Complexity of propositional logics in team semantics

Jonni Virtema

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

Joint work with Miika Hannula[1], Martin Lück[2], Juha Kontinen[3], and Heribert Vollmer[2]
Related to papers in GandALF 2016, MFCS 2015, Information and Computation 2016

[1] The University of Auckland, [2] University of Hanover, [3] University of Helsinki,

8th of December, 2016 – Computational Logic Day 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:

- 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.
- Modal dependence logic by Väänänen 2008.
- 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:

- 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.
- Modal dependence logic by Väänänen 2008.
- 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).

Propositional logic

Syntax of propositional logic:

$$\varphi ::= p \mid \neg p \mid (\varphi \land \varphi) \mid (\varphi \lor \varphi)$$

Semantics via propositional assignments:

"name"	p	q	r
s	0	1	1

$$s \models (q \land r)$$

Team semantics / semantics via sets of assignments:

"name"	p	q	r
s	0	1	1
t	1	1	0
u	0	1	0

$$\{s, t, u\} \models q, \quad \{s, t\} \models (p \lor r)$$

Team semantics

We want that for each formula φ of propositional logic and for each team X

$$X \models \varphi \quad \text{iff} \quad \forall s \in X : s \models \varphi.$$

We define that

$$\begin{aligned}
X \models p \quad &\text{iff} \quad \forall s \in X : s(p) = 1 \\
X \models \neg p \quad &\text{iff} \quad \forall s \in X : s(p) = 0 \\
X \models \varphi \wedge \psi \quad &\text{iff} \quad X \models \varphi \text{ and } X \models \psi \\
X \models \varphi \vee \psi \quad &\text{iff} \quad Y \models \varphi \text{ and } Z \models \psi, \\
&\quad\ \text{for some } Y, Z \subseteq X \text{ such that } Y \cup Z = X.
\end{aligned}$$

Team semantics

We want that for each formula φ of propositional logic and for each team X

$$X \models \varphi \quad \text{iff} \quad \forall s \in X : s \models \varphi.$$

We define that

$$\begin{aligned} X \models p \quad & \text{iff} \quad \forall s \in X : s(p) = 1 \\ X \models \neg p \quad & \text{iff} \quad \forall s \in X : s(p) = 0 \\ X \models \varphi \wedge \psi \quad & \text{iff} \quad X \models \varphi \text{ and } X \models \psi \\ X \models \varphi \vee \psi \quad & \text{iff} \quad Y \models \varphi \text{ and } Z \models \psi, \\ & \quad\quad \text{for some } Y, Z \subseteq X \text{ such that } Y \cup Z = X. \end{aligned}$$

Extensions of propositional logic

We extend PL by adding atomic formulae that describe properties of teams.

Dependence atoms: $\mathrm{dep}(p, q, r)$
"the truth value of r is functionally determined by the truth values of p and q".

	p	q	r
s	0	1	1
t	1	1	0
u	0	1	0

$$\{s, u\} \not\models \mathrm{dep}(p, r)\,, \quad \{s, t\} \models \mathrm{dep}(p, q)\,,$$
$$\{s, t, u\} \models \mathrm{dep}(q)\,, \quad \{s, t, u\} \models \mathrm{dep}(r) \vee \mathrm{dep}(r)\,.$$

Extensions of propositional logic

We extend PL by adding atomic formulae that describe properties of teams.

Inclusion atoms: $(p_1, p_2) \subseteq (q_1, q_2)$
"truth values that appear for p_1, p_2 also appear as truth values for q_1, q_2".

	p	q	r
s	0	1	1
t	1	1	0
u	0	1	0

$$\{s, t\} \not\models p \subseteq q, \quad \{s, t\} \models q \subseteq r, \quad \{s, t, u\} \models (p, q) \subseteq (r, q)$$

Extensions of propositional logic

We extend PL by adding atomic formulae that describe properties of teams.

Syntax of propositional dependence logic PD:

$$\varphi ::= p \mid \neg p \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid \mathrm{dep}(p_1, \dots, p_n, q)$$

Syntax of propositional inclusion logic PLInc:

$$\varphi ::= p \mid \neg p \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid (p_1, \dots, p_n) \subseteq (q_1, \dots, q_n)$$

Syntax of propositional team logic PTL:

$$\varphi ::= p \mid \neg p \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid {\sim}\varphi,$$

with the semantics $X \models {\sim}\varphi$ iff $X \not\models \varphi$.

Important decision problems

Model checking:
Input: A team X and a formula φ.
Output: Does $X \models \varphi$ hold?

Satisfiability:
Input: A formula φ.
Output: Does there exists a non-empty team X s.t. $X \models \varphi$?

Validity:
Input: A formula φ.
Output: Does $X \models \varphi$ hold for every non-empty team X?

Complexity results

	Satisfiability	Validity	Model checking
PL	NP	coNP	NC^1
PD	NP	NEXPTIME	NP
PLInc	EXPTIME	coNP	P
PTL	AEXPTIME(poly)	AEXPTIME(poly)	PSPACE

Source of hardness:

A well-known NP-complete problem:
3SAT:
Input: A 3CNF-formula φ
(e.g.,$(p_2 \vee \neg p_7) \wedge (\neg p_1 \vee p_3 \vee p_2) \wedge (p_3 \vee \neg p_4 \vee \neg p_2) \wedge p_2$).
Output: Does there exists an assignment s s.t. $s \models \varphi$?

Source of hardness:

A well-known NP-complete problem:
3SAT:
Input: A 3CNF-formula φ
(e.g.,$(p_2 \vee \neg p_7) \wedge (\neg p_1 \vee p_3 \vee p_2) \wedge (p_3 \vee \neg p_4 \vee \neg p_2) \wedge p_2$).
Output: Does there exists an assignment s s.t. $s \models \varphi$?

We may rewrite the above as follows:
Input: A existentially prenex quantified QPL-sentence φ
(e.g.,$\exists p_1 \dots \exists p_7 \big((p_2 \vee \neg p_7) \wedge (\neg p_1 \vee p_3 \vee p_2) \wedge (p_3 \vee \neg p_4 \vee \neg p_2) \wedge p_2\big)$).
Output: Does $\emptyset \models \varphi$ hold?

Source of hardness:

A well-known NP-complete problem:
EQBF:
Input: A sentence φ of the form $\exists p_1 \ldots \exists p_n \psi$, where $\psi \in \mathrm{PL}$.
Output: Does $\emptyset \models \varphi$ hold?

Source of hardness:

A well-known NP-complete problem:
EQBF:
Input: A sentence φ of the form $\exists p_1 \ldots \exists p_n \psi$, where $\psi \in \mathrm{PL}$.
Output: Does $\emptyset \models \varphi$ hold?

A well-known PSPACE-complete problem:
QBF:
Input: A sentence φ of the form $\exists p_1 \forall p_2 \ldots \forall p_{n-1} \exists p_n \psi$, where $\psi \in \mathrm{PL}$.
Output: Does $\emptyset \models \varphi$ hold?

From QBF to DQBF

A well-known PSPACE-complete problem:
QBF:
Input: A prenex quantified QPL-sentence φ (e.g., $\exists p_1 \forall p_2 \forall p_3 \exists p_4 \psi$).
Output: Does $\emptyset \models \varphi$ hold?

The formula $\exists p_1 \forall p_2 \forall p_3 \exists p_4 \psi$ may be equivalently written with the help of Skolem functions $f_1 \in \{0,1\}$ and $f_2 : \{0,1\}^2 \to \{0,1\}$:

$$\exists f_1 \exists f_2 \forall p_2 \forall p_3 \, \psi\big(f_1/p_1, f_2(p_2, p_3)/p_4\big)$$

From QBF to DQBF

A well-known PSPACE-complete problem:
QBF:
Input: A prenex quantified QPL-sentence φ (e.g., $\exists p_1 \forall p_2 \forall p_3 \exists p_4 \psi$).
Output: Does $\emptyset \models \varphi$ hold?

The formula $\exists p_1 \forall p_2 \forall p_3 \exists p_4 \psi$ may be equivalently written with the help of Skolem functions $f_1 \in \{0,1\}$ and $f_2 : \{0,1\}^2 \rightarrow \{0,1\}$:

$$\exists f_1 \exists f_2 \forall p_2 \forall p_3 \, \psi\big(f_1/p_1, f_2(p_2,p_3)/p_4\big)$$

Formulae φ of the form $\exists f_1 \ldots \exists f_n \forall p_1 \ldots \forall p_k \psi$, where $\psi \in \mathrm{PL}$ and $\arg(f_i) \subseteq \{p_1, \ldots, p_n\}$, are called as DQBF-sentences. Moreover, if $\arg(f_i) \subseteq \arg(f_{i+1})$ for all i, we say that φ is simple.

From QBF to DQBF

A well-known PSPACE-complete problem:
QBF:
Input: A prenex quantified QPL-sentence φ (e.g., $\exists p_1 \forall p_2 \forall p_3 \exists p_4 \psi$).
Output: Does $\emptyset \models \varphi$ hold?

The above PSPACE-complete problem can be reformulated as follows:
SDQBF:
Input: A simple DQBF-sentence φ.
Output: Does $\emptyset \models \varphi$ hold?

From QBF to DQBF

A well-known PSPACE-complete problem:
QBF:
Input: A prenex quantified QPL-sentence φ (e.g., $\exists p_1 \forall p_2 \forall p_3 \exists p_4 \psi$).
Output: Does $\emptyset \models \varphi$ hold?

The above PSPACE-complete problem can be reformulated as follows:
SDQBF:
Input: A simple DQBF-sentence φ.
Output: Does $\emptyset \models \varphi$ hold?

Not so well-known NEXPTIME-complete problem:
DQBF: (Peterson, Reif, and Azhar 2001)
Input: A DQBF-sentence φ.
Output: Does $\emptyset \models \varphi$ hold?

From DQBF to ADQBF

Example: DQBF

Essentially an instance of DQBF is as follows:

$$\exists f_1 \ldots \exists f_n \forall p_1 \ldots \forall p_k \varphi(p_1, \ldots, p_n, f_1(\vec{c}_1), \ldots, f_n(\vec{c}_n)),$$

where φ is a propositional formula and $\vec{c}_i$ is some tuple of variables from $p_1, \ldots, p_k$.

From DQBF to ADQBF

Example: DQBF

Essentially an instance of DQBF is as follows:

$$\exists f_1 \dots \exists f_n \forall p_1 \dots \forall p_k \varphi(p_1, \dots, p_n, f_1(\vec{c}_1), \dots, f_n(\vec{c}_n)),$$

where φ is a propositional formula and $\vec{c}_i$ is some tuple of variables from $p_1, \dots, p_k$.

Definition

A Σ_k-alternating qBf, Σ_k-ADQBF is a formula of the form

$$(\exists f_1^1 \dots \exists f_{j_1}^1)(\forall f_1^2 \dots \forall f_{j_2}^2) \dots (\exists f_{j_1}^k \dots \exists f_{j_k}^k) \forall p_1 \dots \forall p_n \varphi(p_1, \dots, f_j^i(\vec{c}_j^i), \dots),$$

where φ is a propositional formula and $\vec{c}_j^i$ is some tuple of variables from $p_1, \dots, p_n$.

From DQBF to ADQBF

Definition

A Σ_k-alternating qBf, Σ_k-ADQBF is a formula of the form

$$(\exists f_1^1 \ldots \exists f_{j_1}^1)(\forall f_1^2 \ldots \forall f_{j_2}^2) \ldots (\exists f_{j_1}^k \ldots \exists f_{j_k}^k)\forall p_1 \ldots \forall p_n \varphi(p_1, \ldots, f_j^i(\vec{c}_j^i), \ldots),$$

where φ is a propositional formula and $\vec{c}_j^i$ is some tuple of variables from $p_1, \ldots, p_n$.

- Σ_k-ADQBF is Σ_k^{EXP}-complete odd k, and Σ_{k-1}^{EXP}-complete for even k.
- Π_k-ADQBF is Π_k^{EXP}-complete even k, and Π_{k-1}^{EXP}-complete for odd k.
- ADQBF is AEXPTIME(poly)-complete.

Connection between ADQBF and PTL

A Σ_k-ADQBF is a sentence

$$(\exists f_1^1 \dots \exists f_{j_1}^1)(\forall f_1^2 \dots \forall f_{j_2}^2) \dots (\exists f_{j_1}^k \dots \exists f_{j_k}^k) \forall p_1 \dots \forall p_n \varphi(p_1, \dots, f_j^i(\vec{c}_j^i), \dots)$$

can be written as the following $QPL[\sim, \mathrm{dep}(\cdot)]$-sentence

Connection between ADQBF and PTL

A Σ_k-ADQBF is a sentence

$$(\exists f_1^1 \dots \exists f_{j_1}^1)(\forall f_1^2 \dots \forall f_{j_2}^2) \dots (\exists f_{j_1}^k \dots \exists f_{j_k}^k) \forall p_1 \dots \forall p_n \varphi(p_1, \dots, f_j^i(\vec{c}_j^i), \dots)$$

can be written as the following $QPL[\sim, \mathrm{dep}(\cdot)]$-sentence

$$\forall p_1 \cdots \forall p_n \, (\exists q_1^1 \cdots \exists q_{j_1}^1) \, (U q_1^2 \cdots U q_{j_2}^2) \, (\exists q_1^3 \cdots \exists q_{j_3}^3) \dots (\, \exists q_1^k \cdots \exists q_{j_k}^k)$$
$$\sim \Bigg[\sim(p \wedge \neg p) \wedge \bigwedge_{\substack{1 \le i \le k \\ i \text{ is even} \\ 1 \le l \le j_i}} \mathrm{dep}(\overline{c}_l^i, q_l^i) \Bigg] \vee \Bigg[\Big(\bigwedge_{\substack{1 \le i \le k \\ i \text{ is odd} \\ 1 \le l \le j_i}} \mathrm{dep}(\overline{c}_l^i, q_l^i) \Big) \wedge \theta \Bigg]$$

Connection between ADQBF and PTL

$$\forall p_1 \cdots \forall p_n \, (\exists q_1^1 \cdots \exists q_{j_1}^1) \, (U q_1^2 \cdots U q_{j_2}^2) \, (\exists q_1^3 \cdots \exists q_{j_3}^3) \ldots (\exists q_1^k \cdots \exists q_{j_k}^k)$$

$$\sim \Big[\sim(p \wedge \neg p) \wedge \bigwedge_{\substack{1 \le i \le k \\ i \text{ is even} \\ 1 \le l \le j_i}} \mathrm{dep}(\overline{c}_l^i, q_l^i) \Big] \vee \Big[\Big(\bigwedge_{\substack{1 \le i \le k \\ i \text{ is odd} \\ 1 \le l \le j_i}} \mathrm{dep}(\overline{c}_l^i, q_l^i) \Big) \wedge \theta \Big]$$

Dependence atoms can be eliminated from above by the use of $\sim$.

The quantifiers can be eliminated by a shift to satisfiability and by simulating existential quantifiers by $\vee$ and universal quantifiers by $\sim \vee \sim$.

THANKS!

References

Jonni Virtema, Complexity of validity for propositional dependence logics, *Information and Computation*, Elsevier, 2016.

Miika Hannula, Martin Lück, Juha Kontinen, and Jonni Virtema, On quantified propositional logics and the exponential time hierarchy, proceedings of *the 7th International Symposium on Games, Automata, Logics and Formal Verification, GandALF 2016*.

Miika Hannula, Juha Kontinen, Jonni Virtema, and Heribert Vollmer, Complexity of Propositional Independence and Inclusion Logic, proceedings of *the 40th International Symposium on Mathematical Foundations of Computer Science, MFCS 2015*.