) \not\in \text{Rel}_{\vec{v}}(X)$, 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. \qed

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 $\text{Free}(\Phi) = A$, and let $\vec{v}$ be any set of variables such that $|\vec{v}| = \text{Arity}(A)$. Then there exists an Independence Logic formula $\phi(\vec{v})$ such that

\[ M \models_X \phi(\vec{v}) \iff M \models \Phi(\text{Rel}_{\vec{v}}(X)) \]

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 $t(s) \neq t(s')$.

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 $0 \lor \sim = (t)$, 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 $x_1 \ldots x_n$ be variables, and let $t$ be a term. Then we will write $\neq(x_1 \ldots x_n, t)$ for $\delta^3x_1 \ldots \delta^3x_n \neq(t)$.

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

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

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. \[\square\]

It is worth observing that $\neq(\vec{t}, t')$ is not equivalent to the contradictory negation $\sim\neq(\vec{t}, t')$ 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” $\neq(t_1 \ldots t_n, t')$ 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 $\exists v(v \neq t' \land \vec{t}v \subseteq \vec{t}')$.

Proof. Obvious. \[\square\]

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

$$\forall v_1 v_2 v_3 ((v_1 = v_2) \lor (v_1 \neq v_2 \land v_1 \neq v_3 \land v_2 \neq v_3) \lor ((v_3 = v_2 \land v_2 \neq v_2) \lor (v_3 \neq v_1 \land v_3 = v_2)) \land \exists \vec{u} ((v_3 \neq v_1 \lor \vec{u} = \vec{t}_1) \land (v_3 \neq v_2 \lor \vec{u} = \vec{t}_2) \land (\neq(\vec{u}v_1v_2v_3))).$$
4.6. Announcements, Constancy Atoms, and Inconstancy Atoms

\textbf{Proof.} Suppose that }\mathcal{M} \models X \mathcal{I}_1 \models \mathcal{I}_2, \text{ that is, that } X(\mathcal{I}_1) = X(\mathcal{I}_2), \text{ let } Y = X[M/v_1v_2v_3], \text{ and let}

\begin{itemize}
  \item \( Y_1 = \{ s \in Y : s(v_1) = s(v_2) \}; \)
  \item \( Y_2 = \{ s \in Y : s(v_1), s(v_2), \text{ and } s(v_3) \text{ are all different} \}; \)
  \item \( Y_3 = Y \setminus (Y_1 \cup Y_2). \)
\end{itemize}

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

Furthermore, let \( \bar{H} \) be such that

\[ \bar{H}(s) = \begin{cases} \bar{I}_1(s) & \text{if } s(v_3) = s(v_1); \\ \bar{I}_2(s) & \text{if } s(v_3) = s(v_2) \end{cases} \]

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

1. If \( h(v_3) = h(v_1) \), then \( h(\bar{u}) = \bar{I}_1(h) = \bar{I}_1(s) \) for some \( s \in X \). Since \( X(\bar{I}_1) = X(\bar{I}_2), \) there exists a \( s' \in X \) with \( \bar{I}_1(s) = \bar{I}_2(s') \) Now consider \( h' = s'[h(v_1)/v_1][h(v_2)/v_2][h(v_2)/v_3][\bar{H}/\bar{u}] \in Z,^{16} \) by the definition of \( \bar{H} \).

\[ h'(\bar{u}) = \bar{I}_2(h') = \bar{I}_1(h) = h(\bar{u}), \]

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 \( h(\bar{u}) = \bar{I}_2(h) = \bar{I}_2(s) \) for some \( s \in X \). Since \( X(\bar{I}_1) = X(\bar{I}_2), \) there exists a \( s' \in X \) with \( \bar{I}_1(s) = \bar{I}_2(s') \) Now consider \( h' = s'[h(v_1)/v_1][h(v_2)/v_2][h(v_2)/v_3][\bar{H}/\bar{u}] \in Z, \) by the definition of \( \bar{H} \).

\[ h'(\bar{u}) = \bar{I}_1(h') = \bar{I}_2(h) = h(\bar{u}), \]

and furthermore \( h \) and \( h' \) coincide over \( v_1 \) and \( v_2 \), but they do not coincide over \( v_3 \).

Therefore, \( \mathcal{M} \models Z \neq (\bar{u}v_1v_2v_3) \), 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 \( \bar{H} \) such that, for \( Z = Y_3[\bar{H}/\bar{u}], \) \( \mathcal{M} \models Z (v_3 \neq v_1 \vee \bar{u} = \bar{I}_1) \wedge (v_3 \neq v_2 \vee \bar{u} = \bar{I}_2) \wedge \neq (\bar{u}v_1v_2v_3) \), and this implies that \( \bar{H} \) is also necessarily as we stated before. Now pick any \( s \in X \), and let \( a, b \in \text{Dom}(M) \) be such that \( a \neq b. \)

\(^{16}\text{Strictly speaking, this expression defines a set of assignments of size one. The assignment } h' \text{ is then chosen as its unique element; and it is in } Z \text{ because, by definition, } s'[h(v_1)/v_1][h(v_2)/v_2][h(v_2)/v_3] \text{ is in } Y_3. \)
1. Consider $h = s[a/v_1][b/v_2][a/v_3][\vec{t}_1(s)/\vec{u}] \in Z$. 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 $h'(\vec{u}) = \vec{t}_2(h') = \vec{t}_2(s')$ for some $s' \in X$. Hence, there exists a $s' \in X$ with $\vec{t}_2(s') = h'(\vec{u}) = h(\vec{u}) = \vec{t}_1(s)$.

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

Hence, $M \models X \not\models \vec{t}_1 \bowtie \vec{t}_2$, 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.** $\mathcal{FO}(\delta^1, \#(\cdot))$ 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. 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.