Some denotationally indistinguishable operators

Pietro Galliani

Institute for Logic, Language and Computation
Universiteit van Amsterdam

ESSLLI 2011
Workshop on Logical Constants
A (very strange) question

Let $L$ be a logic, and let $R_1$ and $R_2$ be two alternative semantic rules for the same operator. Suppose furthermore that these two rules lead to the exact same satisfaction relation - they are denotationally equivalent in $L$. Which one should we prefer?

A (very wrong?) possible answer

Who cares? They are exactly the same!
A (very strange) question

Let $\mathcal{L}$ be a logic, and let $R_1$ and $R_2$ be two alternative semantic rules for the same operator. Suppose furthermore that these two rules lead to the exact same satisfaction relation - they are *denotationally equivalent* in $\mathcal{L}$. Which one should we prefer?

A (very wrong?) possible answer

Who cares? They are exactly the same!
**Logics, Logic Families, and Semantic Rules**

**A (very imprecise) alternative answer**

The one that leads to “better” semantics for the “interesting” extensions/variants of the logic.

**A (very general) idea**

Logical operators and their semantics should not be studied in the context of a single, isolated logic. Rather, they should be studied in the context of a “cloud” of related formalisms - extensions, variants, and so on. For example: Consequence Mining, Bonnay and Westerståhl.

**A (very unanswered) objection**

What is a “better” semantics, or an “interesting” extension?
A (very imprecise) alternative answer

The one that leads to “better” semantics for the “interesting” extensions/variants of the logic.

A (very general) idea

Logical operators and their semantics should not be studied in the context of a single, isolated logic. Rather, they should be studied in the context of a “cloud” of related formalisms - extensions, variants, and so on. For example: *Consequence Mining*, Bonnay and Westerståhl.

A (very unanswered) objection

What is a “better” semantics, or an “interesting” extension?
A (very imprecise) alternative answer

The one that leads to “better” semantics for the “interesting” extensions/variants of the logic.

A (very general) idea

Logical operators and their semantics should not be studied in the context of a single, isolated logic. Rather, they should be studied in the context of a “cloud” of related formalisms - extensions, variants, and so on. For example: Consequence Mining, Bonnay and Westerståhl.

A (very unanswered) objection

What is a “better” semantics, or an “interesting” extension?
**Some motivation**

1. *Logics of Imperfect Information* are a family of logical formalisms which allow to reason about (rather general) patterns of dependence and independence between connectives.

2. First example: *Branching Quantifier Logic* (Henkin 1961)

\[
\left( \forall x \exists y \forall z \exists w \right) \phi(x, y, z, w).
\]

Semantics can be given in terms of skolemizations, or in terms of games.
Some motivation

3. Some of the main developments: IF Logic (Hintikka, Sandu 1989), Hodges Semantics (Hodges 1997), Dependence Logic (Väänänen 2007). “Notational” variations have deep effects!

4. One of the main research trends: develop new variants, add new operators to the language, and compare properties.

5. Apart from questions about expressive power, what can we ask? What are “reasonable” or “useful” properties for an operator or a logic to have?
Outline

1. Tarskian Interpreted Languages and Locality
   - Tarskian Interpreted Languages
   - Locality

2. Logics of Imperfect Information
   - Team Semantics
   - Dependence Logic and IF Logic

3. Independence Logic
   - Independence Logic
   - Nonlocality
   - Lax and Strict Semantics

Some denotationally indistinguishable operators
Outline

1. Tarskian Interpreted Languages and Locality
   - Tarskian Interpreted Languages
   - Locality

2. Logics of Imperfect Information
   - Team Semantics
   - Dependence Logic and IF Logic

3. Independence Logic
   - Independence Logic
   - Nonlocality
   - Lax and Strict Semantics
Tarskian interpreted language (Bonnay, Westerståhl 2010)

$L = \langle \text{Symb}_L, \text{Expr}_L, I_L \rangle$, where

1. $\text{Symb}_L$ is a set of *symbols*;
2. $\text{Expr}_L$ is a set of *well-formed expressions* ("sentences");
3. $I_L \subseteq I_L$ is a $L$-interpretation, sending each $u \in \text{Symb}_L$ of category $C$ to a semantic value $I(u) \in S_C$.

**Truth Interpretation**

$\models_L \subseteq I_L \times \text{Expr}_L$. We write $I \models_L \phi$ instead of $(I, \phi) \in \models_L$.
First Order Logic

Let $M$ be a first order model for some fixed signature $\Sigma$, and let $s$ be an assignment over the variables $V$. Then $FO(M, s) = \langle \text{Symb}_{FO(M, s)}, \text{Expr}_{FO(M, s)}, I_{FO(M, s)} \rangle$, where

- $\text{Symb}_{FO(M, s)} = \text{Connectives} \cup \Sigma \cup V$;
- $\text{Expr}_{FO(M, s)} = \text{well-formed formulas}$;
- $I_{FO(M, s)}$ maps connectives to their rules, the signature symbols to their interpretations, and the variables to their values.

Satisfaction in First Order Logic

$I_{FO(M, s)} \models FO \phi \iff M \models_s \phi$ in the usual sense.
Outline

1. Tarskian Interpreted Languages and Locality
   - Tarskian Interpreted Languages
   - Locality

2. Logics of Imperfect Information
   - Team Semantics
   - Dependence Logic and IF Logic

3. Independence Logic
   - Independence Logic
   - Nonlocality
   - Lax and Strict Semantics
Locality

If \( \rho \) is a replacement such that for all \( u \) occurring in \( \phi \), \( I(u) = J(\rho(u)) \), then

\[
I \models \phi \text{ if and only if } J \models \phi[\rho].
\]

Bonnay, Westerståhl 2010:

*Locality means that the question whether a sentence is true or not depends only on the semantic values of its symbols.*
What does this mean?

- If one changes the interpretation of a variable which does not occur in $\phi$, the meaning of $\phi$ does not vary;
- If one changes the interpretation of a symbol which does not occur in $\phi$, the meaning of $\phi$ does not vary;
- If one changes the interpretation of a connective which does not occur in $\phi$, the meaning of $\phi$ does not vary.
What does this mean?

- If one changes the interpretation of a variable which does not occur in $\phi$, the meaning of $\phi$ does not vary;
- If one changes the interpretation of a symbol which does not occur in $\phi$, the meaning of $\phi$ does not vary;
- If one changes the interpretation of a connective which does not occur in $\phi$, the meaning of $\phi$ does not vary.
Locality

What does this mean?

- If one changes the interpretation of a variable which does not occur in $\phi$, the meaning of $\phi$ does not vary;
- If one changes the interpretation of a symbol which does not occur in $\phi$, the meaning of $\phi$ does not vary;
- If one changes the interpretation of a connective which does not occur in $\phi$, the meaning of $\phi$ does not vary.
Example

Suppose that $I \models_{FO} Px \lor Qy$.
Now let $J$ be similar to $I$, except for:

1. $J(x) = I(y)$;
2. $J(z) = I(x)$;
3. $J(\forall) = \text{“there exist uncountably many . . .”}$.

Then $J \models_{FO} Pz \lor Qy$. 
Some observations

1. Locality is a very natural requirement for logics and/or families of logics;

2. If two logical formalisms are otherwise equivalent, but one respects locality and the other does not, the one which does is probably easier to work with;

3. If two semantic rules are equivalent for a connective in our logic \( L \), but only one of them respects locality in certain “reasonable” expansions of it, it is probably a good idea to favor the one which does.
Some observations

1. Locality is a very natural requirement for logics and/or families of logics;

2. If two logical formalisms are otherwise equivalent, but one respects locality and the other does not, the one which does is probably easier to work with;

3. If two semantic rules are equivalent for a connective in our logic $L$, but only one of them respects locality in certain “reasonable” expansions of it, it is probably a good idea to favor the one which does.
Outline

1. Tarskian Interpreted Languages and Locality
   - Tarskian Interpreted Languages
   - Locality

2. Logics of Imperfect Information
   - Team Semantics
   - Dependence Logic and IF Logic

3. Independence Logic
   - Independence Logic
   - Nonlocality
   - Lax and Strict Semantics
### Tarski’s Semantics for First Order Logic

**Tarski Semantics (assuming Negation Normal Form)**

- \( M \models_s R \vec{t} \) if and only if \( \vec{t}(s) \in R^M \);
- \( M \models_s \neg R \vec{t} \) if and only if \( \vec{t}(s) \notin R^M \);
- \( M \models_s t_1 = t_2 \) if and only if \( t_1(s) = t_2(s) \);
- \( M \models_s t_1 \neq t_2 \) if and only if \( t_1(s) = t_2(s) \);
- \( M \models_s \phi \land \psi \) if and only if \( M \models_s \phi \) and \( M \models_s \psi \);
- \( M \models_s \phi \lor \psi \) if and only if \( M \models_s \phi \) or \( M \models_s \psi \);
- \( M \models_s \exists x \phi \) if and only if \( \exists m \in \text{Dom}(M) \) s.t. \( M \models_s[m/x] \phi \);
- \( M \models_s \forall x \phi \) if and only if \( \forall m \in \text{Dom}(M), M \models_s[m/x] \phi \).
Assignments as States of Things

In Tarski’s semantics, an assignment $s$ is a possible state of things. $M \models_s \phi$ if and only if $\phi$ is true of the state $s$.

Teams (= Belief Sets)

A team $X$ is a set of assignments (possible states of things).

Knowledge is Satisfaction

$M \models_X \phi$ if and only if $M \models_s \phi$ for all $s \in X$. 
From Assignments to Teams

Assignments as States of Things

In Tarski’s semantics, an assignment $s$ is a *possible state of things*. $M \models_s \phi$ if and only if $\phi$ is true of the state $s$.

Teams (= Belief Sets)

A team $X$ is a set of assignments (possible states of things).

Knowledge is Satisfaction

$M \models_X \phi$ if and only if $M \models_s \phi$ for all $s \in X$. 
From Assignments to Teams

Assignments as States of Things

In Tarski’s semantics, an assignment $s$ is a possible state of things. $M \models_s \phi$ if and only if $\phi$ is true of the state $s$.

Teams (= Belief Sets)

A team $X$ is a set of assignments (possible states of things).

Knowledge is Satisfaction

$M \models_X \phi$ if and only if $M \models_s \phi$ for all $s \in X$. 
Hodges’ Semantics for First Order Logic

Knowledge is Satisfaction

\[ M \models_X \phi \text{ if and only if } M \models_s \phi \text{ for all } s \in X. \]

- If \( \alpha \) literal, \( M \models_X \alpha \) iff for all \( s \in X, M \models_s \alpha \);
- \( M \models_X \phi \land \psi \) iff for all \( s \in X, M \models_s \phi \) and \( M \models_s \psi \);
- \( M \models_X \phi \lor \psi \) iff for all \( s \in X, M \models_s \phi \) or \( M \models_s \psi \);
- \( M \models_X \exists x \phi \) iff for all \( s \in X, \exists m \in \text{Dom}(M) \) s.t. \( M \models_s[m/x] \phi \);
- \( M \models_X \forall x \phi \) iff for all \( s \in X, \forall m \in \text{Dom}(M), M \models_s[m/x] \phi \).

Aside: Hodges Semantics and Game Theoretic Semantics

Teams correspond to sets of possible states in the subgames of the semantic game.
Hodges’ Semantics for First Order Logic

Knowledge is Satisfaction

\[ M \models_X \phi \text{ if and only if } M \models_s \phi \text{ for all } s \in X. \]

- If \( \alpha \) literal, \( M \models_X \alpha \) iff for all \( s \in X \), \( M \models_s \alpha \);
- \( M \models_X \phi \land \psi \) iff for all \( s \in X \), \( M \models_s \phi \) and \( M \models_s \psi \);
- \( M \models_X \phi \lor \psi \) iff for all \( s \in X \), \( M \models_s \phi \) or \( M \models_s \psi \);
- \( M \models_X \exists x \phi \) iff for all \( s \in X \), \( \exists m \in \text{Dom}(M) \) s.t. \( M \models_s[m/x] \phi \);
- \( M \models_X \forall x \phi \) iff for all \( s \in X \), \( \forall m \in \text{Dom}(M), M \models_s[m/x] \phi \).

Aside: Hodges Semantics and Game Theoretic Semantics

Teams correspond to sets of possible states in the subgames of the semantic game.
Hodges’ Semantics for First Order Logic

Knowledge is Satisfaction

\[ M \models_X \phi \text{ if and only if } M \models_s \phi \text{ for all } s \in X. \]

- If \( \alpha \) literal, \( M \models_X \alpha \) iff for all \( s \in X \), \( M \models_s \alpha \);
- \( M \models_X \phi \land \psi \) iff \( M \models_X \phi \) and \( M \models_X \psi \);
- \( M \models_X \phi \lor \psi \) iff for all \( s \in X \), \( M \models_s \phi \) or \( M \models_s \psi \);
- \( M \models_X \exists x \phi \) iff for all \( s \in X \), \( \exists m \in \text{Dom}(M) \) s.t. \( M \models_{s[m/x]} \phi \);
- \( M \models_X \forall x \phi \) iff for all \( s \in X \), \( \forall m \in \text{Dom}(M) \), \( M \models_{s[m/x]} \phi \).

Aside: Hodges Semantics and Game Theoretic Semantics

Teams correspond to sets of possible states in the subgames of the semantic game.
Hodges’ Semantics for First Order Logic

Knowledge is Satisfaction

\[ M \models_X \phi \text{ if and only if } M \models_s \phi \text{ for all } s \in X. \]

- If \( \alpha \) literal, \( M \models_X \alpha \text{ iff for all } s \in X, M \models_s \alpha; \)
- \( M \models_X \phi \land \psi \text{ iff } M \models_X \phi \text{ and } M \models_X \psi; \)
- \( M \models_X \phi \lor \psi \text{ iff } X = Y \cup Z, M \models_Y \phi \text{ and } M \models_Z \psi; \)
- \( M \models_X \exists x \phi \text{ iff for all } s \in X, \exists m \in \text{Dom}(M) \text{ s.t. } M \models_s[m/x] \phi; \)
- \( M \models_X \forall x \phi \text{ iff for all } s \in X, \forall m \in \text{Dom}(M), M \models_s[m/x] \phi. \)

Aside: Hodges Semantics and Game Theoretic Semantics

Teams correspond to sets of possible states in the subgames of the semantic game.
Hodges’ Semantics for First Order Logic

Knowledge is Satisfaction

\[ M \models_X \phi \text{ if and only if } M \models_s \phi \text{ for all } s \in X. \]

- If \( \alpha \) literal, \( M \models_X \alpha \) if and only if \( M \models_s \alpha \) for all \( s \in X \);
- \( M \models_X \phi \land \psi \) if and only if \( M \models_X \phi \) and \( M \models_X \psi \);
- \( M \models_X \phi \lor \psi \) if and only if \( X = Y \cup Z \), \( M \models_Y \phi \) and \( M \models_Z \psi \);
- \( M \models_X \exists x \phi \) if and only if there exists \( F : X \rightarrow \text{Dom}(M) \) such that \( M \models_X \phi \big|_{F/x} \);
- \( M \models_X \forall x \phi \) if and only if for all \( s \in X \), \( \forall m \in \text{Dom}(M) \), \( M \models_s \phi \big|_{m/x} \).

Aside: Hodges Semantics and Game Theoretic Semantics

Teams correspond to sets of possible states in the subgames of the semantic game.
Hodges’ Semantics for First Order Logic

Knowledge is Satisfaction

\[ M \models_X \phi \text{ if and only if } M \models_s \phi \text{ for all } s \in X. \]

- If \( \alpha \) literal, \( M \models_X \alpha \) iff for all \( s \in X, M \models_s \alpha \);
- \( M \models_X \phi \land \psi \) iff \( M \models_X \phi \) and \( M \models_X \psi \);
- \( M \models_X \phi \lor \psi \) iff \( X = Y \cup Z, M \models_Y \phi \) and \( M \models_Z \psi \);
- \( M \models_X \exists x \phi \) iff \( \exists F : X \to \text{Dom}(M) \) s.t. \( M \models_X[F/x] \phi \);
- \( M \models_X \forall x \phi \) iff \( M \models_X[M/x] \phi \).

Aside: Hodges Semantics and Game Theoretic Semantics

Teams correspond to sets of possible states in the subgames of the semantic game.
Hodges’ Semantics for First Order Logic

Knowledge is Satisfaction

\[ M \models_X \phi \text{ if and only if } M \models_s \phi \text{ for all } s \in X. \]

- If \( \alpha \) literal, \( M \models_X \alpha \) iff for all \( s \in X, M \models_s \alpha; \)
- \( M \models_X \phi \land \psi \) iff \( M \models_X \phi \) and \( M \models_X \psi; \)
- \( M \models_X \phi \lor \psi \) iff \( X = Y \cup Z, M \models_Y \phi \) and \( M \models_Z \psi; \)
- \( M \models_X \exists x \phi \) iff \( \exists F: X \rightarrow \text{Dom}(M) \text{ s.t. } M \models_X[F/x] \phi; \)
- \( M \models_X \forall x \phi \) iff \( M \models_X[M/x] \phi. \)

Aside: Hodges Semantics and Game Theoretic Semantics

Teams correspond to sets of possible states in the subgames of the semantic game.
Outline

1. Tarskian Interpreted Languages and Locality
   - Tarskian Interpreted Languages
   - Locality

2. Logics of Imperfect Information
   - Team Semantics
   - Dependence Logic and IF Logic

3. Independence Logic
   - Independence Logic
   - Nonlocality
   - Lax and Strict Semantics
Knowledge is Satisfaction

For \( \phi \) first order, \( M \models_X \phi \) if and only if \( M \models_s \phi \) for all \( s \in X \).

- Can add new atomic formulas:
  \[
  M \models_X \text{EVEN}(t) \iff |\{s\langle t \rangle : s \in X\}| \text{ is even};
  \]

- Can add new operators:
  \[
  M \models_X \sim \phi \iff M \not\models_X \phi.
  \]
Two formalisms for imperfect information:

<table>
<thead>
<tr>
<th>Dependence Logic</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M \models_{X} (t_1 \ldots t_n)$ iff for all $s, s' \in X$,</td>
</tr>
<tr>
<td>$t_1\langle s \rangle = t_1\langle s' \rangle, \ldots, t_{n-1}\langle s \rangle = t_{n-1}\langle s' \rangle \Rightarrow t_n\langle s \rangle = t_n\langle s' \rangle.$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Independence-Friendly Logic</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M \models_{X} (\exists y/V) \phi$ iff $\exists F : X \rightarrow \text{Dom}(M)$ s.t.</td>
</tr>
<tr>
<td>If $s$ and $s'$ differ only over $V$ then $F(s) = F(s')$;</td>
</tr>
<tr>
<td>$M \models_{X}[F/x] \phi.$</td>
</tr>
</tbody>
</table>
Dependence logic and IF logic are expressively equivalent (wrt teams with fixed domains);

They are not quite different enough to be entirely distinct approaches, but not close enough to be just notational variants;

Which one to use?

A (not particularly new) answer
If we want locality, we should favor Dependence Logic.
Hodgesian interpreted language

\[ L = \langle \text{Symb}_L, \text{Expr}_L, I_L \rangle, \text{ where} \]

1. \text{Symb}_L \text{ is a set of } \textit{symbols};
2. \text{Expr}_L \text{ is a set of } \textit{expressions};
3. \( I_L \subseteq \mathcal{P}(I_L) \) \text{ is a set of } L\text{-interpretations.}

Truth Interpretation

\[ \models_L \subseteq \mathcal{P}(I_L) \times \text{Expr}_L. \text{ We write } I_L \models \phi \text{ instead of } (I, \phi) \in \models_L. \]
Locality in Hodgesian Languages

**Locality**

If $V_\phi = \{ u \in \text{Symb}_L : u \text{ occurs in } \phi \}$ and $\rho$ is a replacement such that $I(V_\phi) = J(\rho(V_\phi))$ then

$$I \models \phi \text{ if and only if } J \models \phi[\rho].$$

**Team-locality**

A formula $\phi$ is team-local iff, for all $M$ and $X$,

$$M \models_X \phi \iff M \models_{X \uparrow \text{Var}(\phi)} \phi$$
Theorem

IF Logic is not team-local, but Dependence Logic is so.

Proof (1).

The proof that Dependence Logic is team-local is a straightforward induction over the structure of the formula. On the other hand, consider $\phi = (\exists z/y)z = y$ and the team

<table>
<thead>
<tr>
<th>$X$</th>
<th>$x$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_0$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$s_1$</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Now, $M \models_X \phi$: just take $F(s_i) = s_i(x)$. 
Locality in Hodgesian Languages

Theorem

IF Logic is not team-local, but Dependence Logic is so.

Proof (1).

The proof that Dependence Logic is team-local is a straightforward induction over the structure of the formula. On the other hand, consider $\phi = (\exists z/y)z = y$ and the team

$$X[F/z] = \begin{array}{c|ccc}
\ x & y & z \\
\hline
s_0 & 0 & 0 & 0 \\
s_1 & 1 & 1 & 1 \\
\end{array}$$

Now, $M \models x \phi$: just take $F(s_i) = s_i(x)$. 

Pietro Galliani
Locality in Hodgesian Languages

**Theorem**

IF Logic is not team-local, but Dependence Logic is so.

**Proof (2).**

Consider $\phi = (\exists z/y) z = y$ and the team

$$X_{\var{\phi}} = \begin{array}{c|c}
\var{\phi} & y \\
\hline
s_0 & 0 \\
\hline
s_1 & 1 \\
\end{array}$$

If $F$ is independent on $y$, then $F(s_0) = F(s_1) = m$; and since $m \neq 0$ or $m \neq 1$, it is not true that $F(s_i) = s_i(y)$ for all $i$. 

Pietro Galliani

Some denotationally indistinguishable operators
Theorem

IF Logic is not team-local, but Dependence Logic is so.

Proof (2).

Consider $\phi = (\exists z/y)z = y$ and the team

$$X_{\text{Var}(\phi)}[F/z] = \begin{array}{c|cc}
    & y & z \\
  s_0 & 0 & m \\
  s_1 & 1 & m \\
\end{array}$$

If $F$ is independent on $y$, then $F(s_0) = F(s_1) = m$; and since $m \neq 0$ or $m \neq 1$, it is not true that $F(s_i) = s_i(y)$ for all $i$.  

Pietro Galliani

Some denotationally indistinguishable operators
Outline

1. Tarskian Interpreted Languages and Locality
   - Tarskian Interpreted Languages
   - Locality

2. Logics of Imperfect Information
   - Team Semantics
   - Dependence Logic and IF Logic

3. Independence Logic
   - Independence Logic
   - Nonlocality
   - Lax and Strict Semantics
Independence Logic (Grädel and Väänänen 2010)

Independence Atoms (Grädel, Väänänen)

\[ M \models_X \vec{t}_2 \perp_{\vec{t}_1} \vec{t}_3 \text{ if and only if, for all } s, s' \in X \text{ such that } \vec{t}_1 \langle s \rangle = \vec{t}_1 \langle s' \rangle \text{ there exists a } s'' \in X \text{ such that} \]
\[ \vec{t}_1 \langle s'' \rangle \vec{t}_2 \langle s'' \rangle = \vec{t}_1 \langle s \rangle \vec{t}_2 \langle s \rangle, \quad \vec{t}_1 \langle s'' \rangle \vec{t}_3 \langle s'' \rangle = \vec{t}_1 \langle s' \rangle \vec{t}_3 \langle s' \rangle. \]

Independence Logic (Grädel, Väänänen)

Independence Logic = First Order Logic + Independence Atoms.
Independence Logic (Grädel and Väänänen 2010)

Independence Atoms = Informational Independence

\[ M \models x \; \vec{t}_2 \perp_{\vec{t}_1} \vec{t}_3 \] if and only if, once the value of \( \vec{t}_1 \) is fixed, learning new information about the value of \( \vec{t}_2 \) tells us nothing new about the value of \( \vec{t}_3 \).

Constant Independence Atoms: \( \vec{t}_2 \perp \vec{t}_3 \equiv \vec{t}_2 \perp_{\emptyset} \vec{t}_3 \)

\[ M \models x \; \vec{t}_2 \perp \vec{t}_3 \] if and only if learning new information about the value of \( \vec{t}_1 \) tells us nothing about the value of \( \vec{t}_3 \).
Independence Logic (Grädel and Väänänen 2010)

Constant Independence Atoms: \( \vec{t}_2 \perp \vec{t}_3 \equiv \vec{t}_2 \perp \emptyset \vec{t}_3 \)

\( M \models_{X} \vec{t}_2 \perp \vec{t}_3 \) if and only if learning new information about the value of \( \vec{t}_1 \) tells us nothing about the value of \( \vec{t}_3 \).

Example (1)

Consider the team

\[
X = \begin{array}{c|cc}
    & x & y \\
\hline
s_0 & 0 & 0 \\
s_1 & 0 & 1 \\
s_2 & 1 & 1 \\
\end{array}
\]

Then \( M \not\models_{X} x \perp y \): learning that \( x = 1 \) would allow us to infer that \( y = 1 \) too.
Constant Independence Atoms: $\vec{t}_2 \perp \vec{t}_3 \equiv \vec{t}_2 \perp \emptyset \vec{t}_3$

$M \models_X \vec{t}_2 \perp \vec{t}_3$ if and only if learning new information about the value of $\vec{t}_1$ tells us nothing about the value of $\vec{t}_3$.

Example (1)
Consider the team

<table>
<thead>
<tr>
<th></th>
<th>$x$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_0$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$s_1$</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$s_2$</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Then $M \not\models_X x \perp y$: learning that $x = 1$ would allow us to infer that $y = 1$ too.
Constant Independence Atoms: \( \vec{t}_2 \perp \vec{t}_3 \equiv \vec{t}_2 \perp_{\emptyset} \vec{t}_3 \)

\( M \models_{\chi} \vec{t}_2 \perp \vec{t}_3 \) if and only if learning new information about the value of \( \vec{t}_1 \) tells us nothing about the value of \( \vec{t}_3 \).

Example (2)

Consider instead

<table>
<thead>
<tr>
<th></th>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>( s_0 )</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( s_1 )</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>( s_2 )</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>( s_3 )</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Then \( M \models_{\chi} x \perp y \): learning that \( x = m \in \{0, 1\} \) tells us nothing about \( y \).
Properties of Independence Logic (Grädel, Väänänen)

Dependence Logic is contained in Independence Logic

$$M \models_{X} = (\vec{t}, t_n) \text{ if and only if } M \models_{X} t_n \perp_{\vec{t}} t_n.$$  

Expressive Power

Equivalent to Dependence Logic (and $\Sigma^1_1$) over sentences.

Team Definability (Galliani 2011)

Its open formulas characterize all $\Sigma^1_1$ classes of teams (stronger than Dependence Logic).
Outline

1. Tarskian Interpreted Languages and Locality
   - Tarskian Interpreted Languages
   - Locality

2. Logics of Imperfect Information
   - Team Semantics
   - Dependence Logic and IF Logic

3. Independence Logic
   - Independence Logic
   - Nonlocality
   - Lax and Strict Semantics
Nonlocality

A problem

With this semantics, Independence Logic is not team-local!

Proof (1).

Consider the team

\[ X = \begin{array}{c|ccc}
  & x & y & z \\
  s_0 & 0 & 0 & 0 \\
  s_1 & 0 & 0 & 1 \\
  s_2 & 0 & 1 & 0 \\
  s_3 & 1 & 1 & 0 \\
\end{array} \]

and the formula \( \phi = \exists w((y = 0 \rightarrow w = z) \land w \perp y) \).
Proof (2).

Consider the team

<table>
<thead>
<tr>
<th></th>
<th>x</th>
<th>y</th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_0$</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$s_1$</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$s_2$</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$s_3$</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

and the formula $\phi = \exists w ((y = 0 \rightarrow w = z) \land w \perp y)$. For $F(s_0) = F(s_2) = 0$, $F(s_1) = F(s_3) = 1$, the formula is verified.
Proof (2).

Consider the team

<table>
<thead>
<tr>
<th></th>
<th>x</th>
<th>y</th>
<th>z</th>
<th>w</th>
</tr>
</thead>
<tbody>
<tr>
<td>(s_0)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>(s_1)</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>(s_2)</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>(s_3)</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

and the formula \(\phi = \exists w((y = 0 \rightarrow w = z) \land w \perp y)\).

For \(F(s_0) = F(s_2) = 0, F(s_1) = F(s_3) = 1\), the formula is verified.
Proof (3).

However,

\[ X_{\text{Var}(\phi)} = \begin{array}{c|cc}
  & y & z \\
  s_0 & 0 & 0 \\
  s_1 & 0 & 1 \\
  s_2 & 1 & 0 \\
\end{array} \]

Then for any \( F \), if \( F(s_0) \neq 0 \) or \( F(s_1) \neq 1 \) it is not true that \( y = 0 \rightarrow w = z \); and otherwise, no matter what \( F(s_2) \) is we have that \( w \perp y \) does not hold.
Proof (3).

However,

\[ X_{\uparrow \text{Var}(\phi)}[F/w] = \begin{array}{c|ccc}
    & y & z & w \\
\hline
s_0 & 0 & 0 & 0 \\
s_1 & 0 & 1 & 1 \\
s_2 & 1 & 0 & \text{F}(s_2) \\
\end{array} \]

Then for any \( F \), if \( F(s_0) \neq 0 \) or \( F(s_1) \neq 1 \) it is not true that \( y = 0 \rightarrow w = z \); and otherwise, no matter what \( F(s_2) \) is we have that \( w \perp y \) does not hold.
Proof (3).

However,

\[
X_{\uparrow \text{Var}(\phi)}[F/w] =
\begin{array}{c|ccc}
 & y & z & w \\
 s_0 & 0 & 0 & 0 \\
 s_1 & 0 & 1 & 1 \\
 s_2 & 1 & 0 & 0?
\end{array}
\]

Then for any \( F \), if \( F(s_0) \neq 0 \) or \( F(s_1) \neq 1 \) it is not true that \( y = 0 \rightarrow w = z \); and otherwise, no matter what \( F(s_2) \) is we have that \( w \perp y \) does not hold.
Proof (3).

However, 

\[ X_{\uparrow \text{Var}(\phi)}[F/w] = \begin{array}{c|ccc}
  & y & z & w \\
  s_0 & 0 & 0 & 0 \\
  s_1 & 0 & 1 & 1 \\
  s_2 & 1 & 0 & 0 \end{array} \]

Then for any \( F \), if \( F(s_0) \neq 0 \) or \( F(s_1) \neq 1 \) it is not true that \( y = 0 \rightarrow w = z \); and otherwise, no matter what \( F(s_2) \) is we have that \( w \perp y \) does not hold.
Proof (3).

However,

\[ X_{\varphi} \upharpoonright \text{Var}(\phi) [F/w] = \begin{array}{c|ccc}
  & y & z & w \\
\hline
s_0 & 0 & 0 & 0 \\
s_1 & 0 & 1 & 1 \\
s_2 & 1 & 0 & 1 \\
\end{array} \]

Then for any \( F \), if \( F(s_0) \neq 0 \) or \( F(s_1) \neq 1 \) it is not true that \( y = 0 \rightarrow w = z \); and otherwise, no matter what \( F(s_2) \) is we have that \( w \perp y \) does not hold.
Proof (3).

However,

\[ X^{\uparrow \text{Var}(\phi)}[F/w] = \]

<table>
<thead>
<tr>
<th></th>
<th>y</th>
<th>z</th>
<th>w</th>
</tr>
</thead>
<tbody>
<tr>
<td>s₀</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>s₁</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>s₂</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Then for any \( F \), if \( F(s₀) \neq 0 \) or \( F(s₁) \neq 1 \) it is not true that \( y = 0 \rightarrow w = z \); and otherwise, no matter what \( F(s₂) \) is we have that \( w \perp y \) does not hold.
Proof (3).

However,

\[ X_{\varphi^{\text{var}}} [F / w] = \]

\[
\begin{array}{c|ccc}
  & y & z & w \\
\hline
  s_0 & 0 & 0 & 0 \\
  s_1 & 0 & 1 & 1 \\
  s_2 & 1 & 0 & \times
\end{array}
\]

Then for any \( F \), if \( F(s_0) \neq 0 \) or \( F(s_1) \neq 1 \) it is not true that \( y = 0 \rightarrow w = z \); and otherwise, no matter what \( F(s_2) \) is we have that \( w \perp y \) does not hold.
Outline

1. Tarskian Interpreted Languages and Locality
   - Tarskian Interpreted Languages
   - Locality

2. Logics of Imperfect Information
   - Team Semantics
   - Dependence Logic and IF Logic

3. Independence Logic
   - Independence Logic
   - Nonlocality
   - Lax and Strict Semantics

Some denotationally indistinguishable operators
An alternative semantics for existential quantifiers

Lax Existential Quantification

\[ M \models^L \exists x \phi \text{ iff } \exists H : X \rightarrow \mathcal{P}(\text{Dom}(M)) \setminus \{\emptyset\} \text{ s.t. } M \models^X [H/x] \phi, \]

where

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

Example

Let \( Y \) be

<table>
<thead>
<tr>
<th></th>
<th>y</th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>s_0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>s_1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>s_2</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

and let \( H(s_0) = \{0\}, H(s_1) = \{1\}, H(s_2) = \{0, 1\} \).
An alternative semantics for existential quantifiers

Lax Existential Quantification

\[ M \models_X \exists^L x \phi \iff \exists H : X \rightarrow \mathcal{P}(\text{Dom}(M)) \setminus \{\emptyset\} \text{ s.t. } M \models_X[H/x] \phi, \]

where

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

Example

Let \( Y \) be

<table>
<thead>
<tr>
<th></th>
<th>y</th>
<th>z</th>
<th>w</th>
</tr>
</thead>
<tbody>
<tr>
<td>( s_0 )</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( s_1 )</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>( s_2 )</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

and let \( H(s_0) = \{0\}, H(s_1) = \{1\}, H(s_2) = \{0, 1\} \).
An alternative semantics for existential quantifiers

### Lax Existential Quantification

- \( M \models_X \exists^L x \phi \) iff \( \exists H : X \rightarrow \mathcal{P}(\text{Dom}(M)) \setminus \{\emptyset\} \) s.t. \( M \models_X [H/x] \phi \),

where

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

### Example

Let \( Y \) be

<table>
<thead>
<tr>
<th></th>
<th>y</th>
<th>z</th>
<th>w</th>
</tr>
</thead>
<tbody>
<tr>
<td>( s_0 )</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( s_1 )</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>( s_2 )</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

and let \( H(s_0) = \{0\} \), \( H(s_1) = \{1\} \), \( H(s_2) = \{0, 1\} \).
An alternative semantics for existential quantifiers

**Lax Existential Quantification**

\[ M \models^L_X \exists x \phi \text{ iff } \exists H : X \rightarrow \mathcal{P}(\text{Dom}(M)) \setminus \{\emptyset\} \text{ s.t. } M \models^L_X H / x \phi, \]

where

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

**Example**

Let \( Y \) be

<table>
<thead>
<tr>
<th></th>
<th>y</th>
<th>z</th>
<th>w</th>
</tr>
</thead>
<tbody>
<tr>
<td>( s_0 )</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( s_1 )</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>( s_2 )</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( s'_2 )</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

and let \( H(s_0) = \{0\}, H(s_1) = \{1\}, H(s_2) = \{0, 1\} \).
An alternative semantics for existential quantifiers

Lax Existential Quantification

\[ M \models^L X \exists x \phi \text{ iff } \exists H : X \rightarrow \mathcal{P}(\text{Dom}(M)) \setminus \{\emptyset\} \text{ s.t. } M \models X[H/x] \phi, \]

where

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

Generalized Quantifiers

The above rule arises naturally from Engström’s treatment of generalized quantifiers in Dependence Logic (Engström 2011).

In Dependence Logic, nothing changes

For Dependence Logic, it does not make any difference whether one uses this rule or the original one.
An alternative semantics for existential quantifiers

**Lax Existential Quantification**

\[ M \models^L x \phi \text{ iff } \exists H : X \rightarrow \mathcal{P}(\text{Dom}(M)) \setminus \{\emptyset\} \text{ s.t. } M \models X[H/x] \phi, \]

where

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

**Theorem**

Independence Logic with lax quantification is team-local.

**Proof.**

By induction.
From strict to lax semantics

If $z$ not in $\psi$, $z \neq x$, and $\psi$ is team-local then

$$M \models x \exists^L x \psi \iff M \models x \forall z \exists x \psi.$$ 

Proof.

Suppose that for $H : X \to \mathcal{P}(\text{Dom}(M)) \setminus \{\emptyset\}$, $M \models^x[H/x] \psi$. For every $s \in X$, let $m_s \in H(s)$; then define $F : X[M/z] \to M$ as

$$F(s[m/z]) = \begin{cases} m & \text{if } m \in H(s); \\ m_s & \text{otherwise}. \end{cases}$$

Forgetting the variable $z$, $X[M/z][F/x]$ is precisely $X[H/z]$; hence, $M \models^x[M/z][F/x] \psi$, as required.
Recovering strict existentials

Let $\vec{x}$ be a fixed tuple of variables. Then, for all teams $X$ with $\text{Dom}(X) = \vec{x}$, for all $z$ and all $\psi \in \mathcal{T}$,

$$M \models_X \exists z \psi \iff M \models_X \exists^L z((\vec{x}, z) \wedge \psi).$$

Corollary

Independence Logic with lax semantics has the same expressive power of Independence Logic with strict semantics (as long as the domain of the team is fixed).
The “lax semantics” for existential quantification is equivalent to the standard one over Dependence Logic formulas, and preserves team-locality in the case of Independence Logic.

Furthermore, the choice does not affect the expressive power of the logic.

Hence, I suggest that this rule is the “better” one for existential quantifiers in Dependence Logic, Independence Logic and in similar formalisms, such as for example Inclusion Logic (Galliani 2011).
Further Observations

4. When examining a new variant of Dependence Logic or of another formalism, one of the first questions that we should ask is “does this logic satisfy (team) locality?”

5. If it does not, the logic might still be interesting, of course. But then we should ask ourselves “Are we gaining anything useful by losing locality here? Or is there an equivalent (for our purposes) formalism which has locality?”

6. Adding new connectives to a logic of imperfect information may affect the properties of the resulting formalism in unexpected ways. We might want to develop a battery of “standard questions” to ask about such variants in order to better understand their properties.
The End (for now...) 

(Very Basic) Bibliography

- Jouko Väänänen, *Dependence Logic*.
- Denis Bonnay, Dag Westerståhl, *Consequence Mining: A New Approach to Logical Constants*.
- Erich Grädel, Jouko Väänänen, *Dependence and Independence*.
- Fredrik Engström, *Generalized Quantifiers in Dependence Logic*.