Polyteam Semantics

Jonni Virtema

Hasselt University, Belgium
jonni.virtema@gmail.com

Joint work with Miika Hannula (University of Auckland) and Juha Kontinen (University of Helsinki)

January 11, 2018

Team Semantics: Motivation and History

Logical modelling of uncertainty, imperfect information, and different notions of dependence, such as functional dependence and independence, from application fields: statistics (probabilistic independence), database theory (database dependencies), social choice theory (arrows theorem), etc.

Historical development:

- First-order logic and Skolem functions.
- Branching quantifiers by Henkin 1959.
- Independence-friendly logic by Hintikka and Sandu 1989.
- Compositional semantics for independence-friendly logic by Hodges 1997. (Origin of team semantics.)
- Dependence logic 2007 and modal dependence logic 2008 by Väänänen.
- Introduction of other dependency notions to team semantics such as inclusion, exclusion, and independence. Galliani, Grädel, Väänänen.
- Approximate dependence by Väänänen 2014 and multiteam semantics by Durand et al. 2016.

Team Semantics: Motivation and History

Logical modelling of uncertainty, imperfect information, and different notions of dependence, such as functional dependence and independence, from application fields: statistics (probabilistic independence), database theory (database dependencies), social choice theory (arrows theorem), etc.

Historical development:

- First-order logic and Skolem functions.
- Branching quantifiers by Henkin 1959.
- Independence-friendly logic by Hintikka and Sandu 1989.
- Compositional semantics for independence-friendly logic by Hodges 1997. (Origin of team semantics.)
- Dependence logic 2007 and modal dependence logic 2008 by Väänänen.
- Introduction of other dependency notions to team semantics such as inclusion, exclusion, and independence. Galliani, Grädel, Väänänen.
- Approximate dependence by Väänänen 2014 and multiteam semantics by Durand et al. 2016.

First-Order Team Semantics (via database theoretic spectacles)

- A team is a set of assignments that have a common domain of variables.
- A team can be seen as a database table.
  - Variables correspond to attributes.
  - Assignments correspond to records.
- Dependency notions of database theory give rise to novel atomic formulae.
  - Functional dependence corresponds to dependence atoms $=(x_1, \ldots, x_n, y)$.
  - Inclusion dependence corresponds 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

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

Definition

Dependence logic $\mathrm{FO}(\mathrm{dep})$ extends the syntax of FO by dependence atoms

$$=\!(x_1, \ldots, x_n)\,.$$

We consider also independence and inclusion atoms (and the corresponding logics) that replace dependence atoms respectively by

$$\overline{y} \perp_{\overline{x}} \overline{z} \text{ and } \overline{x} \subseteq \overline{y}.$$

Assignments and Teams

The semantics of dependence logic is defined using the notion of a team.

Teams:

Let A be a set and $V = \{x_1, \dots, x_k\}$ a finite set of variables. A *team* X with domain V is a set of assignments

$$s \colon V \to A.$$

A is called the co-domain of X (the universe of a model).

Interpretation of Dependence Atoms

Let $\mathfrak{A}$ be a structure and X a team.

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

$$\bigwedge_{0<i<n} s(x_i) = s'(x_i) \implies s(x_n) = s'(x_n).$$

Interpreting Inclusion and Independence Atoms

Inclusion atoms:
$\mathfrak{A} \models_X \overline{x} \subseteq \overline{y}$, if and only if, for all $s \in X$ there exists $s' \in X$ s.t. $s(\overline{x}) = s'(\overline{y})$.

Independence atoms:
$\mathfrak{A} \models_X \overline{y} \perp_{\overline{x}} \overline{z}$, iff, for all $s, s' \in X$: if $s(\overline{x}) = s'(\overline{x})$ then there exists $s'' \in X$ such that

- $s''(\overline{x}) = s(\overline{x})$,
- $s''(\overline{y}) = s(\overline{y})$,
- $s''(\overline{z}) = s'(\overline{z})$.

Examples of teams

We may think of the variables x_i as attributes of a database such as $x_0 = \text{SALARY}$ and $x_2 = \text{JOB TITLE}$.

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

Then dependence atom $=\!(x_2, x_0)$ expresses the functional dependence

$$\text{JOB TITLE} \to \text{ SALARY}.$$

Team semantics for first-order logic

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

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

Team semantics for first-order logic

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

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

Team semantics for first-order logic

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

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

Team semantics for first-order logic

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

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

Team semantics for first-order logic

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

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

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

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

Expressive Power

With respect to sentences

- Dependence logic and independence logic corresponds to existential second-order logic ESO and thus the complexity class NP (Väänänen 2007, Grädel & Väänänen 2010).
- Inclusion logic corresponds to the positive greatest fixed point logic GFP^{+} and thus the complexity class P on ordered structres (Galliani & Hella 13).

Expressive Power

Dependence logic defines all downward closed ESO properties of teams.

Theorem (Kontinen, Väänänen 2009)

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

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

Independence logic defines all ESO properties of teams (Galliani 2012).

Axiomatisations in team semantics

- No axiomatisations in general due to high expressive powers.
- First-order consequences of dependence logic formulae can be axiomatised (Kontinen, Väänänen 2013).
- Entailment of conjunctions of atoms (dependence, inclusion, independence etc.) has been axiomatised.
- (Axiomatisation exist for modal variants of the related logics.)

Amstrong's Axioms for Functional Dependence

This inference system consists of only three rules which we depict below using the standard notation for functional dependencies, i.e., $X \to Y$ denotes that an attribute set X functionally determines another attribute set Y.

Definition (Armstrong 1974)

- Reflexivity: If $Y \subseteq X$, then $X \to Y$.
- Augmentation: if $X \to Y$, then $XZ \to YZ$
- Transitivity: if $X \to Y$ and $Y \to Z$, then $X \to Z$.

The same axiomatization works for dependence atoms $=(\overline{x}, y)$ when we add some rules that permutes and adds/removes duplicates to/from $\overline{x}$.

Amstrong's Axioms for Functional Dependence

This inference system consists of only three rules which we depict below using the standard notation for functional dependencies, i.e., $X \to Y$ denotes that an attribute set X functionally determines another attribute set Y.

Definition (Armstrong 1974)

- Reflexivity: If $Y \subseteq X$, then $X \to Y$.
- Augmentation: if $X \to Y$, then $XZ \to YZ$
- Transitivity: if $X \to Y$ and $Y \to Z$, then $X \to Z$.

The same axiomatization works for dependence atoms $=\!(\overline{x}, y)$ when we add some rules that permutes and adds/removes duplicates to/from $\overline{x}$.

From Teams to Polyteams

- Team semantics is a framework well suited to express different dependency notions, e.g., studied in database theory, when restricted to the unirelational case.
- However dependencies between different tables cannot be expressed in this framework. E.g., a *source-to-target* embedded dependency $\forall \overline{x}\big(\phi(\overline{x}) \rightarrow \exists \overline{y}\psi(\overline{x}, \overline{y})\big)$ is an FO-sentence where ϕ is a formula over source relations and ψ over target relations.
- We next define a generalisation of team semantics in which we replace teams by tuples of teams to be able model dependencies between different tables.

Polyteams

For $i \in \mathbb{N}$, let $\mathrm{Var}(i)$ denote a distinct countable set of FO variable symbols.

Definition

A tuple $\overline{X} = (X_i)_{i\in\mathbb{N}}$ is a *polyteam* of $\mathfrak{A}$ with domain $\overline{D} = (D_i)_{i\in\mathbb{N}}$, if

- $D_i \subseteq \mathrm{Var}(i)$ for all $i \in \mathbb{N}$, and
- X_i is a team with domain D_i and co-domain A for each $i \in \mathbb{N}$.

We identify $\overline{X}$ with $(X_1, \ldots, X_n)$ if X_i is empty for all i greater than n.
We write x^i, y^i, $\overline{x}^i$, etc., to denote variables from $\mathrm{Var}(i)$.

Polyatoms

Poly-Inclusion atoms:
$\mathfrak{A} \models_{\overline{X}} \overline{x}^i \subseteq \overline{y}^j$, iff, for all $s \in X_i$ there exists $s' \in X_j$ s.t. $s(\overline{x}^i) = s'(\overline{y}^j)$.

Poly-Dependence atoms:
Let $\overline{x}^i\overline{y}^i$ and $\overline{u}^j\overline{v}^j$ be sequences of variables s.t. $|\overline{x}^i| = |\overline{u}^j|$ and $|\overline{y}^i| = |\overline{u}^j|$.

$$\mathfrak{A} \models_{\overline{X}} =(\overline{x}^i, \overline{y}^i / \overline{u}^j, \overline{v}^j) \Leftrightarrow \forall s \in X_i \forall s' \in X_j : s(\overline{x}^i) = s'(\overline{u}^j) \text{ implies } s(\overline{y}^i) = s'(\overline{v}^j).$$

Note that the atom $=(\overline{x}, \overline{y} / \overline{x}, \overline{y})$ corresponds to the dependence atom $=(\overline{x}, \overline{y})$.

Polyatoms

Poly-Inclusion atoms:
$\mathfrak{A} \models_{\overline{X}} \overline{x}^i \subseteq \overline{y}^j$, iff, for all $s \in X_i$ there exists $s' \in X_j$ s.t. $s(\overline{x}^i) = s'(\overline{y}^j)$.

Poly-Dependence atoms:
Let $\overline{x}^i\overline{y}^i$ and $\overline{u}^j\overline{v}^j$ be sequences of variables s.t. $|\overline{x}^i| = |\overline{u}^j|$ and $|\overline{y}^i| = |\overline{u}^j|$.

$$\mathfrak{A} \models_{\overline{X}} =\!(\overline{x}^i, \overline{y}^i / \overline{u}^j, \overline{v}^j) \Leftrightarrow \forall s \in X_i \forall s' \in X_j : s(\overline{x}^i) = s'(\overline{u}^j) \text{ implies } s(\overline{y}^i) = s'(\overline{v}^j).$$

Note that the atom $=\!(\overline{x}, \overline{y} / \overline{x}, \overline{y})$ corresponds to the dependence atom $=\!(\overline{x}, \overline{y})$.

Polyatoms

Poly-Inclusion atoms:
$\mathfrak{A} \models_{\overline{X}} \overline{x}^i \subseteq \overline{y}^j$, iff, for all $s \in X_i$ there exists $s' \in X_j$ s.t. $s(\overline{x}^i) = s'(\overline{y}^j)$.

Poly-Dependence atoms:
Let $\overline{x}^i \overline{y}^i$ and $\overline{u}^j \overline{v}^j$ be sequences of variables s.t. $|\overline{x}^i| = |\overline{u}^j|$ and $|\overline{y}^i| = |\overline{u}^j|$.

$$\mathfrak{A} \models_{\overline{X}} =\!(\overline{x}^i, \overline{y}^i / \overline{u}^j, \overline{v}^j) \Leftrightarrow \forall s \in X_i \forall s' \in X_j : s(\overline{x}^i) = s'(\overline{u}^j) \text{ implies } s(\overline{y}^i) = s'(\overline{v}^j).$$

Note that the atom $=\!(\overline{x}, \overline{y} / \overline{x}, \overline{y})$ corresponds to the dependence atom $=\!(\overline{x}, \overline{y})$.

Axioms of Poly-Dependence

Definition (Axiomatization for poly-dependence atoms)

- Reflexivity: $=(\overline{x}^i, \mathrm{pr}_k(\overline{x}^i)/\overline{y}^j, \mathrm{pr}_k(\overline{y}^j))$, where $k = 1, \ldots, |\overline{x}^i|$ and pr_k takes the kth projection of a sequence.
- Augmentation: if $=(\overline{x}^i, \overline{y}^i/\overline{u}^j, \overline{v}^j)$, then $=(\overline{x}^i\overline{z}^i, \overline{y}^i\overline{z}^i/\overline{u}^j\overline{w}^j, \overline{v}^j\overline{w}^j)$
- Transitivity: if $=(\overline{x}^i, \overline{y}^i/\overline{u}^j, \overline{v}^j)$ and $=(\overline{y}^i, \overline{z}^i/\overline{v}^j, \overline{w}^j)$, then $=(\overline{x}^i, \overline{z}^i/\overline{u}^j, \overline{w}^j)$
- Union: if $=(\overline{x}^i, \overline{y}^i/\overline{u}^j, \overline{v}^j)$ and $=(\overline{x}^i, \overline{z}^i/\overline{u}^j, \overline{w}^j)$ then $=(\overline{x}^i, \overline{y}^i\overline{z}^i/\overline{u}^j, \overline{v}^j\overline{w}^j)$
- Symmetry: if $=(\overline{x}^i, \overline{y}^i/\overline{u}^j, \overline{v}^j)$, then $=(\overline{u}^j, \overline{v}^j/\overline{x}^i, \overline{y}^i)$
- Weak Transitivity: if $=(\overline{x}^i, \overline{y}^i\overline{z}^i\overline{z}^i/\overline{u}^j, \overline{v}^j\overline{v}^j\overline{w}^j)$, then $=(\overline{x}^i, \overline{y}^i/\overline{u}^j, \overline{w}^j)$

This proof system forms a complete characterization of logical implication for poly-dependence atoms.

Poly-Independence Atom

Independence atoms:
$\mathfrak{A} \models_X \overline{y} \perp_{\overline{x}} \overline{z}$, iff, for all $s, s' \in X$: if $s(\overline{x}) = s'(\overline{x})$ then there exists $s'' \in X$ such that

- $s''(\overline{x}) = s(\overline{x})$,
- $s''(\overline{y}) = s(\overline{y})$,
- $s''(\overline{z}) = s'(\overline{z})$.

Poly-Independence Atom

Poly-Independence atoms:

Let $\overline{x}^i$, $\overline{y}^i$, $\overline{a}^j$,$\overline{b}^j$, $\overline{u}^k$, $\overline{v}^k$, and $\overline{w}^k$ be tuples of variables such that $|\overline{x}^i| = |\overline{a}^j| = |\overline{u}^k|$, $|\overline{y}^i| = |\overline{v}^k|$, $|\overline{b}^j| = |\overline{w}^k|$.

$\mathfrak{A} \models_{\overline{X}} \overline{y}^i/\overline{v}^k \perp_{\overline{x}^i,\overline{a}^j/\overline{u}^k} \overline{b}^j/\overline{w}^k$, iff, for all $s \in X_i, s' \in X_j$: if $s(\overline{x}^i) = s'(\overline{a}^j)$ then there exists $s'' \in X_k$ such that

- $s''(\overline{u}^k) = s(\overline{x}^i)$,
- $s''(\overline{v}^k) = s(\overline{y}^i)$,
- $s''(\overline{w}^k) = s'(\overline{b}^j)$.

Poly-Independence Atom

Poly-Independence atoms:

Let $\overline{x}^i$, $\overline{y}^i$, $\overline{a}^j$,$\overline{b}^j$, $\overline{u}^k$, $\overline{v}^k$, and $\overline{w}^k$ be tuples of variables such that $|\overline{x}^i| = |\overline{a}^j| = |\overline{u}^k|$, $|\overline{y}^i| = |\overline{v}^k|$, $|\overline{b}^j| = |\overline{w}^k|$.

$\mathfrak{A} \models_{\overline{X}} \overline{y}^i/\overline{v}^k \perp_{\overline{x}^i,\overline{a}^j/\overline{u}^k} \overline{b}^j/\overline{w}^k$, iff, for all $s \in X_i, s' \in X_j$: if $s(\overline{x}^i) = s'(\overline{a}^j)$ then there exists $s'' \in X_k$ such that

- $s''(\overline{u}^k) = s(\overline{x}^i)$,
- $s''(\overline{v}^k) = s(\overline{y}^i)$,
- $s''(\overline{w}^k) = s'(\overline{b}^j)$.

The atom $\overline{y}/\overline{y} \perp_{\overline{x},\overline{x}/\overline{x}} \overline{z}/\overline{z}$ is the standard independence atom $\overline{y} \perp_{\overline{x}} \overline{z}$.

Example

A relational database schema

$$\mathrm{P(\textsc{rojects})} = \{\texttt{project},\texttt{team}\}, \quad \mathrm{T(\textsc{eams})} = \{\texttt{team},\texttt{employee}\},$$
$$\mathrm{E(\textsc{mployees})} = \{\texttt{employee},\texttt{team},\texttt{project}\},$$

stores information about distribution of employees for teams and projects in a workplace. The poly-independence atom

$$\mathrm{P}[\texttt{project}]/\mathrm{E}[\texttt{project}] \perp_{\mathrm{P}[\texttt{team}],\mathrm{T}[\texttt{team}]/\mathrm{E}[\texttt{team}]} \mathrm{T}[\texttt{employee}]/\mathrm{E}[\texttt{employee}]$$

expresses that the relation $\textsc{Employees}$ includes as a subrelation the natural join of $\textsc{Projects}$ and $\textsc{Teams}$.

By using additional inclusion atoms the precise natural join can be obtained.

Desired Properties of Polyteam Semantics

- Let $\phi \in \mathrm{FO}$.
  For every team X it holds that $\mathfrak{A} \models_X \phi$ iff $\mathfrak{A} \models_s \phi$, for every $s \in X$.
- Let $\phi \in \mathrm{FO}$ whose variables are all of sort $i \in \mathbb{N}$.
  For every poly-team $\overline{X}$ it holds that $\mathfrak{A} \models_{\overline{X}} \phi$ iff $\mathfrak{A} \models_{X_i} \phi$.
- Let $\mathcal{L}$ be a team-based logic and $\phi \in \mathcal{L}$ whose variables are all of sort $i \in \mathbb{N}$.
  For every poly-team $\overline{X}$ it holds that $\mathfrak{A} \models_{\overline{X}} \phi$ iff $\mathfrak{A} \models_{X_i} \phi$.

Poly-first-order logic

Definition

The syntax of *poly first-order logic* PFO is given by the following grammar:

$$\phi ::= x = y \mid x \neq y \mid R(\vec{x}) \mid \neg R(\vec{x}) \mid (\phi \wedge \phi) \mid (\phi \vee \phi) \mid (\phi \vee^{j} \phi) \mid \exists x \phi \mid \forall x \phi,$$

where $\vec{x} \subseteq \mathrm{Var}(i)^{\mathrm{ar}(R)}$ for some $i \in \mathbb{N}$.

Poly-dependence logics. *Poly-dependence* $\mathrm{PFO(pdep)}$ is obtained by extending PFO with poly-dependence atoms.

Poly-independence, *poly-inclusion*, and *poly-exclusion logics* are obtained analogously.

Definition (Polyteam semantics for poly-first-order logic PFO)

Let $\mathfrak{A}$ be a τ-structure and $\overline{X}$ a polyteam of $\mathfrak{A}$. The satisfaction relation $\models_{\overline{X}}$ for first-order logic is defined as follows:

$$\begin{aligned}
\mathfrak{A} \models_{\overline{X}} x = y \quad &\Leftrightarrow \text{ if } x, y \in \mathrm{Var}(i) \text{ then } \forall s \in X_i : s(x) = s(y)\\
\mathfrak{A} \models_{\overline{X}} x \neq y \quad &\Leftrightarrow \text{ if } x, y \in \mathrm{Var}(i) \text{ then } \forall s \in X_i : s(x) \neq s(y)\\
\mathfrak{A} \models_{\overline{X}} R(\overline{x}) \quad &\Leftrightarrow \text{ if } \overline{x} \in \mathrm{Var}(i)^k \text{ then } \forall s \in X_i : s(\overline{x}) \in R^{\mathfrak{A}}\\
\mathfrak{A} \models_{\overline{X}} \neg R(\overline{x}) \quad &\Leftrightarrow \text{ if } \overline{x} \in \mathrm{Var}(i)^k \text{ then } \forall s \in X_i : s(\overline{x}) \notin R^{\mathfrak{A}}
\end{aligned}$$

Definition (Polyteam semantics for poly-first-order logic PFO)

Let $\mathfrak{A}$ be a τ-structure and $\overline{X}$ a polyteam of $\mathfrak{A}$. The satisfaction relation $\models_{\overline{X}}$ for first-order logic is defined as follows:

$$
\begin{aligned}
\mathfrak{A} \models_{\overline{X}} (\psi \wedge \theta) &\Leftrightarrow \mathfrak{A} \models_{\overline{X}} \psi \text{ and } \mathfrak{A} \models_{\overline{X}} \theta \\
\mathfrak{A} \models_{\overline{X}} (\psi \vee \theta) &\Leftrightarrow \mathfrak{A} \models_{\overline{Y}} \psi \text{ and } \mathfrak{A} \models_{\overline{Z}} \theta \text{ for some } \overline{Y}, \overline{Z} \subseteq \overline{X} \text{ s.t. } \overline{Y} \cup \overline{Z} = \overline{X} \\
\mathfrak{A} \models_{\overline{X}} (\psi \vee^j \theta) &\Leftrightarrow \mathfrak{A} \models_{\overline{X}[Y_j/X_j]} \psi \text{ and } \mathfrak{A} \models_{\overline{X}[Z_j/X_j]} \theta \\
&\qquad \text{for some } Y_j, Z_j \subseteq X_j \text{ s.t. } Y_j \cup Z_j = X_j
\end{aligned}
$$

Definition (Polyteam semantics for poly-first-order logic PFO)

Let $\mathfrak{A}$ be a τ-structure and $\overline{X}$ a polyteam of $\mathfrak{A}$. The satisfaction relation $\models_{\overline{X}}$ for first-order logic is defined as follows:

$$\mathfrak{A} \models_{\overline{X}} \forall x\psi \quad \Leftrightarrow \mathfrak{A} \models_{\overline{X}[X_i[A/x]/X_i]} \psi, \text{ when } x \in \mathrm{Var}(i)$$

$$\mathfrak{A} \models_{\overline{X}} \exists x\psi \quad \Leftrightarrow \mathfrak{A} \models_{\overline{X}[X_i[F/x]/X_i]} \psi \text{ holds for some } F\colon X_i \to \mathcal{P}(A) \setminus \{\emptyset\}, \text{ when } x \in \mathrm{Var}(i).$$

Definition (Polyteam semantics for poly-first-order logic PFO)

Let $\mathfrak{A}$ be a τ-structure and $\overline{X}$ a polyteam of $\mathfrak{A}$. The satisfaction relation $\models_{\overline{X}}$ for first-order logic is defined as follows:

$$
\begin{aligned}
\mathfrak{A} \models_{\overline{X}} x = y \quad &\Leftrightarrow \text{ if } x, y \in \mathrm{Var}(i) \text{ then } \forall s \in X_i : s(x) = s(y)\\
\mathfrak{A} \models_{\overline{X}} x \neq y \quad &\Leftrightarrow \text{ if } x, y \in \mathrm{Var}(i) \text{ then } \forall s \in X_i : s(x) \neq s(y)\\
\mathfrak{A} \models_{\overline{X}} R(\overline{x}) \quad &\Leftrightarrow \text{ if } \overline{x} \in \mathrm{Var}(i)^k \text{ then } \forall s \in X_i : s(\overline{x}) \in R^{\mathfrak{A}}\\
\mathfrak{A} \models_{\overline{X}} \neg R(\overline{x}) \quad &\Leftrightarrow \text{ if } \overline{x} \in \mathrm{Var}(i)^k \text{ then } \forall s \in X_i : s(\overline{x}) \notin R^{\mathfrak{A}}\\
\mathfrak{A} \models_{\overline{X}} (\psi \wedge \theta) \quad &\Leftrightarrow \mathfrak{A} \models_{\overline{X}} \psi \text{ and } \mathfrak{A} \models_{\overline{X}} \theta\\
\mathfrak{A} \models_{\overline{X}} (\psi \vee \theta) \quad &\Leftrightarrow \mathfrak{A} \models_{\overline{Y}} \psi \text{ and } \mathfrak{A} \models_{\overline{Z}} \theta \text{ for some } \overline{Y}, \overline{Z} \subseteq \overline{X} \text{ s.t. } \overline{Y} \cup \overline{Z} = \overline{X}\\
\mathfrak{A} \models_{\overline{X}} (\psi \vee^j \theta) \quad &\Leftrightarrow \mathfrak{A} \models_{\overline{X}[Y_j/X_j]} \psi \text{ and } \mathfrak{A} \models_{\overline{X}[Z_j/X_j]} \theta\\
&\qquad \text{for some } Y_j, Z_j \subseteq X_j \text{ s.t. } Y_j \cup Z_j = X_j\\
\mathfrak{A} \models_{\overline{X}} \forall x \psi \quad &\Leftrightarrow \mathfrak{A} \models_{\overline{X}[X_i[A/x]/X_i]} \psi, \text{ when } x \in \mathrm{Var}(i)\\
\mathfrak{A} \models_{\overline{X}} \exists x \psi \quad &\Leftrightarrow \mathfrak{A} \models_{\overline{X}[X_i[F/x]/X_i]} \psi \text{ holds for some } F \colon X_i \to \mathcal{P}(A) \setminus \{\emptyset\},\\
&\qquad \text{when } x \in \mathrm{Var}(i).
\end{aligned}
$$

Example

A relational database schema

$$\textsc{Patient} = \{\texttt{patient\_id}, \texttt{patient\_name}\},$$
$$\textsc{Case} = \{\texttt{case\_id}, \texttt{patient\_id}, \texttt{diagnosis\_id}, \texttt{confirmation}\},$$
$$\textsc{Test} = \{\texttt{diagnosis\_id}, \texttt{test\_id}\},$$
$$\textsc{Results} = \{\texttt{patient\_id}, \texttt{test\_id}, \texttt{result}\}$$

On $\textsc{Case}$ the foreign key $\texttt{patient\_id}$ referring to $\texttt{patient\_id}$ on $\textsc{Patient}$ (i.e. the inclusion atom $\textsc{Case}[\texttt{patient\_id}] \subseteq \textsc{Patient}[\texttt{patient\_id}]$) enforces that patient ids on $\textsc{Case}$ refer to real patients.

Example

The poly-inclusion formula

$$\phi_0 = \texttt{confirmation} \neq \textit{positive} \vee_{\textsc{Case}} \exists x_1 x_2 \Big(x_1 \neq x_2 \wedge \bigwedge_{i=1,2} \big(\textsc{Case}[\texttt{diagnosis\_id}, x_i] \subseteq \textsc{Test}[\texttt{diagnosis\_id,test\_id}] \wedge \textsc{Case}[\texttt{patient\_id}, x_i, \textit{positive}] \subseteq \textsc{Results}[\texttt{patient\_id,test\_id,result}] \big) \Big)$$

ensures that a diagnosis may be confirmed only if it has been affirmed by two different appropriate tests.

Expressive Power of uni-dependencies

Uni-atoms describe properties of single teams (e.g., dependence and independence atoms are uni-atoms while poly-dependence atoms are not).

Theorem

Let $\mathcal{C}$ be a set of uni-atoms. Each formula $\phi(\overline{x}^1, \ldots, \overline{x}^n) \in \mathrm{PFO}(\mathcal{C})$ can be associated with a sequence of formulae $\psi_1(\overline{x}^1), \ldots, \psi_n(\overline{x}^n) \in \mathrm{FO}(\mathcal{C})$ such that for all $\overline{X} = (X_1, \ldots, X_n)$, where X_i is a team with domain $\overline{x}^i$,

$$\mathcal{M} \models_{\overline{X}} \phi(\overline{x}^1, \ldots, \overline{x}^n) \Leftrightarrow \forall i = 1, \ldots, n : \mathcal{M} \models_{X_i} \psi_i(\overline{x}^i).$$

Similarly, the statement holds vice versa.

Corollary

The poly-constancy atom $=(x^1/x^2)$ cannot be expressed in $\mathrm{PFO}(\mathrm{dep})$.

Expressive Power of poly-dependencies

PFO(pdep) defines all downward closed ESO properties of polyteams.

Theorem

Let $\psi(R_1, \ldots, R_n)$ be an ESO sentence that is downward closed with respect to R_i. Then there is a PFO(pdep) formula $\phi(\overline{x}^1, \ldots, \overline{x}^n)$, where $|\overline{x}^i| = ar(R_i)$, such that for all polyteams $\overline{X} = (X_1, \ldots, X_n)$ with $Dom(X_i) = \overline{x}^i$ and $X_i \neq \emptyset$,

$$\mathcal{M} \models_{\overline{X}} \phi(\overline{x}^1, \ldots, \overline{x}^n) \Leftrightarrow (\mathcal{M}, R_1 := \mathrm{Rel}(X_1), \ldots, R_n := \mathrm{Rel}(X_n)) \models \psi(R_1, \ldots, R_n).$$

The statement holds also vice versa.

PFO(pind) defines all ESO properties of polyteams.

Example

A relational database schemas

$$\mathcal{S}: \quad \mathrm{P(\textsc{rojects})} = \{\texttt{name, employee, employee\_position}\},$$
$$\mathcal{T}: \quad \mathrm{E(\textsc{mployees})} = \{\texttt{name, project\_1, project\_2}\}$$

are used to store information about employees positions in different projects. The $\mathrm{PFO}(\mathrm{pinc}, \mathrm{dep})$-formula

$$\phi := \exists x_1 \exists x_2 \exists x_3 \Big((\mathrm{P}[\texttt{employee, name}] \subseteq \mathrm{E}[x_1, x_2] \vee_{\mathrm{P}} \mathrm{P}[\texttt{employee, name}] \subseteq \mathrm{E}[x_1, x_3]) \wedge =(x_1, (x_2, x_3)) \Big),$$

when evaluated on a polyteam that encodes an instance of the schema $\mathcal{S}$, expresses that a solution for the data exchange problem exists. The variables x_1, x_2 and x_3 above are of the sort E and are used to encode attribute names `name`, `project_1` and `project_2`, respectively. The dependence atom above enforces that the attribute `name` is a key.

Future directions

- Model important questions of database dependency theory in our setting.
- Develop axiomatisations for fragments of related logics.
- Study related complexity theoretic issues.
- Much more...

Thanks!

Future directions

- Model important questions of database dependency theory in our setting.
- Develop axiomatisations for fragments of related logics.
- Study related complexity theoretic issues.
- Much more...

Thanks!