Logics of independence and dependence

Jonni Virtema

University of Helsinki, Finland

Phileth seminar @ Hokkaido University, 11th of May 2017

Outline

Introduction

What is logic ?

What is logic ?

- Mathematics, philosophy, linguistics, and theoretical computer science.

What is logic ?

- Mathematics, philosophy, linguistics, and theoretical computer science.
- Logic is an abstract machinery that can be used in describing the state and behaviour of systems, and deduction related to these systems.

What is logic (model theoretic view)?

- Mathematics, philosophy, linguistics, and theoretical computer science.
- Logic is an abstract machinery that can be used in describing the state and behaviour of systems, and deduction related to these systems.
- The core notions are **models** and **formulae**.

What are models and formulae?

- A model is an abstraction of some state of affairs.
- A collection of points together with some arrows from points to points is a model (e.g., modelling a rail network).

What are models and formulae?

- A model is an abstraction of some state of affairs.
- A collection of points together with some arrows from points to points is a model (e.g., modelling a rail network).
- A formula is an abstraction of a claim related to some state of affairs.
- A sentence that describes a possible property of a model is a formula (e.g., a statement that there is a rail connection from Tokyo to Sapporo).

First-order models

- An n-ary relation R over a set A is a subset of A^n.
- For simplicity we now consider only 1-ary and 2-ary relations, i.e., subsets of A and $\{(a, b) \mid a, b \in A\}$, respectively.

First-order models

- An n-ary relation R over a set A is a subset of A^n.
- For simplicity we now consider only 1-ary and 2-ary relations, i.e., subsets of A and $\{(a, b) \mid a, b \in A\}$, respectively.
- (We omit function symbols and constant symbols)

First-order models

- An n-ary relation R over a set A is a subset of A^n.
- For simplicity we now consider only 1-ary and 2-ary relations, i.e., subsets of A and $\{(a, b) \mid a, b \in A\}$, respectively.
- (We omit function symbols and constant symbols)

Definition

Let $\tau = \{P, R\}$ where P is a 1-ary relation symbol and R a 2-ary relation symbol. A *τ-model* $\mathfrak{A}$ is a tuple (A, P, R), where

- A is a nonempty set called *the domain* of $\mathfrak{A}$,
- $R \subseteq A \times A$ is a binary relation over A, and
- $P \subseteq A$ is a unary relation over A.

First-order models

Example

- Let A be the set of all cities in Japan.
- Let $R = \{(Sapporo, Hakodate), (Hakodate, Hirosaki), (Hirosaki, Sapporo)\}$.
- Let $P = \{Hakodate, Hirosaki\}$.

Now (A, R, P) is a first-order model that describes my travel during the Golden Week.

First-order language

Definition

The formulae for first-order logic $\mathcal{FO}$ over vocabulary $\{R, P\}$ is generated by the following grammar:

$$\varphi ::= P(x) \mid R(x,y) \mid x = y \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \exists x\varphi \mid \forall x\varphi,$$

where x and y are variable symbols.

First-order language

Definition

The formulae for first-order logic $\mathcal{FO}$ over vocabulary $\{R, P\}$ is generated by the following grammar:

$$\varphi ::= P(x) \mid R(x,y) \mid x = y \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \exists x\varphi \mid \forall x\varphi,$$

where x and y are variable symbols.

Example

Using the example model of the last slide, the sentence:
I visited two Japanese cities during the Golden Week can be written in first-oder logic as follows:

$$\exists x \exists y(\neg x = y \wedge P(x) \wedge P(y))$$

The connection between models and formulae

A model describes some system and a formula describes some possible property of that system.

The connection between models and formulae

A model describes some system and a formula describes some possible property of that system.

The question:
Does the property described hold in the system or not?

The connection between models and formulae

A model describes some system and a formula describes some possible property of that system.

The question:
Does the property described hold in the system or not?
Is the formula true in the model?

The connection between models and formulae

A model describes some system and a formula describes some possible property of that system.

The question:
Does the property described hold in the system or not?
Is the formula true in the model?

Formally:
Does $\mathfrak{A}, s \models \varphi$ hold?

Tarski semantics of first-order logic (Tarski 1930s)

An assignment $s : \mathrm{Var} \to A$ is a function that gives a value for each variable symbol in Var.

Definition

Let $\mathfrak{A}$ be a $\{R, P\}$ model and s an assignment. The satisfaction relation $\mathfrak{A}, s \models \varphi$ for $\mathcal{FO}$ is defined as follows.

$$
\begin{aligned}
\mathfrak{A}, s \models x = y &\quad\Leftrightarrow\quad s(x) = s(y).\\
\mathfrak{A}, s \models P(x) &\quad\Leftrightarrow\quad s(x) \in P.\\
\mathfrak{A}, s \models R(x, y) &\quad\Leftrightarrow\quad (s(x), s(y)) \in R.\\
\mathfrak{A}, s \models \neg\varphi &\quad\Leftrightarrow\quad \mathfrak{A}, s \not\models \varphi.\\
\mathfrak{A}, s \models (\varphi \wedge \psi) &\quad\Leftrightarrow\quad \mathfrak{A}, s \models \varphi \text{ and } \mathfrak{A}, s \models \psi.\\
\mathfrak{A}, s \models (\varphi \vee \psi) &\quad\Leftrightarrow\quad \mathfrak{A}, s \models \varphi \text{ or } \mathfrak{A}, s \models \psi.\\
\mathfrak{A}, s \models \exists x\varphi &\quad\Leftrightarrow\quad \mathfrak{A}, s(x \mapsto a) \models \varphi \text{ for some } a \in A.\\
\mathfrak{A}, s \models \forall x\varphi &\quad\Leftrightarrow\quad \mathfrak{A}, s(x \mapsto a) \models \varphi \text{ for every } a \in A.
\end{aligned}
$$

Toy example

We can now use Tarski semantics to check that, indeed, I visited two Japanense cities during Golden Week, since

$$\mathfrak{A}, s \models \exists x \exists y (\neg x = y \wedge P(x) \wedge P(y)),$$

where $\mathfrak{A} = (A, R, P)$ is the first-order model defined such that

- A is the set of all cities in Japan.
- $R = \{(Sapporo, Hakodate), (Hakodate, Hirosaki), (Hirosaki, Sapporo)\}$.
- $P = \{Hakodate, Hirosaki\}$.

PART 1

GTS semantics for FO

Game theoretic semantics of $\mathcal{FO}$ (Hintikka 1968)

- An alternative way to give meaning for $\mathcal{FO}$-formulae.
- The game $G(\varphi, (\mathfrak{A}, s))$ is a two-player game.
- Players: $(\forall)$ Abelard and $(\exists)$ Eloise.

Game theoretic semantics of $\mathcal{FO}$ (Hintikka 1968)

- An alternative way to give meaning for $\mathcal{FO}$-formulae.
- The game $G(\varphi, (\mathfrak{A}, s))$ is a two-player game.
- Players: $(\forall)$ Abelard and $(\exists)$ Eloise.
- Abelard wants to establish that φ is not true in $\mathfrak{A}, s$, while Eloise wishes to show that φ is true in $\mathfrak{A}, s$.
- Abelard controls $\forall$ and $\wedge$, while Eloise controls $\exists$ and $\vee$.

Game theoretic semantics of $\mathcal{FO}$ (Hintikka 1968)

- An alternative way to give meaning for $\mathcal{FO}$-formulae.
- The game $G(\varphi, (\mathfrak{A}, s))$ is a two-player game.
- Players: $(\forall)$ Abelard and $(\exists)$ Eloise.
- Abelard wants to establish that φ is not true in $\mathfrak{A}, s$, while Eloise wishes to show that φ is true in $\mathfrak{A}, s$.
- Abelard controls $\forall$ and $\wedge$, while Eloise controls $\exists$ and $\vee$.
- (Negation $\neg$ switches the roles of Abelard and Eloise. However in order to simplify things we assume that negations may occur only in front of atomic formulae.)

Rules of $G(\varphi, (\mathfrak{A}, s))$

- A *play* of the game starts from the position $(\varphi, (\mathfrak{A}, s))$.

Rules of $G(\varphi, (\mathfrak{A}, s))$

- A *play* of the game starts from the position $(\varphi, (\mathfrak{A}, s))$.
- In position $(\psi_1 \wedge \psi_2, (\mathfrak{A}, s))$, Abelard chooses $i \in \{1, 2\}$ and the game continues from position $(\psi_i, (\mathfrak{A}, s))$.

Rules of $G(\varphi, (\mathfrak{A}, s))$

- A *play* of the game starts from the position $(\varphi, (\mathfrak{A}, s))$.
- In position $(\psi_1 \wedge \psi_2, (\mathfrak{A}, s))$, Abelard chooses $i \in \{1, 2\}$ and the game continues from position $(\psi_i, (\mathfrak{A}, s))$.
- In position $(\psi_1 \vee \psi_2, (\mathfrak{A}, s))$, Eloise chooses $i \in \{1, 2\}$ and the game continues from position $(\psi_i, (\mathfrak{A}, s))$.

Rules of $G(\varphi, (\mathfrak{A}, s))$

- A *play* of the game starts from the position $(\varphi, (\mathfrak{A}, s))$.
- In position $(\psi_1 \wedge \psi_2, (\mathfrak{A}, s))$, Abelard chooses $i \in \{1, 2\}$ and the game continues from position $(\psi_i, (\mathfrak{A}, s))$.
- In position $(\psi_1 \vee \psi_2, (\mathfrak{A}, s))$, Eloise chooses $i \in \{1, 2\}$ and the game continues from position $(\psi_i, (\mathfrak{A}, s))$.
- In position $(\forall x \psi, (\mathfrak{A}, s))$, Abelard chooses $a \in A$ and the game continues from position $(\psi, (\mathfrak{A}, s(x \mapsto a)))$.

Rules of $G(\varphi, (\mathfrak{A}, s))$

- A *play* of the game starts from the position $(\varphi, (\mathfrak{A}, s))$.
- In position $(\psi_1 \wedge \psi_2, (\mathfrak{A}, s))$, Abelard chooses $i \in \{1, 2\}$ and the game continues from position $(\psi_i, (\mathfrak{A}, s))$.
- In position $(\psi_1 \vee \psi_2, (\mathfrak{A}, s))$, Eloise chooses $i \in \{1, 2\}$ and the game continues from position $(\psi_i, (\mathfrak{A}, s))$.
- In position $(\forall x \psi, (\mathfrak{A}, s))$, Abelard chooses $a \in A$ and the game continues from position $(\psi, (\mathfrak{A}, s(x \mapsto a)))$.
- In position $(\exists x \psi, (\mathfrak{A}, s))$, Eloise chooses $a \in A$ and the game continues from position $(\psi, (\mathfrak{A}, s(x \mapsto a)))$.

Rules of $G(\varphi, (\mathfrak{A}, s))$

- A *play* of the game starts from the position $(\varphi, (\mathfrak{A}, s))$.
- In position $(\psi_1 \wedge \psi_2, (\mathfrak{A}, s))$, Abelard chooses $i \in \{1, 2\}$ and the game continues from position $(\psi_i, (\mathfrak{A}, s))$.
- In position $(\psi_1 \vee \psi_2, (\mathfrak{A}, s))$, Eloise chooses $i \in \{1, 2\}$ and the game continues from position $(\psi_i, (\mathfrak{A}, s))$.
- In position $(\forall x \psi, (\mathfrak{A}, s))$, Abelard chooses $a \in A$ and the game continues from position $(\psi, (\mathfrak{A}, s(x \mapsto a)))$.
- In position $(\exists x \psi, (\mathfrak{A}, s))$, Eloise chooses $a \in A$ and the game continues from position $(\psi, (\mathfrak{A}, s(x \mapsto a)))$.
- The game is played until a position $(\psi, (\mathfrak{A}, t))$ is reached, where ψ is a literal.

End Game

- Eloise wins the play of the game if
  - $(x = y, (\mathfrak{A}, t))$ is reached and $t(x) = t(y)$,
  - $(\neg x = y, (\mathfrak{A}, t))$ is reached and $t(x) \neq t(y)$

End Game

- Eloise wins the play of the game if
  - $(x = y, (\mathfrak{A}, t))$ is reached and $t(x) = t(y)$,
  - $(\neg x = y, (\mathfrak{A}, t))$ is reached and $t(x) \neq t(y)$
  - $(P(x), (\mathfrak{A}, t))$ is reached and $t(x) \in P$,
  - $(\neg P(x), (\mathfrak{A}, t))$ is reached and $t(x) \notin P$,
  - $(R(x, y), (\mathfrak{A}, t))$ is reached and $(t(x), t(y)) \in R$,
  - $\neg(R(x, y), (\mathfrak{A}, t))$ is reached and $(t(x), t(y)) \notin R$,
- otherwise Abelard wins the play.

End Game

- Eloise wins the play of the game if
  - $(x = y, (\mathfrak{A}, t))$ is reached and $t(x) = t(y)$,
  - $(\neg x = y, (\mathfrak{A}, t))$ is reached and $t(x) \neq t(y)$
  - $(P(x), (\mathfrak{A}, t))$ is reached and $t(x) \in P$,
  - $(\neg P(x), (\mathfrak{A}, t))$ is reached and $t(x) \notin P$,
  - $(R(x, y), (\mathfrak{A}, t))$ is reached and $(t(x), t(y)) \in R$,
  - $\neg(R(x, y), (\mathfrak{A}, t))$ is reached and $(t(x), t(y)) \notin R$,
- otherwise Abelard wins the play.

To win the Game it is not enough to win a single play. To win the game the player has to be able to win EVERY play.

Strategies

- A strategy for Eloise in the game $G(\varphi, (\mathfrak{A}, s))$ is a function that maps every possible position that she controls and that may occur in some play of the game to a single next position of the play.

Strategies

- A strategy for Eloise in the game $G(\varphi, (\mathfrak{A}, s))$ is a function that maps every possible position that she controls and that may occur in some play of the game to a single next position of the play.
- A strategy for Abelard in the game $G(\varphi, (\mathfrak{A}, s))$ is a function that maps every possible position that he controls and that may occur in some play of the game to a single next position of the play.

Strategies

- A strategy for Eloise in the game $G(\varphi, (\mathfrak{A}, s))$ is a function that maps every possible position that she controls and that may occur in some play of the game to a single next position of the play.
- A strategy for Abelard in the game $G(\varphi, (\mathfrak{A}, s))$ is a function that maps every possible position that he controls and that may occur in some play of the game to a single next position of the play.
- A stategy of a player is a winning strategy for that player if she/he wins every play of the game in which she/he plays according to her/his strategy.

Winner of the game

- Eloise wins the game $G(\varphi, (\mathfrak{A}, s))$ if she has a winning strategy for that game.

Winner of the game

- Eloise wins the game $G(\varphi, (\mathfrak{A}, s))$ if she has a winning strategy for that game.
- A formula φ is true in $\mathfrak{A}, s$ under GTS (denoted by $\mathfrak{A}, s \models_{\mathrm{GTS}} \varphi$) if Eloise wins the game $G(\varphi, (\mathfrak{A}, s))$.

Winner of the game

- Eloise wins the game $G(\varphi, (\mathfrak{A}, s))$ if she has a winning strategy for that game.
- A formula φ is true in $\mathfrak{A}, s$ under GTS (denoted by $\mathfrak{A}, s \models_{\mathrm{GTS}} \varphi$) if Eloise wins the game $G(\varphi, (\mathfrak{A}, s))$.
- $\mathfrak{A}, s \models_{\mathrm{GTS}} \varphi$ holds if and only if $\mathfrak{A}, s \models \varphi$ holds (uses the axiom of choice).

Toy examples

- $\mathfrak{A}, s \models_{\mathrm{GTS}} \forall x \exists y (x = y)$ is always true.
- $\mathfrak{A}, s \models_{\mathrm{GTS}} \forall x \exists y (\neg x = y)$ is true if the model has at least two elements.

Games of perfect information

- Both players in the plays of GTS for $\mathcal{FO}$, when making a move, remember all previous moves made by both players in that play.
- Next we move to a variant of GTS in which this is not the case.

PART 2

GTS semantics for IF-logic

Independence-Friendly logic (Hintikka and Sandu 1989)

Syntax for IF-logic:

$$\varphi ::= x = y \mid \neg x = y \mid R(\overline{x}) \mid \neg R(\overline{x}) \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \exists x/W\varphi \mid \forall x\varphi$$

where W is a set of variable symbols.

- The reading of $\exists x/\{y\}\varphi$ is that there exists x independently of y such that φ holds.

Independence-Friendly logic (Hintikka and Sandu 1989)

Syntax for IF-logic:

$$\varphi ::= x = y \mid \neg x = y \mid R(\overline{x}) \mid \neg R(\overline{x}) \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \exists x/W\varphi \mid \forall x\varphi$$

where W is a set of variable symbols.

- The reading of $\exists x/\{y\}\varphi$ is that there exists x independently of y such that φ holds.
- This kind of language is used, e.g., in many definitions in mathematics.
  - Continuity: a function $f : D \to \mathbb{R}$ is continuous, if for all $a \in D$ and for all $\epsilon > 0$ there exists $\delta > 0$ such that for all $x \in D$, if $|x - a| < \delta$, then $|f(x) - f(a)| < \epsilon$.
  - Uniformly continuity: a function $f : D \to \mathbb{R}$ is uniformly continuous, if for all $a \in D$ and for all $\epsilon > 0$ there exists $\delta > 0$ independent of a such that for all $x \in D$, if $|x - a| < \delta$, then $|f(x) - f(a)| < \epsilon$.

Game theoretic semantics of $\mathcal{IF}$

- The game $G_{\mathrm{IF}}(\varphi, (\mathfrak{A}, s))$ is a two-player game of imperfect information.
- Players: $(\forall)$ Abelard and $(\exists)$ Eloise.
- Abelard wants to establish that φ is not true in $\mathfrak{A}, s$, while Eloise wishes to show that φ is true in $\mathfrak{A}, s$.
- Abelard controls $\forall$ and $\wedge$, while Eloise controls $\exists$ and $\vee$.

Rules of $G_{\mathrm{IF}}(\varphi, (\mathfrak{A}, s))$

- Plays of $G_{\mathrm{IF}}(\varphi, (\mathfrak{A}, s))$ and $G(\varphi, (\mathfrak{A}, s))$ are identical.

Rules of $G_{\mathrm{IF}}(\varphi, (\mathfrak{A}, s))$

- Plays of $G_{\mathrm{IF}}(\varphi, (\mathfrak{A}, s))$ and $G(\varphi, (\mathfrak{A}, s))$ are identical.
- A *play* of the game starts from the position $(\varphi, (\mathfrak{A}, s))$.
- In position $(\psi_1 \wedge \psi_2, (\mathfrak{A}, s))$, Abelard chooses $i \in \{1, 2\}$ and the game continues from position $(\psi_i, (\mathfrak{A}, s))$.
- In position $(\psi_1 \vee \psi_2, (\mathfrak{A}, s))$, Eloise chooses $i \in \{1, 2\}$ and the game continues from position $(\psi_i, (\mathfrak{A}, s))$.
- In position $(\forall x \psi, (\mathfrak{A}, s))$, Abelard chooses $a \in A$ and the game continues from position $(\psi, (\mathfrak{A}, s(x \mapsto a)))$.
- In position $(\exists x / W \psi, (\mathfrak{A}, s))$, Eloise chooses $a \in A$ and the game continues from position $(\psi, (\mathfrak{A}, s(x \mapsto a)))$.
- The game is played until a position $(\psi, (\mathfrak{A}, t))$ is reached, where ψ is a literal.

End Game

- Eloise wins the play of the game if
  - $(x = y, (\mathfrak{A}, t))$ is reached and $t(x) = t(y)$,
  - $(\neg x = y, (\mathfrak{A}, t))$ is reached and $t(x) \neq t(y)$
  - $(P(x), (\mathfrak{A}, t))$ is reached and $t(x) \in P$,
  - $(\neg P(x), (\mathfrak{A}, t))$ is reached and $t(x) \notin P$,
  - $(R(x, y), (\mathfrak{A}, t))$ is reached and $(t(x), t(y)) \in R$,
  - $\neg(R(x, y), (\mathfrak{A}, t))$ is reached and $(t(x), t(y)) \notin R$,
- otherwise Abelard wins the play.

To win the Game it is not enough to win a single play.

Strategies

- A strategy for Eloise in the game $G_{\mathrm{IF}}(\varphi, (\mathfrak{A}, s))$ is a function that maps every possible position that she controls and that may occur in some play of the game to a single next position of the play.

Strategies

- A strategy for Eloise in the game $G_{\mathrm{IF}}(\varphi, (\mathfrak{A}, s))$ is a function that maps every possible position that she controls and that may occur in some play of the game to a single next position of the play.
- A strategy function f of Eloise is consistent (with her knowledge) if the following holds:
  If $(\exists x/W\varphi, (\mathfrak{A}, t))$ and $(\exists x/W\varphi, (\mathfrak{A}, t'))$ are positions in the game such that $t(y) = t'(y)$ for every $y \in \mathrm{Var} \setminus W$, then
  $f((\exists x/W\varphi, (\mathfrak{A}, t)) = f((\exists x/W\varphi, (\mathfrak{A}, t'))$.

Strategies

- A strategy for Eloise in the game $G_{\mathrm{IF}}(\varphi, (\mathfrak{A}, s))$ is a function that maps every possible position that she controls and that may occur in some play of the game to a single next position of the play.
- A strategy function f of Eloise is consistent (with her knowledge) if the following holds:
  If $(\exists x/W\varphi, (\mathfrak{A}, t))$ and $(\exists x/W\varphi, (\mathfrak{A}, t'))$ are positions in the game such that $t(y) = t'(y)$ for every $y \in \mathrm{Var} \setminus W$, then
  $f((\exists x/W\varphi, (\mathfrak{A}, t)) = f((\exists x/W\varphi, (\mathfrak{A}, t'))$.
- A stategy of Eloise is a winning strategy if she wins every play of the game in which she plays according to her strategy.

Winner of the game

- Eloise wins the game $G_{\mathrm{IF}}(\varphi, (\mathfrak{A}, s))$ if she has a winning consistent strategy for that game.
- A formula φ is true in $\mathfrak{A}, s$ under GTS-IF (denoted by $\mathfrak{A}, s \models_{\mathrm{GTS-IF}} \varphi$) if Eloise wins the game $G_{\mathrm{IF}}(\varphi, (\mathfrak{A}, s))$.

Winner of the game

- Eloise wins the game $G_{\mathrm{IF}}(\varphi, (\mathfrak{A}, s))$ if she has a winning consistent strategy for that game.
- A formula φ is true in $\mathfrak{A}, s$ under GTS-IF (denoted by $\mathfrak{A}, s \models_{\mathrm{GTS-IF}} \varphi$) if Eloise wins the game $G_{\mathrm{IF}}(\varphi, (\mathfrak{A}, s))$.
- Hintikka claimed that there does not exists a compostional (Taski-style) semantics for $\mathcal{IF}$-logic.

Winner of the game

- Eloise wins the game $G_{\mathrm{IF}}(\varphi, (\mathfrak{A}, s))$ if she has a winning consistent strategy for that game.
- A formula φ is true in $\mathfrak{A}, s$ under GTS-IF (denoted by $\mathfrak{A}, s \models_{\mathrm{GTS-IF}} \varphi$) if Eloise wins the game $G_{\mathrm{IF}}(\varphi, (\mathfrak{A}, s))$.
- Hintikka claimed that there does not exists a compostional (Taski-style) semantics for $\mathcal{IF}$-logic.
- Hodges (1997) presented a Tarski-style semantics for $\mathcal{IF}$-logic that instead of assignments use sets of assignments as satisfying elements (team semantics).

Toy examples

- $\mathfrak{A}, s \models_{\mathrm{GTS-IF}} \forall x \exists y/\{x\}(x = y)$
- $\mathfrak{A}, s \models_{\mathrm{GTS-IF}} \forall x \exists y/\{x\}(\neg x = y)$

Toy examples

- $\mathfrak{A}, s \models_{\mathrm{GTS-IF}} \forall x \exists y/\{x\}(x = y)$ is true only if A is a singleton set.
- $\mathfrak{A}, s \models_{\mathrm{GTS-IF}} \forall x \exists y/\{x\}(\neg x = y)$ is never true.

PART 3

Team semantics for FO and IF

First-order logic

Grammar of first-order logic $\mathcal{FO}$ in negation normal form:

$$\varphi ::= x = y \mid \neg(x = y) \mid R(\overline{x}) \mid \neg R(\overline{x}) \mid (\varphi \vee \varphi) \mid (\varphi \wedge \varphi) \mid \exists x \varphi(x) \mid \forall x \varphi(x)$$

A team of an $\mathcal{FO}$-structure $\mathfrak{A}$ is any set X of assignments $s : \mathrm{VAR} \to A$ with a common domain VAR of $\mathcal{FO}$ variables.

First-order logic

Grammar of first-order logic $\mathcal{FO}$ in negation normal form:

$$\varphi ::= x = y \mid \neg(x = y) \mid R(\overline{x}) \mid \neg R(\overline{x}) \mid (\varphi \vee \varphi) \mid (\varphi \wedge \varphi) \mid \exists x \varphi(x) \mid \forall x \varphi(x)$$

A team of an $\mathcal{FO}$-structure $\mathfrak{A}$ is any set X of assignments $s : \mathrm{VAR} \to A$ with a common domain VAR of $\mathcal{FO}$ variables.

We want to define team semantics for $\mathcal{FO}$ s.t. we have the following property (*flattness*):

If φ is an $\mathcal{FO}$-formula, $\mathfrak{A}$ a first-order structure, and X a set of assignments:

$$\mathfrak{A} \models_X \varphi \iff \forall s \in X : \mathfrak{A}, s \models \varphi.$$

Team semantics for first-order logic

Recall that a team is a set of first-order assignments with a common domain.

$$
\begin{aligned}
\mathfrak{A}, s \models R(\overline{x}) &\quad\Leftrightarrow\quad s(\overline{x}) \in R^{\mathfrak{A}}\\
\mathfrak{A}, s \models \neg R(\overline{x}) &\quad\Leftrightarrow\quad s(\overline{x}) \notin R^{\mathfrak{A}}\\
\mathfrak{A}, s \models \varphi \wedge \psi &\quad\Leftrightarrow\quad \mathfrak{A}, s \models \varphi \text{ and } \mathfrak{A}, s \models \psi\\
\mathfrak{A}, s \models \varphi \vee \psi &\quad\Leftrightarrow\quad \mathfrak{A}, s \models \varphi \text{ or } \mathfrak{A}, s \models \psi\\
\mathfrak{A}, s \models \forall x \varphi &\quad\Leftrightarrow\quad \mathfrak{A}, s(x \mapsto a) \models \varphi \text{ for all } a \in A\\
\mathfrak{A}, s \models \exists x \varphi &\quad\Leftrightarrow\quad \mathfrak{A}, s(x \mapsto a) \models \varphi \text{ for some } a \in A
\end{aligned}
$$

Team semantics for first-order logic

Recall that a team is a set of first-order assignments with a common domain.

$$
\begin{aligned}
\mathfrak{A} \models_X R(\overline{x}) \quad &\Leftrightarrow \quad \forall s \in X : s(\overline{x}) \in R^{\mathfrak{A}} \\
\mathfrak{A} \models_X \neg R(\overline{x}) \quad &\Leftrightarrow \quad \forall s \in X : s(\overline{x}) \notin R^{\mathfrak{A}} \\
\mathfrak{A}, s \models \varphi \wedge \psi \quad &\Leftrightarrow \quad \mathfrak{A}, s \models \varphi \text{ and } \mathfrak{A}, s \models \psi \\
\mathfrak{A}, s \models \varphi \vee \psi \quad &\Leftrightarrow \quad \mathfrak{A}, s \models \varphi \text{ or } \mathfrak{A}, s \models \psi \\
\mathfrak{A}, s \models \forall x \varphi \quad &\Leftrightarrow \quad \mathfrak{A}, s(x \mapsto a) \models \varphi \text{ for all } a \in A \\
\mathfrak{A}, s \models \exists x \varphi \quad &\Leftrightarrow \quad \mathfrak{A}, s(x \mapsto a) \models \varphi \text{ for some } a \in A
\end{aligned}
$$

Team semantics for first-order logic

Recall that a team is a set of first-order assignments with a common domain.

$$
\begin{aligned}
\mathfrak{A} \models_X R(\overline{x}) \quad &\Leftrightarrow \quad \forall s \in X : s(\overline{x}) \in R^{\mathfrak{A}} \\
\mathfrak{A} \models_X \neg R(\overline{x}) \quad &\Leftrightarrow \quad \forall s \in X : s(\overline{x}) \notin R^{\mathfrak{A}} \\
\mathfrak{A} \models_X \varphi \wedge \psi \quad &\Leftrightarrow \quad \mathfrak{A} \models_X \varphi \text{ and } \mathfrak{A} \models_X \psi \\
\mathfrak{A}, s \models \varphi \vee \psi \quad &\Leftrightarrow \quad \mathfrak{A}, s \models \varphi \text{ or } \mathfrak{A}, s \models \psi \\
\mathfrak{A}, s \models \forall x \varphi \quad &\Leftrightarrow \quad \mathfrak{A}, s(x \mapsto a) \models \varphi \text{ for all } a \in A \\
\mathfrak{A}, s \models \exists x \varphi \quad &\Leftrightarrow \quad \mathfrak{A}, s(x \mapsto a) \models \varphi \text{ for some } a \in A
\end{aligned}
$$

Team semantics for first-order logic

Recall that a team is a set of first-order assignments with a common domain.

$$
\begin{aligned}
\mathfrak{A} \models_X R(\overline{x}) \quad &\Leftrightarrow \quad \forall s \in X : s(\overline{x}) \in R^{\mathfrak{A}} \\
\mathfrak{A} \models_X \neg R(\overline{x}) \quad &\Leftrightarrow \quad \forall s \in X : s(\overline{x}) \notin R^{\mathfrak{A}} \\
\mathfrak{A} \models_X \varphi \wedge \psi \quad &\Leftrightarrow \quad \mathfrak{A} \models_X \varphi \text{ and } \mathfrak{A} \models_X \psi \\
\mathfrak{A} \models_X \varphi \vee \psi \quad &\Leftrightarrow \quad \mathfrak{A} \models_Y \varphi \text{ and } \mathfrak{A} \models_Z \psi \text{ for some } Y \cup Z = X \\
\mathfrak{A}, s \models \forall x \varphi \quad &\Leftrightarrow \quad \mathfrak{A}, s(x \mapsto a) \models \varphi \text{ for all } a \in A \\
\mathfrak{A}, s \models \exists x \varphi \quad &\Leftrightarrow \quad \mathfrak{A}, s(x \mapsto a) \models \varphi \text{ for some } a \in A
\end{aligned}
$$

Team semantics for first-order logic

Recall that a team is a set of first-order assignments with a common domain.

$$
\begin{aligned}
\mathfrak{A} \models_X R(\overline{x}) &\quad\Leftrightarrow\quad \forall s \in X : s(\overline{x}) \in R^{\mathfrak{A}} \\
\mathfrak{A} \models_X \neg R(\overline{x}) &\quad\Leftrightarrow\quad \forall s \in X : s(\overline{x}) \notin R^{\mathfrak{A}} \\
\mathfrak{A} \models_X \varphi \wedge \psi &\quad\Leftrightarrow\quad \mathfrak{A} \models_X \varphi \text{ and } \mathfrak{A} \models_X \psi \\
\mathfrak{A} \models_X \varphi \vee \psi &\quad\Leftrightarrow\quad \mathfrak{A} \models_Y \varphi \text{ and } \mathfrak{A} \models_Z \psi \text{ for some } Y \cup Z = X \\
\mathfrak{A} \models_X \forall x \varphi &\quad\Leftrightarrow\quad \mathfrak{A} \models_{X[A/x]} \varphi \\
\mathfrak{A}, s \models \exists x \varphi &\quad\Leftrightarrow\quad \mathfrak{A}, s(x \mapsto a) \models \varphi \text{ for some } a \in A
\end{aligned}
$$

Where $X[A/x] := \{s(x \mapsto a) \mid s \in X, a \in A\}$

Team semantics for first-order logic

Recall that a team is a set of first-order assignments with a common domain.

$$
\begin{aligned}
\mathfrak{A} \models_X R(\overline{x}) &\quad\Leftrightarrow\quad \forall s \in X : s(\overline{x}) \in R^{\mathfrak{A}} \\
\mathfrak{A} \models_X \neg R(\overline{x}) &\quad\Leftrightarrow\quad \forall s \in X : s(\overline{x}) \notin R^{\mathfrak{A}} \\
\mathfrak{A} \models_X \varphi \wedge \psi &\quad\Leftrightarrow\quad \mathfrak{A} \models_X \varphi \text{ and } \mathfrak{A} \models_X \psi \\
\mathfrak{A} \models_X \varphi \vee \psi &\quad\Leftrightarrow\quad \mathfrak{A} \models_Y \varphi \text{ and } \mathfrak{A} \models_Z \psi \text{ for some } Y \cup Z = X \\
\mathfrak{A} \models_X \forall x \varphi &\quad\Leftrightarrow\quad \mathfrak{A} \models_{X[A/x]} \varphi \\
\mathfrak{A} \models_X \exists x \varphi &\quad\Leftrightarrow\quad \mathfrak{A} \models_{X[F/x]} \varphi \text{ for some } F : X \to A
\end{aligned}
$$

Where $X[A/x] := \{s(x \mapsto a) \mid s \in X, a \in A\}$ and
$X[F/x] := \{s(x \mapsto F(x)) \mid s \in X\}$

Team semantics for first-order logic

Recall that a team is a set of first-order assignments with a common domain.

$$
\begin{aligned}
\mathfrak{A} \models_X R(\overline{x}) &\Leftrightarrow \forall s \in X : s(\overline{x}) \in R^{\mathfrak{A}} \\
\mathfrak{A} \models_X \neg R(\overline{x}) &\Leftrightarrow \forall s \in X : s(\overline{x}) \notin R^{\mathfrak{A}} \\
\mathfrak{A} \models_X \varphi \wedge \psi &\Leftrightarrow \mathfrak{A} \models_X \varphi \text{ and } \mathfrak{A} \models_X \psi \\
\mathfrak{A} \models_X \varphi \vee \psi &\Leftrightarrow \mathfrak{A} \models_Y \varphi \text{ and } \mathfrak{A} \models_Z \psi \text{ for some } Y \cup Z = X \\
\mathfrak{A} \models_X \forall x \varphi &\Leftrightarrow \mathfrak{A} \models_{X[A/x]} \varphi \\
\mathfrak{A} \models_X \exists x \varphi &\Leftrightarrow \mathfrak{A} \models_{X[F/x]} \varphi \text{ for some } F : X \to A
\end{aligned}
$$

Where $X[A/x] := \{s(x \mapsto a) \mid s \in X, a \in A\}$ and $X[F/x] := \{s(x \mapsto F(x)) \mid s \in X\}$

For every $\mathcal{FO}$-formula φ the following holds:

$$\mathfrak{A} \models_X \varphi \iff \forall s \in X : \mathfrak{A}, s \models \varphi.$$

Team semantics for independence-friendly logic

- Otherwise the same as for $\mathcal{FO}$, but an additional rule for $\exists x/W$ is needed.

Team semantics for independence-friendly logic

- Otherwise the same as for $\mathcal{FO}$, but an additional rule for $\exists x/W$ is needed.
- A function $F : X \to A$ is W-independent if for every $s, t \in X$ the implication

$$(\forall x \in \mathrm{dom}(X) \setminus W : s(x) = t(x)) \Rightarrow F(s) = F(t)$$

holds (above $\mathrm{dom}(X)$ denotes the domain of the assignments of X).

Team semantics for independence-friendly logic

- Otherwise the same as for $\mathcal{FO}$, but an additional rule for $\exists x/W$ is needed.
- A function $F : X \to A$ is W-independent if for every $s, t \in X$ the implication

$$(\forall x \in \mathrm{dom}(X) \setminus W : s(x) = t(x)) \Rightarrow F(s) = F(t)$$

  holds (above $\mathrm{dom}(X)$ denotes the domain of the assignments of X).
- Now we have the following rule for $\exists x/W$:

$$\mathfrak{A} \models_X \exists x/W\varphi \quad \Leftrightarrow \quad \mathfrak{A} \models_{X[F/x]} \varphi \text{ for some } W\text{-independent } F : X \to A$$

Properties of IF-logic

- Downward closure holds: $(\mathfrak{A} \models_X \varphi$ and $Y \subseteq X) \Rightarrow \mathfrak{A} \models_Y \varphi$.

Properties of IF-logic

- Downward closure holds: $(\mathfrak{A} \models_X \varphi$ and $Y \subseteq X) \Rightarrow \mathfrak{A} \models_Y \varphi$.
- $\mathcal{IF}$ is strictly more expressive than $\mathcal{FO}$.
- Violates flattness: $\mathfrak{A} \models_X \varphi$ iff $\forall s \in X : \mathfrak{A}, s \models \varphi$ may not hold.

Properties of IF-logic

- Downward closure holds: $(\mathfrak{A} \models_X \varphi$ and $Y \subseteq X) \Rightarrow \mathfrak{A} \models_Y \varphi$.
- $\mathcal{IF}$ is strictly more expressive than $\mathcal{FO}$.
- Violates flattness: $\mathfrak{A} \models_X \varphi$ iff $\forall s \in X : \mathfrak{A}, s \models \varphi$ may not hold.
- Violates locality: Truth of an IF-formula may depend on the interpretations of variables that do not occur in the formula!

Toy examples

- $\forall x \exists y (\exists z/\{x\} x = z \land \exists z'/\{x, y\} \neg x = z')$
- $\exists y/\{x\} x = y$

Toy examples

- $\forall x \exists y (\exists z/\{x\} x = z \wedge \exists z'/\{x, y\} \neg x = z')$ is true if the model is infinite.
- $\exists y/\{x\} x = y$

Toy examples

- $\forall x \exists y (\exists z/\{x\} x = z \wedge \exists z'/\{x, y\} \neg x = z')$ is true if the model is infinite.
- $\exists y/\{x\} x = y$ in a team X with domain $\{x, y, z\}$ depends on the values of z in X, although z does not occur in the formula (signalling).

PART 4

Team semantics and database theory and dependence logic

Team semantics via database theoretic spectacles

- A team is a set of assignments that have a common domain of variables.
- A team is a database table.
  - Variables correspond to attributes.
  - Assignments correspond to records.
- Dependency notions of database theory give rise to novel atomic formulae.
  - Functional dependence gives rise to dependence atoms $\mathrm{dep}(x_1, \ldots, x_n)$.
  - Inclusion dependence gives rise to inclusion atoms $\overline{x} \subseteq \overline{y}$.
  - Embedded multivalued dependency gives rise to independence atoms $\overline{y} \perp_{\overline{x}} \overline{z}$.

Dependence logic (Väänänen 2007)

In FO, formulas are formed using connectives $\vee$, $\wedge$, $\neg$, and quantifiers $\exists$ and $\forall$.

Definition

Dependence logic $\mathcal{D}$ extends the syntax of FO by dependence atoms

$$\mathrm{dep}(x_1, \ldots, x_n, y) .$$

Dependence logic (Väänänen 2007)

In FO, formulas are formed using connectives $\vee$, $\wedge$, $\neg$, and quantifiers $\exists$ and $\forall$.

Definition

Dependence logic $\mathcal{D}$ extends the syntax of FO by dependence atoms

$$\mathrm{dep}(x_1, \ldots, x_n, y)\,.$$

The reading of $\mathrm{dep}(x_1, \ldots, x_n, y)$ is that the value for y is determined by the values of the variables $x_1 \ldots, x_n$.

Interpretation of dependence atoms

Let $\mathfrak{A}$ be a model and X a team with co-domain A and domain V s.t. $\{x_1, \ldots, x_n, y\} \subseteq V$.

$\mathfrak{A} \models_X \mathrm{dep}(x_1, \ldots, x_n, y)$, if and only if, for all $s, s' \in X$:

$$\bigwedge_{1 \leq i \leq n} s(x_i) = s'(x_i) \implies s(y) = s'(y).$$

Examples of teams

We may think of the variables x_i as attributes of a database such as $x_0 = \mathrm{SALARY}$ and $x_2 = \mathrm{ID\ NUMBER}$.

	x_0	.	.	.	x_n
s_0	$a_{0,m}$	.	.	.	$a_{n,m}$
.					
.					
.					
s_m	$a_{0,m}$	.	.	.	$a_{n,m}$

Then *dependence atom* $\mathrm{dep}(x_2, x_0)$ expresses the functional dependence

$$\mathrm{ID\ NUMBER} \rightarrow \ \mathrm{SALARY}.$$

Toy example

- $\mathfrak{A} \models_X \forall x(\mathrm{dep}(x) \vee \mathrm{dep}(x) \vee \mathrm{dep}(x))$ holds if and only if $|A| \leq 3$.
- There is a simple formula expressing that $|A|$ is even.

Dependence logic vs. Independence-Friendly logic

- Idea in $\mathcal{IF}$: in $\exists x/W\varphi$ the value for x is picked independently on the values of the variables in W.
- Idea in $\mathcal{D}$: $\mathrm{dep}(x_1, \ldots, x_n, y)$ states that the value for y depends only on variables $x_1 \ldots, x_n$

Dependence logic vs. Independence-Friendly logic

- Idea in $\mathcal{IF}$: in $\exists x/W\varphi$ the value for x is picked independently on the values of the variables in W.
- Idea in $\mathcal{D}$: $\mathrm{dep}(x_1, \ldots, x_n, y)$ states that the value for y depends only on variables $x_1 \ldots, x_n$
- $\mathcal{D}$ is an adaptation of $\mathcal{IF}$ in which we shift from declaring independences in quantification of variables to declaring dependences by atomic statements.
- For $\mathcal{D}$ locality holds!

Dependence logic vs. Independence-Friendly logic

- Idea in $\mathcal{IF}$: in $\exists x/W\varphi$ the value for x is picked independently on the values of the variables in W.
- Idea in $\mathcal{D}$: $\mathrm{dep}(x_1, \ldots, x_n, y)$ states that the value for y depends only on variables $x_1 \ldots, x_n$
- $\mathcal{D}$ is an adaptation of $\mathcal{IF}$ in which we shift from declaring independences in quantification of variables to declaring dependences by atomic statements.
- For $\mathcal{D}$ locality holds!
- It comes with no surprise that the expressive powers of the logics coincide.

PART 5

Expressive power of D and IF

Expressive power of $\mathcal{IF}$ and $\mathcal{D}$

Theorem

For every formula $\varphi \in \mathcal{IF}$ *that uses only variables* $\{x_1, \ldots, x_n\}$ *there exists a formula* $\varphi^* \in \mathcal{D}$ *such that for every model* $\mathfrak{A}$ *and team* X *of* $\mathfrak{A}$*, where* $\mathrm{Dom}(X) = \{x_1, \ldots x_n\}$*, it holds that*

$$\mathfrak{A} \models_X \varphi \Leftrightarrow \mathfrak{A} \models_X \varphi^*.$$

Theorem

For every formula $\varphi \in \mathcal{D}$ *that uses only variables* $\{x_1, \ldots, x_n\}$ *there exists a formula* $\varphi^* \in \mathcal{IF}$ *such that for every model* $\mathfrak{A}$ *and team* X *of* $\mathfrak{A}$*, where* $\mathrm{Dom}(X) = \{x_1, \ldots x_n\}$*, it holds that*

$$\mathfrak{A} \models_X \varphi \Leftrightarrow \mathfrak{A} \models_X \varphi^*.$$

Existential second-order logic

Formulas of existential second-order logic ($\mathcal{ESO}$) over vocabulary τ are of the form

$$\exists R_1 \ldots \exists R_n \varphi,$$

where R_i:s are relation variables (of some arity) and φ is a formula of first-order logic over the vocabulary $\tau \cup \{R_1, \ldots R_n\}$.

Existential second-order logic

Formulas of existential second-order logic ($\mathcal{ESO}$) over vocabulary τ are of the form

$$\exists R_1 \ldots \exists R_n \varphi,$$

where R_i:s are relation variables (of some arity) and φ is a formula of first-order logic over the vocabulary $\tau \cup \{R_1, \ldots R_n\}$.

Semantics for $\mathcal{ESO}$ is defined analogously to $\mathcal{FO}$, with the additional rule:

$$\mathfrak{A}, s \models \exists R \varphi \quad \Leftrightarrow \quad \mathfrak{A}, s(R \mapsto B) \models \varphi \text{ for some } B \subseteq A^n,$$

where R is a relation variable of arity n.

Expressive Power

For sentences the expressive power of $\mathcal{D}$ and $\mathcal{IF}$ coincide with the expressive power of $\mathcal{ESO}$ (Ederton 1970, Walkoe 1970; the result for partially ordered quantifiers).

Expressive Power

For sentences the expressive power of $\mathcal{D}$ and $\mathcal{IF}$ coincide with the expressive power of $\mathcal{ESO}$ (Ederton 1970, Walkoe 1970; the result for partially ordered quantifiers).

Dependence logic (and thus also $\mathcal{IF}$-logic) defines all downward closed ESO properties of teams.

Theorem (Kontinen, Väänänen 2009)

For every sentence $\psi \in \mathcal{ESO}[\tau \cup \{R\}]$, in which R appears only negatively, there is $\phi(y_1, \ldots, y_k) \in \mathcal{D}[\tau]$ s.t. for all $\mathfrak{A}$ and $X \neq \emptyset$ with domain $\{y_1, \ldots, y_k\}$

$$\mathfrak{A} \models_X \phi \iff (\mathfrak{A}, R := X(\bar{y})) \models \psi.$$

Related works

The framework of team semantics has been introduced to many areas of logic and variety of results have been obtained.

- In first-order setting: dependence logic, inclusion logic, independence logic, probabilistic variants, etc.
- Modal and propostional variants: modal dependence logic, propositional dependence logics etc.
- Expressive power, axiomatizations, definability, frame definabily etc. have been studied.
- Connections to dependency theory of database theory.
- Connections to notions of independence in statistics.

THANKS!