Decidability of predicate logics with team semantics

Jonni Virtema

University of Helsinki, Finland
jonni.virtema@gmail.com

Joint work with Juha Kontinen and Antti Kuusisto

MFCS 2016
23rd of August, 2016

Core of Team Semantics

- In most studied logics formulae are evaluated in a single state of affairs. E.g.,
  - a first-order assignment in first-order logic,
  - a propositional assignment in propositional logic,
  - a possible world of a Kripke structure in modal logic.
- In team semantics sets of states of affairs are considered. E.g.,
  - a set of first-order assignments in first-order logic,
  - a set of propositional assignments in propositional logic,
  - a set of possible worlds of a Kripke structure in modal logic.
- These sets of things are called teams.

Core of Team Semantics

- In most studied logics formulae are evaluated in a single state of affairs. E.g.,
  - a first-order assignment in first-order logic,
  - a propositional assignment in propositional logic,
  - a possible world of a Kripke structure in modal logic.
- In team semantics sets of states of affairs are considered. E.g.,
  - a set of first-order assignments in first-order logic,
  - a set of propositional assignments in propositional logic,
  - a set of possible worlds of a Kripke structure in modal logic.
- These sets of things are called teams.

Core of Team Semantics

- In most studied logics formulae are evaluated in a single state of affairs. E.g.,
  - a first-order assignment in first-order logic,
  - a propositional assignment in propositional logic,
  - a possible world of a Kripke structure in modal logic.
- In team semantics sets of states of affairs are considered. E.g.,
  - a set of first-order assignments in first-order logic,
  - a set of propositional assignments in propositional logic,
  - a set of possible worlds of a Kripke structure in modal logic.
- These sets of things are called teams.

Team Semantics: Motivation and History

Logical modelling of uncertainty, imperfect information, and different notions of dependence such as functional dependence and independence. Related to similar concepts in statistics, database theory 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 by Väänänen 2007.
- Introduction of other dependency notions to team semantics such as inclusion, exclusion, and independence. Galliani, Grädel, Väänänen.
- Generalized atoms by Kuusisto (derived from generalised quantifiers).

Team Semantics: Motivation and History

Logical modelling of uncertainty, imperfect information, and different notions of dependence such as functional dependence and independence. Related to similar concepts in statistics, database theory 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 by Väänänen 2007.
- Introduction of other dependency notions to team semantics such as inclusion, exclusion, and independence. Galliani, Grädel, Väänänen.
- Generalized atoms by Kuusisto (derived from generalised quantifiers).

First-order logic

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

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

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:

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

First-order logic

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

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

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.

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

Team semantics for first-order logic

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

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

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.

$$\mathfrak{A} \models_X R(\vec{x}) \quad \Leftrightarrow \quad \forall s \in X : s(\vec{x}) \in R^{\mathfrak{A}}$$

$$\mathfrak{A} \models_X R(\vec{x}) \quad \Leftrightarrow \quad \forall s \in X : s(\vec{x}) \notin R^{\mathfrak{A}}$$

$$\mathfrak{A} \models_X \varphi \wedge \psi \quad \Leftrightarrow \quad \mathfrak{A} \models_X \varphi \text{ and } \mathfrak{A} \models_X \psi$$

$$\mathfrak{A} \models_X \varphi \vee \psi \quad \Leftrightarrow \quad \mathfrak{A} \models_Y \varphi \text{ and } \mathfrak{A} \models_Z \psi \text{ for some } Y \cup Z = X$$

$$\mathfrak{A}, s \models \forall x \varphi \quad \Leftrightarrow \quad \mathfrak{A}, s(a/x) \models \varphi \text{ for all } a \in A$$

$$\mathfrak{A}, s \models \exists x \varphi \quad \Leftrightarrow \quad \mathfrak{A}, s(a/x) \models \varphi \text{ for some } a \in A$$

Team semantics for first-order logic

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

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

Team semantics for first-order logic

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

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

Team semantics for first-order logic

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

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

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

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

Dependence logic

Dependence logic is the extension of first-order logic with dependence atoms.

The intuitive meaning of the dependence atom $=(\vec{x}, y)$ is that inside a team the value of the variable y depends solely on the values of the variables in $\vec{x}$.

The semantics for dependence atoms is defined as follows:

$$\mathfrak{A} \models_X =(\vec{x}, y) \text{ iff } \forall s, s' \in X : \text{ if } s(\vec{x}) = s'(\vec{x}) \text{ then } s(y) = s'(y).$$

Properties of dependence logic

- Downwards closure: If $\mathfrak{A} \models_X \varphi$ and $Y \subseteq X$ then $\mathfrak{A} \models_Y \varphi$
- Locality: $\mathfrak{A} \models_X \varphi$ if and only if $\mathfrak{A} \models_{X \upharpoonright \mathrm{Fr}(\varphi)} \varphi$
- Expressive power on sentences: Existential second order logic

Two-variable dependence logic

- The formula
$$\forall x \bigvee_{1\leq i\leq k} =\!(x)$$
defines the class of models of cardinality $\leq k$.
- The formula
$$\forall x \exists y\Big(=\!(y,x) \wedge \exists x\big(=\!(x) \wedge \neg x = y\big)\Big)$$
defines the class of infinite models.
- Satifiability and finite satisfiability problems are NEXPTIME-complete (Kontinen, Kuusisto, Lohmann, V. 2011). Proof follows from translations: $\mathcal{FO}^2 \mapsto \mathcal{D}^2 \mapsto \Sigma^1_1(\mathcal{FOC}^2)$.
- Data complexity for $\mathcal{D}^2$ is NP-complete (V. 2014). (Dominating set problem.)

Complexity of validity for $\mathcal{D}^2$

We show that the validity problem for $\mathcal{D}^2$ is undecidable.

We give a reduction from non-tiling problem.

Non-tiling problem

The grid is the structure $\mathcal{G} = (\mathbb{N}^2, V, H)$, where
$V = \{((i,j),(i,j+1)) \in \mathbb{N}^2 \times \mathbb{N}^2 \mid i,j \in \mathbb{N}\}$ and
$H = \{((i,j),(i+1,j)) \in \mathbb{N}^2 \times \mathbb{N}^2 \mid i,j \in \mathbb{N}\}$.

Tiling problem:

- Input: A set of tile types T, i.e., squares with coloured sides.
- Output: Can the grid be tiled with the tile types in T?

Non-tiling problem is the complement of the tiling problem.

Reducing from tiling: Does there exist a model that is a grid and that has a valid T-tiling?

Reducing from non-tiling: Does every model that has a valid T-tiling violate the grid conditions?

Non-tiling problem

The grid is the structure $\mathcal{G} = (\mathbb{N}^2, V, H)$, where
$V = \{((i,j),(i,j+1)) \in \mathbb{N}^2 \times \mathbb{N}^2 \mid i,j \in \mathbb{N}\}$ and
$H = \{((i,j),(i+1,j)) \in \mathbb{N}^2 \times \mathbb{N}^2 \mid i,j \in \mathbb{N}\}$.

Tiling problem:

- Input: A set of tile types T, i.e., squares with coloured sides.
- Output: Can the grid be tiled with the tile types in T?

Non-tiling problem is the complement of the tiling problem.

Reducing from tiling: Does there exist a model that is a grid and that has a valid T-tiling?

Reducing from non-tiling: Does every model that has a valid T-tiling violate the grid conditions?

Grid conditions

Definition

Let $\mathfrak{A} = (A, V, H)$ be a structure with two binary relation symbols. We say that $\mathfrak{A}$ is *gridlike* if the conditions below hold.

1. The extension of V in $\mathfrak{A}$ is serial (i.e., $\forall x \in A\, \exists y \in A$ s.t. $V(x, y)$).
2. The extension of H in $\mathfrak{A}$ is serial (i.e., $\forall x \in A\, \exists y \in A$ s.t. $H(x, y)$).
3. If $a, b, c, b', c' \in A$ are such that $V(a, b)$, $H(b, c)$, $H(a, b')$, and $V(b', c')$, then $c = c'$.

We say that a $\{U, P, Q, C, V, H\}$-structure $\mathfrak{A}$ is *striped and gridlike* if the $\{V, H\}$-reduct of $\mathfrak{A}$ is gridlike, the extensions of P and Q in $\mathfrak{A}$ are *distinct* singleton sets, the extension of U in $\mathfrak{A}$ is the union of the extensions of P and Q, and $\mathfrak{A}$ has the following property (intuitively C creates stripes in $\mathfrak{A}$):

$$\big(H(a, b) \Rightarrow (C(a) \Leftrightarrow C(b))\big) \text{ and } \big(V(a, b) \Rightarrow (C(a) \Leftrightarrow \neg C(b))\big).$$

From grid to gridlike structures

Lemma

If $\mathfrak{A}$ is striped and gridlike, then there exists a homomorphism from the grid into $\mathfrak{A}$.

Lemma

Let T be an input to the non-tiling problem. The grid is non-T-tilable iff (the $\{H, V\}$-reduct of) every striped gridlike structure is non-T-tilable.

Thus it suffices to show that the following can be expressed in $\mathcal{D}^2$:

The structure is not striped and gridlike or it is not correctly T-tiled.

Failure of the grid condition 3

Express the following: There exists distinct points c, c' s.t $V(b,c)$, $H(b',c')$, $H(a,b)$, and $V(a,b')$, for some b, b', a.

Essentially the following formula:

$$\forall x\Big(\neg U(x) \vee \exists y\Big(C(y) \wedge =\!(y,x)$$
$$\wedge\, \exists x\Big(=\!(x,y) \wedge \big((=\!(x) \wedge H(x,y)) \vee (=\!(x) \wedge V(x,y))\big)$$
$$\wedge\, \exists y\big(=\!(y) \wedge (V(y,x) \vee H(y,x)) \wedge \neg C(y)\big)\Big)\Big)\Big).$$

Failure of the grid condition 3

Express the following: There exists distinct points c, c' s.t $V(b, c)$, $H(b', c')$, $H(a, b)$, and $V(a, b')$, for some b, b', a.

Essentially the following formula:

$$\forall x\Big(\neg U(x) \vee \exists y\Big(C(y) \wedge =(y,x) \\ \wedge \exists x\Big(=(x,y) \wedge ((=(x) \wedge H(x,y)) \vee (=(x) \wedge V(x,y))) \\ \wedge \exists y(=(y) \wedge (V(y,x) \vee H(y,x)) \wedge \neg C(y)))\Big)\Big)\Big).$$

THANKS!