On quantified propositional logics and the exponential time hierarchy

Jonni Virtema

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

Joint work with Miika Hannula, Juha Kontinen, and Martin Lück

GandALF 2016
15th of September, 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).

Quantified propositional logic

Grammar of quantified propositional logic $\mathcal{QPL}$ (or QBF) in negation normal form:

$$\varphi ::= p \mid \neg p \mid (\varphi \vee \varphi) \mid (\varphi \wedge \varphi) \mid \exists p\varphi \mid \forall p\varphi.$$

A propositional team is a set of assigments $s : \mathrm{PROP} \to \{0,1\}$ with the same domain.

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

If φ is an $\mathcal{QPL}$-formula and X a set of propositional assignments:

$$X \models \varphi \iff \forall s \in X : s \models \varphi.$$

Quantified propositional logic

Grammar of quantified propositional logic $\mathcal{QPL}$ (or QBF) in negation normal form:

$$\varphi ::= p \mid \neg p \mid (\varphi \vee \varphi) \mid (\varphi \wedge \varphi) \mid \exists p \varphi \mid \forall p \varphi.$$

A propositional team is a set of assigments $s : \mathrm{PROP} \to \{0, 1\}$ with the same domain.

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

If φ is an $\mathcal{QPL}$-formula and X a set of propositional assignments:

$$X \models \varphi \iff \forall s \in X : s \models \varphi.$$

Team Semantics for Propositional Logics

A propositional team is a set of assigments $s : \mathrm{PROP} \to \{0,1\}$ with the same domain.

$$
\begin{aligned}
s \models p \quad &\Leftrightarrow \quad s(p) = 1\\
s \models \neg p \quad &\Leftrightarrow \quad s(p) = 0\\
s \models \varphi \wedge \psi \quad &\Leftrightarrow \quad s \models \varphi \text{ and } s \models \psi\\
s \models \varphi \vee \psi \quad &\Leftrightarrow \quad s \models \varphi \text{ or } s \models \psi\\
s \models \exists p\, \varphi \quad &\Leftrightarrow \quad s(b/p) \models \varphi \text{ for some } b \in \{0,1\}\\
s \models \forall p\, \varphi \quad &\Leftrightarrow \quad s(b/p) \models \varphi \text{ for all } b \in \{0,1\}
\end{aligned}
$$

Team Semantics for Propositional Logics

A propositional team is a set of assigments $s : \mathrm{PROP} \to \{0,1\}$ with the same domain.

$$
\begin{aligned}
X \models p \quad &\Leftrightarrow \quad \forall s \in X : s(p) = 1\\
s \models \neg p \quad &\Leftrightarrow \quad s(p) = 0\\
s \models \varphi \wedge \psi \quad &\Leftrightarrow \quad s \models \varphi \text{ and } s \models \psi\\
s \models \varphi \vee \psi \quad &\Leftrightarrow \quad s \models \varphi \text{ or } s \models \psi\\
s \models \exists p\, \varphi \quad &\Leftrightarrow \quad s(b/p) \models \varphi \text{ for some } b \in \{0,1\}\\
s \models \forall p\, \varphi \quad &\Leftrightarrow \quad s(b/p) \models \varphi \text{ for all } b \in \{0,1\}
\end{aligned}
$$

Team Semantics for Propositional Logics

A propositional team is a set of assigments $s : \mathrm{PROP} \to \{0, 1\}$ with the same domain.

$$
\begin{aligned}
X \models p &\quad\Leftrightarrow\quad \forall s \in X : s(p) = 1 \\
X \models \neg p &\quad\Leftrightarrow\quad \forall s \in X : s(p) = 0 \\
s \models \varphi \wedge \psi &\quad\Leftrightarrow\quad s \models \varphi \text{ and } s \models \psi \\
s \models \varphi \vee \psi &\quad\Leftrightarrow\quad s \models \varphi \text{ or } s \models \psi \\
s \models \exists p\, \varphi &\quad\Leftrightarrow\quad s(b/p) \models \varphi \text{ for some } b \in \{0, 1\} \\
s \models \forall p\, \varphi &\quad\Leftrightarrow\quad s(b/p) \models \varphi \text{ for all } b \in \{0, 1\}
\end{aligned}
$$

Team Semantics for Propositional Logics

A propositional team is a set of assigments $s : \mathrm{PROP} \to \{0,1\}$ with the same domain.

$$
\begin{aligned}
X \models p &\quad\Leftrightarrow\quad \forall s \in X : s(p) = 1\\
X \models \neg p &\quad\Leftrightarrow\quad \forall s \in X : s(p) = 0\\
X \models \varphi \wedge \psi &\quad\Leftrightarrow\quad X \models \varphi \text{ and } X \models \psi\\
s \models \varphi \vee \psi &\quad\Leftrightarrow\quad s \models \varphi \text{ or } s \models \psi\\
s \models \exists p\, \varphi &\quad\Leftrightarrow\quad s(b/p) \models \varphi \text{ for some } b \in \{0,1\}\\
s \models \forall p\, \varphi &\quad\Leftrightarrow\quad s(b/p) \models \varphi \text{ for all } b \in \{0,1\}
\end{aligned}
$$

Team Semantics for Propositional Logics

A propositional team is a set of assigments $s : \mathrm{PROP} \to \{0,1\}$ with the same domain.

$$
\begin{aligned}
X \models p &\quad\Leftrightarrow\quad \forall s \in X : s(p) = 1\\
X \models \neg p &\quad\Leftrightarrow\quad \forall s \in X : s(p) = 0\\
X \models \varphi \wedge \psi &\quad\Leftrightarrow\quad X \models \varphi \text{ and } X \models \psi\\
X \models \varphi \vee \psi &\quad\Leftrightarrow\quad Y \models \varphi \text{ and } Z \models \psi \text{ for some } Y \cup Z = X\\
s \models \exists p\, \varphi &\quad\Leftrightarrow\quad s(b/p) \models \varphi \text{ for some } b \in \{0,1\}\\
s \models \forall p\, \varphi &\quad\Leftrightarrow\quad s(b/p) \models \varphi \text{ for all } b \in \{0,1\}
\end{aligned}
$$

Team Semantics for Propositional Logics

A propositional team is a set of assigments $s : \mathrm{PROP} \to \{0,1\}$ with the same domain.

$$
\begin{aligned}
X \models p \quad &\Leftrightarrow \quad \forall s \in X : s(p) = 1 \\
X \models \neg p \quad &\Leftrightarrow \quad \forall s \in X : s(p) = 0 \\
X \models \varphi \wedge \psi \quad &\Leftrightarrow \quad X \models \varphi \text{ and } X \models \psi \\
X \models \varphi \vee \psi \quad &\Leftrightarrow \quad Y \models \varphi \text{ and } Z \models \psi \text{ for some } Y \cup Z = X \\
X \models \exists p\, \varphi \quad &\Leftrightarrow \quad Y \models \varphi \text{ for some } Y \text{ s.t } Y \upharpoonright \mathrm{dom}(X) = X \\
s \models \forall p\, \varphi \quad &\Leftrightarrow \quad s(b/p) \models \varphi \text{ for all } b \in \{0,1\}
\end{aligned}
$$

Team Semantics for Propositional Logics

A propositional team is a set of assigments $s : \mathrm{PROP} \to \{0,1\}$ with the same domain.

$$
\begin{aligned}
X \models p \quad &\Leftrightarrow \quad \forall s \in X : s(p) = 1\\
X \models \neg p \quad &\Leftrightarrow \quad \forall s \in X : s(p) = 0\\
X \models \varphi \wedge \psi \quad &\Leftrightarrow \quad X \models \varphi \text{ and } X \models \psi\\
X \models \varphi \vee \psi \quad &\Leftrightarrow \quad Y \models \varphi \text{ and } Z \models \psi \text{ for some } Y \cup Z = X\\
X \models \exists p\, \varphi \quad &\Leftrightarrow \quad Y \models \varphi \text{ for some } Y \text{ s.t } Y \upharpoonright \mathrm{dom}(X) = X\\
X \models \forall p\, \varphi \quad &\Leftrightarrow \quad Y \models \varphi \text{ for the maximal } Y \text{ s.t } Y \upharpoonright \mathrm{dom}(X) = X
\end{aligned}
$$

Note that $\emptyset \models \varphi$ for every $\mathcal{QPL}$-formula φ, and additionally:

$$X \models \varphi \quad \Leftrightarrow \quad \forall s \in X : s \models \varphi.$$

Propositional Dependence, Inclusion and Independence Logic

Grammar of propositional logic $\mathcal{PL}$:

$$\varphi ::= p \mid \neg p \mid (\varphi \vee \varphi) \mid (\varphi \wedge \varphi).$$

Extensions $\mathcal{PL}$ and $\mathcal{QPL}$ by inclusion atoms, independence atoms, and classical negation.

$$\varphi ::= p_1, \dots, p_n \subseteq q_1, \dots, q_n \mid \vec{r} \perp_{\vec{p}} \vec{q} \mid {\sim}\varphi.$$

The logics are denoted by $\mathcal{PL}[\perp_{\mathrm{c}}, \sim]$, $\mathcal{PL}[\subseteq, \sim]$, $\mathcal{QPL}[\sim]$ etc.

Team Sematics for the Extensions

Atoms $\subseteq$ and $\perp_c$ can be expressed in $\mathcal{PL}[\sim]$ with exponential blow up. The atom $\mathrm{dep}(\cdot)$ requires just polynomial blow up to be expresssed in $\mathcal{PL}[\sim]$.

$$
\begin{aligned}
X \models \mathrm{dep}(\vec{p}, q) \quad &\Leftrightarrow \quad \forall s, t \in X : s(\vec{p}) = t(\vec{p}) \Rightarrow s(q) = t(q) \\
X \models \vec{p} \subseteq \vec{q} \quad &\Leftrightarrow \quad \forall s \in X \exists t \in X : s(\vec{p}) = t(\vec{q}) \\
X \models \vec{q} \perp_{\vec{p}} \vec{r} \quad &\Leftrightarrow \quad \forall s, t \in X : \text{ if } s(\vec{p}) = t(\vec{p}) \\
&\quad\quad \text{then there exists } u \in X : u(\vec{p}\vec{q}) = s(\vec{p}\vec{q}) \text{ and } u(\vec{r}) = t(\vec{r}) \\
X \models {\sim}\varphi \quad &\Leftrightarrow \quad X \not\models \varphi
\end{aligned}
$$

Already $\mathcal{PL}[\sim]$ is Highly Expressive!

Most connectives studied in team sematics can be defined in $\mathcal{PL}[\sim]$.

The connectives below can be defined in $\mathcal{PL}[\sim]$ with polynomial blow up.

$$
\begin{aligned}
X \models \varphi \owedge \psi \quad &\Leftrightarrow \quad X \models \varphi \text{ or } X \models \psi, \\
X \models \varphi \otimes \psi \quad &\Leftrightarrow \quad \forall Y, Z \subseteq X: \text{ if } Y \cup Z = X, \text{ then } Y \models \varphi \text{ or } Z \models \psi, \\
X \models \varphi \to \psi \quad &\Leftrightarrow \quad \forall Y \subseteq X: \text{if } Y \models \varphi, \text{ then } Y \models \psi, \\
X \models \max(p_1, \dots, p_n) \quad &\Leftrightarrow \quad \{(s(p_1), \dots, s(p_n)) \mid s \in X\} = \{0,1\}^n.
\end{aligned}
$$

In $\mathcal{QPL}[\sim]$ the dual $\sim\exists\sim$ is denoted by U and it has the following semantics:

$$X \models U p \varphi \quad \Leftrightarrow \quad Y \models \varphi \text{ for all } Y \text{ s.t } Y \upharpoonright \mathrm{dom}(X) = X$$

Already $\mathcal{PL}[\sim]$ is Highly Expressive!

Most connectives studied in team sematics can be defined in $\mathcal{PL}[\sim]$.

The connectives below can be defined in $\mathcal{PL}[\sim]$ with polynomial blow up.

$$
\begin{aligned}
X \models \varphi \owedge \psi \quad &\Leftrightarrow \quad X \models \varphi \text{ or } X \models \psi,\\
X \models \varphi \otimes \psi \quad &\Leftrightarrow \quad \forall Y, Z \subseteq X: \text{ if } Y \cup Z = X, \text{ then } Y \models \varphi \text{ or } Z \models \psi,\\
X \models \varphi \rightarrow \psi \quad &\Leftrightarrow \quad \forall Y \subseteq X: \text{if } Y \models \varphi, \text{ then } Y \models \psi,\\
X \models \max(p_1, \dots, p_n) \quad &\Leftrightarrow \quad \{(s(p_1), \dots, s(p_n)) \mid s \in X\} = \{0,1\}^n.
\end{aligned}
$$

In $\mathcal{QPL}[\sim]$ the dual $\sim\exists\sim$ is denoted by U and it has the following semantics:

$$X \models U p\, \varphi \quad \Leftrightarrow \quad Y \models \varphi \text{ for all } Y \text{ s.t } Y \upharpoonright \mathrm{dom}(X) = X$$

Complexity Results

Logic	SAT	VAL	MC
$\mathcal{PL}$	NP [0]	coNP [0]	NC_1 [1]
$\mathcal{PL}[\mathrm{dep}(\cdot)]$	NP [3]	NEXPTIME [4]	NP [2]
$\mathcal{PL}[\perp_{\mathrm{c}}]$	NP [6]	in $\mathrm{coNEXPTIME}^{\mathrm{NP}}$ [6]	NP [6]
$\mathcal{PL}[\subseteq]$	EXPTIME [5]	coNP [6]	P [7]
$\mathcal{PL}[\perp_{\mathrm{c}}, \sim]$	AEXPTIME(poly) [6]	AEXPTIME(poly) [6]	PSPACE [6]
$\mathcal{PL}[\subseteq, \sim]$	AEXPTIME(poly) [6]	AEXPTIME(poly) [6]	PSPACE [6]

[0] Cook 1971, Levin 1973, [1] Buss 1987, [2] Ebbing, Lohmann 2012,
[3] Lohmann, Vollmer 2013, [4] V. 2014, [5] Hella, Kuusisto, Meier, Vollmer 2015,
[6] Hannula, Kontinen, V., Vollmer 2015, [7] Hella, Kuusisto, Meier, V.

Complexity Results

Logic	SAT	VAL	MC
$\mathcal{PL}$	NP [0]	coNP [0]	NC_1 [1]
$\mathcal{PL}[\mathrm{dep}(\cdot)]$	NP [3]	NEXPTIME [4]	NP [2]
$\mathcal{PL}[\bot_c]$	NP [6]	in $\text{coNEXPTIME}^{\text{NP}}$ [6]	NP [6]
$\mathcal{PL}[\subseteq]$	EXPTIME [5]	coNP [6]	P [7]
$\mathcal{PL}[\bot_c, \sim]$	AEXPTIME(poly) [6]	AEXPTIME(poly) [6]	PSPACE [6]
$\mathcal{PL}[\subseteq, \sim]$	AEXPTIME(poly) [6]	AEXPTIME(poly) [6]	PSPACE [6]

[0] Cook 1971, Levin 1973, [1] Buss 1987, [2] Ebbing, Lohmann 2012,
[3] Lohmann, Vollmer 2013, [4] V. 2014, [5] Hella, Kuusisto, Meier, Vollmer 2015,
[6] Hannula, Kontinen, V., Vollmer 2015, [7] Hella, Kuusisto, Meier, V.

Complexity Results

Below $\sim_k$ means that nesting of $\sim$ is restricted to k. Defined analogously to quantifier rank and modal depth.

Logic	$\mathcal{QPL}[\mathrm{dep}(\cdot)]$	$\mathcal{QPL}[\subseteq]$	$\mathcal{QPL}[\sim_k, \mathrm{dep}(\cdot)]$	$\mathcal{QPL}[\sim_k]$
TRUE	NEXPTIME	EXPTIME	between $\Sigma_k^{EXP}, \Sigma_{k+1}^{EXP}$	between $\Sigma_{k-2}^{EXP}, \Sigma_{k+1}^{EXP}$

Proof for $\mathcal{QPL}[\mathrm{dep}(\cdot)]$ is immediate from known results. For inclusion of $\mathcal{QPL}[\subseteq]$ in EXPTIME, we reduce to satisfiability of modal inclusion logic $\mathcal{ML}(\subseteq)$. For the rest we take a closer look of different characterisations of the exponential hierarchy.

Complexity Results

Below $\sim_k$ means that nesting of $\sim$ is restricted to k. Defined analogously to quantifier rank and modal depth.

Logic	$\mathcal{QPL}[\text{dep}(\cdot)]$	$\mathcal{QPL}[\subseteq]$	$\mathcal{QPL}[\sim_k, \text{dep}(\cdot)]$	$\mathcal{QPL}[\sim_k]$
TRUE	NEXPTIME	EXPTIME	between $\Sigma_k^{EXP}, \Sigma_{k+1}^{EXP}$	between $\Sigma_{k-2}^{EXP}, \Sigma_{k+1}^{EXP}$

Proof for $\mathcal{QPL}[\text{dep}(\cdot)]$ is immediate from known results. For inclusion of $\mathcal{QPL}[\subseteq]$ in EXPTIME, we reduce to satisfiability of modal inclusion logic $\mathcal{ML}(\subseteq)$. For the rest we take a closer look of different characterisations of the exponential hierarchy.

Oracle Turing Machines

The exponential-time hierarchy corresponds to the class of problems that can be recognized by an exponential-time alternating Turing machine with constantly many alternations.

In 1983 Orponen characterized the classes Σ_k^{EXP} and Π_k^{EXP} of the exponential time hierarchy by polynomial-time constant-alternation oracle Turing machines that query to k oracles.

Later this was generalized to exponential-time alternating Turing machines with polynomially many alternations (i.e. the class AEXPTIME(poly)) by allowing queries to polynomially many oracles.

Alternation can be replaced by a sequence of word quantifiers (Chandra, Kozen, and Stockmeyer 1981).

Characterization via Oracle Machines

The classes Σ_k^{EXP} and Π_k^{EXP} of the exponential time hierarchy are characterized by polynomial-time constant-alternation oracle Turing machines that query to k oracles.

Theorem (Orponen 1983)

A set A belongs to the class Σ_k^{EXP} iff there exist a polynomial-time constant-alternating oracle Turing machine M such that, for all x,

$$x \in A \textit{ iff } \exists A_1 \ldots Q_k A_k (M \textit{ accepts } x \textit{ with oracles } (A_1, \ldots, A_k)),$$

where $Q_2, \ldots, Q_k$ alternate between $\exists$ and $\forall$, i.e $Q_{i+1} \in \{\forall, \exists\} \setminus \{Q_i\}$.

From oracles and machines to functions and logics

Intuitively quantification of oracles can be replaced by quantification of Boolean functions and deterministic polynomial-time Turing machines can be replaced by propositional logic.

Example: DQBF by Peterson, Reif, and Azha 2001

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 $p_1, \dots, p_k$.

DQBF is NEXPTIME-complete.

From oracles and machines to functions and logics

Intuitively quantification of oracles can be replaced by quantification of Boolean functions and deterministic polynomial-time Turing machines can be replaced by propositional logic.

Example: DQBF by Peterson, Reif, and Azha 2001

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 $p_1, \ldots, p_k$.

DQBF is NEXPTIME-complete.

Functions and logics

Lohrey 2012 and Lück 2016 have characterised levels of the exponential time hierarchy such that the alternation of function quantification corresponds to the level of the hierarchy.

We need a variant that uses Skolem functions and thus generalises directly DQBF.

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 $p_1, \dots, p_n$.

Functions and logics

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 $p_1, \ldots, p_n$.

Theorem

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

From functions to team semantics

A Σ_k-ADQBF is a sentence

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

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

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

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

THANKS!