On Quantified Propositional Logics and the Exponential Time Hierarchy

Miika Hannula
Department of Computer Science
The University of Auckland
m.hannula@auckland.ac.nz

Juha Kontinen
Department of Mathematics and Statistics
University of Helsinki
juha.kontinen@helsinki.fi

Martin Lück
Institut für Theoretische Informatik
Leibniz Universität Hannover
lueck@thi.uni-hannover.de

Jonni Virtema
Department of Mathematics and Statistics
University of Helsinki
Institut für Theoretische Informatik
Leibniz Universität Hannover
jonni.virtema@helsinki.fi

We study quantified propositional logics from the complexity theoretic point of view. First we introduce alternating dependency quantified boolean formulae (ADQBF) which generalize both quantified and dependency quantified boolean formulae. We show that the truth evaluation for ADQBF is **AEXPTIME**(poly)-complete. We also identify fragments for which the problem is complete for the levels of the exponential hierarchy. Second we study propositional team-based logics. We show that DQBF formulae correspond naturally to quantified propositional dependence logic and present a general **NEXPTIME** upper bound for quantified propositional logic with a large class of generalized dependence atoms. Moreover we show **AEXPTIME**(poly)-completeness for extensions of propositional team logic with generalized dependence atoms.

1 Introduction

Deciding whether a given quantified propositional formula (qBf) is valid is a canonical **PSPACE**-complete problem [14]. *Dependency quantified propositional formulae* (dqBf) introduced by Peterson et al. [13] are variants of qBfs for which the corresponding decision problem is **NEXPTIME**-complete. Intuitively the rise of complexity stems from the fact that existential second-order quantification (existential quantification of Boolean functions) can be expressed in dqBf.

We present several logical formalisms, based on quantified propositional logic, that capture the concept of function quantification. We start from a variant of qBf where quantification happens on the level of Boolean functions in form of explicit syntactical objects. This *second-order qBf*, introduced in Section 2, captures the exponential hierarchy [10, 11]. In Section 3 we extend dqBf to incorporate universal quantification of Skolem functions, and show that second-order qBf can be translated to this novel formalism in logspace. In Sections 4 and 5 we finally study *dependence logic* and *team logic* [15] in the framework of qBf. We give efficient translations between these different formalisms and prove that they all capture the exponential hierarchy.

For a detailed exposition on dependence logics see the recent survey [2]. For the definition of the relevant complexity classes we follow the definition of *alternating Turing machines* by Chandra, Kozen and Stockmeyer [1]. The class **AEXPTIME**(poly) is the class of all problems decidable by alternating Turing machines in exponential time, i.e., $\mathcal{O}\left(2^{n^{\mathcal{O}(1)}}\right)$, and polynomially

D. Cantone and G. Delzanno (Eds.): Seventh Symposium on Games, Automata, Logics and Formal Verification (GandALF'16)
EPTCS 226, 2016, pp. 198–212, doi:10.4204/EPTCS.226.14

many alternations between existential and universal states. The classes Σ_k^{E} and Π_k^{E} of the exponential hierarchy are similar but with at most k alternations, where $k \in \mathbb{N}$, starting in an existential resp. universal state. The classes are closed both under $\leq_{\mathrm{m}}^{\mathrm{P}}$- and $\leq_{\mathrm{m}}^{\log}$-*reductions*. In this paper, if not specified otherwise, when we speak of *reductions* we mean $\leq_{\mathrm{m}}^{\log}$-*reductions*.

2 Second-order propositional logic

Second-order propositional logic is obtained from usual qBf by shifting from quantification over proposition variables to quantification over Boolean functions. In this setting Boolean functions with arity 0 correspond to propositional variables. Boolean functions with arity ≥ 1 are called *proper functions*.

For a formal definition let Φ be a set of function symbols. Every function symbol $f \in \Phi$ has its own well-defined arity $\mathrm{ar}(f)$. The syntax of $\mathsf{SO}_2(\Phi)$ is given as follows:

$$\phi ::= (\phi \wedge \phi) \mid \neg\phi \mid \exists f\phi \mid f(\underbrace{\phi, \ldots, \phi}_{n \text{ times}}), \text{ where } f \in \Phi \text{ and } \mathrm{ar}(f) = n.$$

The symbols $\forall$, $\vee$, $\rightarrow$ and $\leftrightarrow$ are defined as the usual abbreviations. We call this logic SO_2 as it essentially corresponds to second-order predicate logic SO restricted to the domain $\{0,1\}$. ESO_2 is the fragment of SO_2 where quantifiers $\exists$ for proper functions may occur only in the scope of even number of negations, i.e., universal quantification of proper functions is disallowed.

Definition 2.1 (SO_2 semantics)**.** The semantics of $\mathsf{SO}_2(\Phi)$ is defined with assignments that map variables to Boolean functions: A *Φ-interpretation* S is a map from Φ to Boolean functions, i.e., for any $f \in \Phi$ with arity $\mathrm{ar}(f) = n$, $S(f) : \{0,1\}^n \to \{0,1\}$ is an n-ary Boolean function. Given an $\mathsf{SO}_2(\Phi)$-formula ϕ, write $[\![\phi]\!]_S$ for its valuation in S, which is defined as

$$\begin{aligned}
[\![\phi \wedge \psi]\!]_S &:= [\![\phi]\!]_S \cdot [\![\psi]\!]_S,\\
[\![\neg\phi]\!]_S &:= 1 - [\![\phi]\!]_S,\\
[\![f(\phi_1, \ldots, \phi_n)]\!]_S &:= S(f)([\![\phi_1]\!]_S, \ldots, [\![\phi_n]\!]_S), \text{ where } \mathrm{ar}(f) = n,\\
[\![\exists f\phi]\!]_S &:= \max\left\{ [\![\phi]\!]_{S_F^f} \;\middle|\; F : \{0,1\}^n \to \{0,1\} \right\},
\end{aligned}$$

where S_F^f is the Φ-interpretation s. t. $S_F^f(f) = F$ and $S_F^f(g) = S(g)$ for $g \neq f$. An $\mathsf{SO}_2(\Phi)$-formula ϕ is *valid* if $[\![\phi]\!]_S = 1$ for all Φ-interpretations S. It is *satisfiable* if there is at least one S s. t. $[\![\phi]\!]_S = 1$. Finally, ϕ is *true* if it contains no free variables and it is valid. If $\overline{f} = (f_1, \ldots, f_n)$ is a tuple of function symbols, we sometimes write $\forall\overline{f}$ for $\forall f_1 \ldots \forall f_n$ and $\exists\overline{f}$ for $\exists f_1 \ldots \exists f_n$.

In the following we drop Φ and just assume that it contains sufficiently many function symbols of any finite arity.

Definition 2.2. A second-order formula is *simple* if functions have only propositions as arguments. It is in *prenex form* if all quantifiers are at the beginning of the formula, and all proper functions are quantified before propositions, i.e., it is of the form

$$\phi = \mathfrak{Q}_1 f_1 \ldots \mathfrak{Q}_n f_n \mathfrak{Q}'_1 x_1 \ldots \mathfrak{Q}'_m x_m \psi,$$

where $n, m \geq 0$, $\{\mathfrak{Q}_1, \ldots, \mathfrak{Q}_n, \mathfrak{Q}'_1, \ldots, \mathfrak{Q}'_m\} \subseteq \{\exists, \forall\}$, the functions $f_1, \ldots, f_n$ have arity > 0, the functions $x_1, \ldots, x_m$ have arity 0, and ψ is a quantifier-free propositional formula.

Write Σ_k-SO_2 for the restriction of SO_2 to formulae of the form $\phi = \eth_1 \overline{f}_1 \ldots \eth_k \overline{f}_k \psi$, where $k \in \mathbb{N}$, $\eth_1 = \exists$, $\{\eth_2, \ldots, \eth_n\} \subseteq \{\exists, \forall\}$, each f_i is a possibly empty tuple of proper function symbols, and ψ is an SO_2 formula in which all quantifications are of propositional variables. If $\eth_1 = \forall$, the corresponding fragment is called Π_k-SO_2. The restriction to formulae where each function symbol f occurs only with a fixed tuple $\overline{a}_f$ of arguments is denoted by the suffix $\cdot^{\mathrm{u}}$.

Let L be some logic. The problems TRUE(L), SAT(L), and VAL(L) are defined as follows: Given a formula $\phi \in \mathsf{L}$, decide whether the formula is true, satisfiable, or valid, respectively.

The restricted variant SO_2 of second-order logic SO is obviously decidable due to its finite domain. Moreover it captures exactly the levels of the exponential hierarchy.

Proposition 2.3 ([10])**.** *For any $k \geq 1$ the following problems restricted to prenex formulae are complete w. r. t. $\leq^{\mathrm{P}}_{\mathrm{m}}$:* TRUE($\Sigma_k$-$\mathsf{SO}_2$) *is Σ^{E}_k-complete,* TRUE(Π_k-SO_2) *is Π^{E}_k-complete, and* TRUE(SO_2) *is* **AEXPTIME**(poly)*-complete.*

3 Dependency quantified propositional formulae

In the previous section we considered second-order propositional logic. Now we turn to logics in which functions are quantified only implicitly in form of *Skolem functions* of variables. Well-known such logics are *dependency quantified propositional formulae* (DQBF), but also *independence-friendly logic* (IF) by Hintikka and Sandu [7]. They have in common the syntactical property that Skolem functions are specified by denoting *constraints* for quantified variables. It is worth noting that we get the standard quantified propositional logic by restricting attention to formulae of SO_2 in which it is only allowed to quantify functions of arity 0. Furthermore, DQBF correspond to the fragment $\mathsf{ESO}^{\mathrm{u}}_2$. In this section we introduce a generalization of DQBF that analogously corresponds to the full logic $\mathsf{SO}^{\mathrm{u}}_2$.

We start by giving the definition of DQBF and some required notation. For the definitions related to DQBF, we follow Virtema [16]. For a set C of propositional variables, we denote by $\overline{c}$ the canonically ordered tuple of the variables in the set C. We refer to usual propositional assignments, in contrast to function assignments, by s instead of S.

A formula that does not have any free variables is called *closed* (or a *sentence*). A *simple qBf* is a closed qBf of the type $\phi := \forall p_1 \cdots \forall p_n \exists q_1 \cdots \exists q_m \theta$, where θ is a propositional formula and the propositional variables p_i, q_j are all distinct. Any tuple $(C_1, \ldots, C_m)$ such that $C_1, \ldots, C_m \subseteq \{p_1, \ldots, p_n\}$ is called a *constraint* for ϕ.

Definition 3.1. A simple qBf $\forall p_1 \cdots \forall p_n \exists q_1 \cdots \exists q_m \theta$ is *true under a constraint* $(C_1, \ldots, C_m)$ if there exist functions $f_1, \ldots, f_m$ with $f_i \colon \{0,1\}^{|C_i|} \to \{0,1\}$ such that for each assignment $s \colon \{p_1, \ldots, p_n\} \to \{0,1\}$, $s(q_1 \mapsto f_1(s(\overline{c}_1)), \ldots, q_m \mapsto f_m(s(\overline{c}_m))) \models \theta$.

A *dependency quantified propositional formula* is a pair $(\phi, \overline{C})$, where ϕ is a simple quantified propositional formula and $\overline{C}$ is a constraint for ϕ. We say that $(\phi, \overline{C})$ is *true* if ϕ is true under the constraint $\overline{C}$. Let DQBF denote the set of all dependency quantified propositional formulae.

Proposition 3.2 ([13, 5.2.2])**.** TRUE(DQBF) *is* **NEXPTIME***-complete problem w.r.t. $\leq^{\log}_{\mathrm{m}}$.*

We next introduce a novel variant of DQBF called *alternating dependency quantified propositional formulae* ADQBF. The syntax of *alternating quantified propositional formulae* extends the syntax of quantified propositional formulae with a new quantifier U. The quantifier U is used to express universal quantification of Skolem functions of propositional symbols.

Definition 3.3. A *simple Σ_k-alternating qBf* is a closed formula of the form

$$\phi := \forall p_1 \cdots \forall p_n \, (\exists q_1^1 \cdots \exists q_{j_1}^1)\, (\mathsf{U}\, q_1^2 \cdots \mathsf{U}\, q_{j_2}^2)\, (\exists q_1^3 \cdots \exists q_{j_3}^3) \ldots (Q q_1^k \cdots Q q_k^k)\, \theta,$$

where $Q \in \{\exists, \mathsf{U}\}$, θ is a propositional formula and the quantified propositional variables are all distinct. Similarly, a *simple Π_k-alternating qBf* is a closed formula of the form

$$\phi := \forall p_1 \cdots \forall p_n \, (\mathsf{U}\, q_1^1 \cdots \mathsf{U}\, q_{j_1}^1)\, (\exists q_1^2 \cdots \exists q_{j_2}^2)\, (\mathsf{U}\, q_1^3 \cdots \mathsf{U}\, q_{j_3}^3) \ldots (Q q_1^k \cdots Q q_{j_k}^k)\, \theta.$$

A *simple alternating qBf* is a simple Σ_i-alternating or Π_i-alternating qBf for some i. Any tuple $(C_1^1, \ldots, C_{j_k}^k)$ such that $C_1^1, \ldots, C_{j_k}^k \subseteq \{p_1, \ldots, p_n\}$ is called a *constraint* for ϕ.

Definition 3.4. Truth of a simple alternating qBf under a constraint $(C_1^1, \ldots, C_{j_k}^k)$ is defined by generalizing Definition 3.1 such that each $\mathsf{U}\, q_i^l$ is interpreted as universal quantification over (Skolem) functions $f_i^l \colon \{0,1\}^{|C_i^l|} \to \{0,1\}$.

Example 3.5: Let $\phi := \forall p_1 \forall p_2 \exists q_1 \mathsf{U}\, q_2\, \theta$ and $C := (\{p_2\}, \{p_1\})$. Now (ϕ, C) is true iff there exists a function $f_1 : \{0,1\} \to \{0,1\}$ s. t. for all functions $f_2 : \{0,1\} \to \{0,1\}$ it holds that for each assignment $s \colon \{p_1, p_2\} \to \{0,1\}$, $s(q_1 \mapsto f_1(s(p_2)), q_2 \mapsto f_2(s(p_1))) \models \theta$.

Example 3.6: The formula $\forall \overline{x} (\mathsf{U}\, y \exists z) \neg y \leftrightarrow z$ under the constraint $(\{\overline{x}\}, \{\overline{x}\})$ expresses that every $|\overline{x}|$-ary Boolean function has a negation.

Definition 3.7. The set ADQBF is the set of all pairs $(\phi, \overline{C})$ where ϕ is a simple alternating qBf and $\overline{C}$ is a constraint of ϕ. The set Σ_k-ADQBF (Π_k-ADQBF) is then the subset of ADQBF where ϕ is Σ_k-alternating (Π_k-alternating).

Lemma 3.8. *For all $k \geq 1$ it holds that* $\mathrm{TRUE}(\Sigma_k\text{-ADQBF}) \in \Sigma_k^{\mathrm{E}}$, $\mathrm{TRUE}(\Pi_k\text{-ADQBF}) \in \Pi_k^{\mathrm{E}}$, *and* $\mathrm{TRUE}(\mathrm{ADQBF}) \in \mathbf{AEXPTIME}(\mathrm{poly})$.

Proof. We give a brute-force algorithm. Let the universal quantified prefix of the given qBf be $\forall p_1 \ldots \forall p_n$. For every $\exists$-quantified block $\exists q_1 \ldots \exists q_j$ with constraints $C_1, \ldots, C_j$, existentially guess and write down a Boolean function from the variables $C_i \subseteq \{p_1, \ldots, p_n\}$ for every q_i. For every U-quantified block, switch to universal branching and do the same. The quantifier-free part can then be evaluated in deterministic exponential time for every possible assignment to $p_1, \ldots, p_n$. The algorithm runs in exponential time and its alternations are bounded by the alternations of $\exists$ and U quantifiers in the formula. □

For the hardness direction we first show how the uniqueness property can be obtained for arbitrary SO_2-formulae by introducing additional function symbols. The following lemma will be needed in the sequel (see, e.g., Väänänen [15]).

Lemma 3.9. *Let S be an SO_2 interpretation, x a propositional variable, f and f' function variables with $\mathrm{ar}(f') = \mathrm{ar}(f) + 1$, and $\phi(x, f)$ an SO_2-formula in which x and f occur only as free variables and in which f' does not occur. Then it holds that $S \models \forall x \exists f \phi \Leftrightarrow S \models \exists f' \forall x \phi'$ and $S \models \exists x \forall f \phi \Leftrightarrow S \models \forall f' \exists x \phi'$, where ϕ' is obtained from ϕ by replacing $f(\overline{y})$ with $f'(x, \overline{y})$.*

Lemma 3.10. *Every Σ_k-SO_2-sentence (Π_k-SO_2-sentence, SO_2-sentence) ϕ can be translated to an equivalent simple prenex Σ_k-SO_2^u-sentence (Π_k-SO_2^u-sentence, SO_2^u-sentence) ψ in polynomial time.*

Proof. First we prove that ϕ can be transformed into a simple formula in polynomial time. Whenever a subformula $\xi = f(\psi_1, \ldots, \psi_i, \ldots, \psi_n)$ occurs and ψ_i is not a proposition, then replace ξ by $\forall b((b \leftrightarrow \psi_i) \rightarrow f(\psi_1, \ldots, b, \ldots, \psi_n))$, where b is a new proposition symbol.

Then by the usual translation move all quantifiers to the beginning of the formula. Swap the order of the quantifiers according to Lemma 3.9 until all quantified proper function symbols precede the quantified propositions. Such obtained ϕ is simple and of the form $\mathfrak{Q}_1 f_1 \ldots \mathfrak{Q}_n f_n \forall p_1 \exists q_1 \ldots \forall p_m \exists q_m \theta$ where θ is quantifier-free, $\{\mathfrak{Q}_1, \ldots, \mathfrak{Q}_n\} \subseteq \{\exists, \forall\}$, and $f_1, \ldots, f_n$ are the only proper functions that occur in ϕ.

Finally we "split" the quantified function symbols in ϕ s. t. every proper function symbol occurs with exactly one fixed argument tuple. Let $\chi = \exists f \mathfrak{Q}_1 g_1 \ldots \mathfrak{Q}_k g_k \mathfrak{Q}'_1 p_1 \ldots \mathfrak{Q}'_m p_m \, \theta(f(\overline{x}_1), \ldots, f(\overline{x}_n))$ where θ is quantifier-free be a subformula of ϕ, meaning that f occurs in θ at n different positions with n (possibly different) argument tuples $\overline{x}_1, \ldots, \overline{x}_n$. In what follows, $f_1 \ldots f_n$ and $\overline{y}_1, \ldots, \overline{y}_n$ are assumed to be distinct and fresh. Then ϕ is equivalent to the formula obtained from ϕ by substituting χ by the formula $\exists f_1 \ldots \exists f_n \mathfrak{Q}_1 g_1 \ldots \mathfrak{Q}_k g_k \mathfrak{Q}'_1 p_1 \ldots \mathfrak{Q}'_m p_m \forall \overline{y}_1 \ldots \forall \overline{y}_n \, (\psi_1 \wedge \psi_2)$, where $\psi_1 := \bigwedge_{i=1}^{n-1} ((\overline{y}_i \leftrightarrow \overline{y}_{i+1}) \rightarrow (f_i(\overline{y}_i) \leftrightarrow f_{i+1}(\overline{y}_{i+1})))$ ensures that the functions $f_1, \ldots, f_n$ are all the same, and $\psi_2 := (\bigwedge_{i=1}^{n} \overline{x}_i \leftrightarrow \overline{y}_i) \rightarrow \Big(\theta(f_1(\overline{y}_1), \ldots, f_n(\overline{y}_n))\Big)$ simulates $\theta(f_1(\overline{x}_1), \ldots, f_n(\overline{x}_n))$. The "split" of universal quantifiers is done analogously. Clearly ϕ remains simple and in prenex form.

The steps introduced above do not add additional alternations of function quantifiers, hence the resulting formula is now an Σ_k-$\mathsf{SO}_2^{\mathrm{u}}$ resp. Π_k-$\mathsf{SO}_2^{\mathrm{u}}$ resp. $\mathsf{SO}_2^{\mathrm{u}}$ sentence. □

Theorem 3.11. *Let $k \geq 1$. For odd k the problem* TRUE(Σ_k-ADQBF) *is Σ_k^{E}-complete. For even k the problem* TRUE(Π_k-ADQBF) *is Π_k^{E}-complete. The problem* TRUE(ADQBF) *is* **AEXPTIME**(poly)*-complete.*

Proof. The membership was shown in Lemma 3.8. For the hardness we start with the problem TRUE(Σ_k-ADQBF). We give a reduction from TRUE(Σ_k-$\mathsf{SO}_2^{\mathrm{u}}$) which is by Proposition 2.3 and Lemma 3.10 $\leq_{\mathrm{m}}^{\mathrm{P}}$-complete for Σ_k^{E}. Let $\phi := Q_1 \overline{f_1} Q_2 \overline{f_2} \ldots \exists \overline{f_k} \forall p_1 \exists q_1 \ldots \forall p_n \exists q_n \psi$, where ψ is quantifier-free, be a simple prenex Σ_k-$\mathsf{SO}_2^{\mathrm{u}}$-sentence. Note that $Q_k = \exists$ since k is odd. For each function symbol f_i that occurs in ψ, let $(a_1^i, \ldots, a_{m_i}^i)$ denote the unique tuple that occurs as an argument of f_i. Each of these functions with arguments can be simulated by a single constrained propositional variable; a problem in this translation is however that some a_j^i may be existentially quantified and thus not part of the $p_1, \ldots, p_n$. However, this problem can be easily solved by introducing fresh universally quantified propositional variables:

Assume that ψ is in negation normal form. Any subformula $f_i(\overline{a})$ is replaced by $\forall \overline{r}((\overline{a} \leftrightarrow \overline{r}) \rightarrow f_i(\overline{r}))$, where $\overline{r} = r_1, \ldots, r_{\mathrm{ar}(f_i)}$ are fresh distinct variables. Analogously, $\neg f_i(\overline{a})$ is replaced by $\forall \overline{r}((\overline{a} \leftrightarrow \overline{r}) \rightarrow \neg f_i(\overline{r}))$. Clearly the such obtained sentence can be transformed to prenex form by just moving all the freshly introduced quantifiers to the right end of the quantifier prefix. The such obtained sentence is equivalent to ϕ. Thus we may assume w.l.o.g. that if $f_i(a_1^i, \ldots, a_{m_i}^i)$ occurs in ϕ, then $\{a_1^i, \ldots, a_{m_i}^i\} \subseteq \{p_1, \ldots, p_n\}$.

For the reduction to ADQBF now just consider $\forall p_1 \ldots \forall p_n$ as universal quantified variables. For every $f_i(a_1^i, \ldots, a_{m_i}^i)$ introduce a quantified variable $\exists f_i$ with constraint $\{a_1^i, \ldots, a_{m_i}^i\}$ if f_i is existentially quantified, and introduce $\mathsf{U} f_i$ with the same constraint otherwise. For every q_i introduce a quantified variable $\exists q_i$ with constraint $\{p_1, \ldots, p_i\}$. Let ψ' denote the formula obtained from ψ by substituting, for each i, $f_i(a_1^i, \ldots, a_{m_i}^i)$ by f_i. Thus the resulting ADQBF has the form $\phi' := \forall p_1 \ldots \forall p_n \, (\exists \overline{f_1}) \, (\mathsf{U} \overline{f_2}) \, \ldots \, (\exists \overline{f_k} \exists q_1 \ldots \exists q_n) \, \psi'$. Since the final function quantifier Q_k was existential, we can merge the functions $\overline{f_k}$ and the quantified propositions $q_1 \ldots q_n$ to

a single existentially quantified block in ADQBF. So ϕ' is in Σ_k-ADQBF, and by definition of ADQBF, ϕ' is true under the constraint C (where C is constructed as above) if and only if $\phi \in \mathrm{TRUE}(\Sigma_k\text{-}\mathsf{SO}_2^{\mathrm{u}})$.

For general $\mathsf{SO}_2^{\mathrm{u}}$ formulae we can again assume that the last function quantifier is existential. The same holds for Π_k-$\mathsf{SO}_2^{\mathrm{u}}$ formulae if k is even. In these cases a similar proof yields a reduction to ADQBF resp. Π_k-ADQBF. □

Theorem 3.12. *Let $k \geq 2$. For even k the problem* TRUE(Σ_k-ADQBF) *is $\Sigma^{\mathrm{E}}_{k-1}$-complete. For odd k the problem* TRUE(Π_k-ADQBF) *is Π^{E}_{k-1}-complete.*

Proof. The hardness results follow from Theorem 3.11. For inclusion, we prove the case for Σ_k-ADQBF. We give a $\leq^{\log}_{\mathrm{m}}$-reduction from TRUE(Σ_k-ADQBF) to TRUE(Σ_{k-1}-ADQBF). The result then follows from Theorem 3.11. The case for Π_k-ADQBF is analogous. Consider a formula $\phi := \forall p_1 \cdots \forall p_n\, (\exists q_1^1 \cdots \exists q_{j_1}^1)\, (\mathsf{U}\, q_1^2 \cdots \mathsf{U}\, q_{j_2}^2) \ldots (\mathsf{U}\, q_1^k \cdots \mathsf{U}\, q_{j_k}^k)\, \theta$ and a constraint $C = (C_1^1, \ldots, C_{j_k}^k)$. We claim that (ϕ, C) is equivalent to

$$\phi' := \forall p_1 \cdots \forall p_n \forall q_1^k \cdots \forall q_{j_k}^k\, (\exists q_1^1 \cdots \exists q_{j_1}^1)\, (\mathsf{U}\, q_1^2 \cdots \mathsf{U}\, q_{j_2}^2) \ldots (\exists q_1^{k-1} \cdots \exists q_{j_{k-1}}^{k-1})\, \theta$$

under the constraint $C' = (C_1^1, \ldots, C_{j_{k-1}}^{k-1})$.

By definition (ϕ, C) is true if and only if for all extensions of the tuple of quantified Skolem functions $f_1^1, \ldots, f_{j_{k-1}}^{k-1}$ (some of which are existentially/universally quantified) and for all extensions of the Skolem functions f_i^k it holds that:

$$\forall t \in T : t \models \theta, \text{ where } T := \{s(q_1^1 \mapsto f_1^1(\bar{c}_1^1), \ldots, q_{j_k}^k \mapsto f_{j_k}^k(\bar{c}_{j_k}^k)) \mid s : \{p_1, \ldots, p_n\} \to \{0,1\}\} \tag{1}$$

Note that, in fact, T is the set of all expansions of assignments $s(q_1^1 \mapsto f_1^1(\bar{c}_1^1), \ldots, q_{j_{k-1}}^{k-1} \mapsto f_{j_{k-1}}^{k-1}(\bar{c}_{j_{k-1}}^{k-1}))$, $s : \{p_1, \ldots, p_n\} \to \{0,1\}$, into domain $\{p_1, \ldots, p_n, q_1^1, \ldots, q_{j_k}^k\}$. Thus (1) can be equivalently written as

$$\forall t \in T : t \models \theta, \text{ where } T := \{s(q_1^1 \mapsto f_1^1(\bar{c}_1^1), \ldots, q_{j_{k-1}}^{k-1} \mapsto f_{j_{k-1}}^{k-1}(\bar{c}_{j_{k-1}}^{k-1})) \mid s : \{p_1, \ldots, p_n, q_1^k, \ldots, q_{j_k}^k\} \to \{0,1\}\} \tag{2}$$

Now note that as the constraints and quantifiers for $q_1^1, \ldots, q_{j_{k-1}}^{k-1}$ are exactly the same in C and in C' each extension of the tuple of quantified Skolem functions $f_1^1, \ldots, f_{j_{k-1}}^{k-1}$ in the evaluation of (ϕ, C) can be directly interpreted in (ϕ, C'), and vice versa. From this together with the equivalence of (1) and (2), we conclude that (ϕ, C) and (ϕ, C') are equivalent. □

Using the translation from $\mathsf{SO}_2^{\mathrm{u}}$ to ADQBF introduced in the proof of Theorem 3.11, we obtain the following corollary.

Corollary 3.13. *For every SO_2-sentence ϕ there is a polynomial time computable* ADQBF*-instance $(\psi, \overline{C})$ which is true iff ϕ is true. For every ESO_2-sentence ϕ there is a polynomial time computable* DQBF*-instance $(\psi, \overline{C})$ which is true iff ϕ is true.*

4 Quantified propositional logics with team semantics

The study of propositional logics with team semantics has so far concentrated on extensions of propositional logics with different dependency notions such as functional dependence, independence and inclusion. Here we extend the perspective to quantified propositional logics.

4.1 Basic notions and results

In the team semantics context it is usual to consider assignments over finite sets of proposition symbols. We begin by fixing some notation. Let D be a finite, possibly empty set of proposition symbols. A set X of assignments $s\colon D \to \{0,1\}$ is called a *team*. The set D is the *domain* $\mathsf{Dom}(X)$ of X. We denote by 2^D the set of *all assignments* $s\colon D \to \{0,1\}$. If $\overline{p} = (p_1,\ldots,p_n)$ is a tuple of propositions and s is an assignment, we write $s(\overline{p})$ for $(s(p_1),\ldots,s(p_n))$. If $b \in \{0,1\}$ and s is an assignment with domain D, we let $s(q \mapsto b)$ denote the assignment with domain $D \cup \{q\}$ defined as follows: $s(q \mapsto b)(p) = b$ if $p = q$ and $s(q \mapsto b)(p) = s(p)$ if $p \neq q$.

Let X be a team. A function $F : X \to \{\{0\},\{1\},\{0,1\}\}$ is called a *supplementing function* of X. Supplementing functions are used for giving semantics for existential quantifiers. For a proposition symbol p, we define $X[F/p] := \{s(p \mapsto b) \mid s \in X, b \in F(s)\}$. We say that $X[F/p]$ is a *supplemented team* of X in p.

For $A \subseteq \{0,1\}$ we define $X[A/p] := \{s(p \mapsto b) \mid s \in X, b \in A\}$. The team $X[\{0,1\}/p]$ is the *duplicating team* of X in p. Duplicating teams are used to give semantics for universal quantifiers.

Let Φ be a set of proposition symbols. The syntax of *quantified propositional team logic* $\mathsf{QPTL}(\Phi)$ is given by the following grammar:

$$\phi ::= p \mid \neg p \mid (\phi \wedge \phi) \mid (\phi \vee \phi) \mid {\sim}\phi \mid \forall p\,\phi \mid \exists p\,\phi, \text{ where } p \in \Phi,$$

Its quantifier-free fragment is called *propositional team logic* $\mathsf{PTL}(\Phi)$, similar to the first-order team logic TL by Väänänen [15]. Likewise its $\sim$-free fragment is called *quantified propositional logic* $\mathsf{QPL}(\Phi)$. The usual *propositional logic* $\mathsf{PL}(\Phi)$ is then just the quantifier-free fragment of $\mathsf{QPL}(\Phi)$.

Let us denote by $\mathsf{Prop}(\phi)$ the set of proposition symbols that occur in ϕ, and by $\mathsf{Fr}(\phi)$ the set of proposition symbols that occur free in ϕ. We sometimes write $\phi(p_1,\ldots,p_n)$ to denote that ϕ is a formula whose free proposition symbols are in $\{p_1,\ldots,p_n\}$. A formula in which no proposition symbol occurs free is called a *sentence*. We denote by $\models_{\mathsf{PL}}$ the ordinary satisfaction relation of quantified propositional logic defined via assignments in the standard way. Next we give team semantics for quantified propositional logic. The semantics for the quantifiers follow the corresponding definitions of first-order team semantics (as quantified propositional logic can be seen as first-order logic over domain $\{0,1\}$).

Definition 4.1 (Lax team semantics)**.** Let Φ be a set of atomic propositions and let X be a team. The satisfaction relation $X \models \phi$ for $\phi \in \mathsf{QPTL}(\Phi)$ is defined as follows.

$$\begin{aligned}
X \models p &\;\Leftrightarrow\; \forall s \in X : s(p) = 1 \text{ if } p \in \Phi.\\
X \models \neg p &\;\Leftrightarrow\; \forall s \in X : s(p) = 0 \text{ if } p \in \Phi.\\
X \models (\phi \wedge \psi) &\;\Leftrightarrow\; X \models \phi \text{ and } X \models \psi.\\
X \models (\phi \vee \psi) &\;\Leftrightarrow\; Y \models \phi \text{ and } Z \models \psi, \text{ for some } Y, Z \text{ such that } Y \cup Z = X.\\
X \models {\sim}\phi &\;\Leftrightarrow\; X \not\models \phi.\\
X \models \exists p\,\phi &\;\Leftrightarrow\; X[F/p] \models \phi \text{ for some function } F : X \to \{\{0\},\{1\},\{0,1\}\}.\\
X \models \forall p\,\phi &\;\Leftrightarrow\; X[\{0,1\}/p] \models \phi.
\end{aligned}$$

We say that a sentence ϕ is *true* if $\{\emptyset\} \models \phi$, i.e., if the team with just the empty assignment satisfies ϕ.

The next proposition shows that the team semantics and the ordinary semantics for QPL-formulae coincide.

Proposition 4.2 (Flatness property [15])**.** *Let ϕ be a formula of quantified propositional logic and let X be a propositional team. Then $X \models \phi$ iff $\forall s \in X : s \models_{\mathsf{PL}} \phi$.*

The syntax of *quantified propositional dependence logic* $\mathsf{QPD}(\Phi)$ is obtained by extending the syntax of $\mathsf{QPL}(\Phi)$ by the following grammar rule for each $n \in \mathbb{N}$:

$$\phi ::= \mathrm{dep}(p_1, \ldots, p_n, q), \text{ where } p_1, \ldots, p_n, q \in \Phi.$$

The meaning of the *propositional dependence atom* $\mathrm{dep}(p_1, \ldots, p_n, q)$ is that the truth value of the proposition symbol q is functionally determined by the truth values of the proposition symbols $p_1, \ldots, p_n$. The semantics for the atoms is defined as follows: $X \models \mathrm{dep}(p_1, \ldots, p_n, q)$ iff for all $s, t \in X : s(p_1) = t(p_1), \ldots, s(p_n) = t(p_n)$ implies $s(q) = t(q)$.

The next well-known result is proved in the same way as the analogous result for first-order dependence logic [15].

Proposition 4.3 (Downwards closure)**.** *Let ϕ be a* QPD*-formula and let $Y \subseteq X$ be propositional teams. Then $X \models \phi$ implies $Y \models \phi$.*

In this article we study also a variant of QPD obtained by replacing dependence atoms by the so-called *inclusion atoms*. The syntax of *quantified propositional inclusion logic* $\mathsf{QPLInc}(\Phi)$ is obtained by extending the syntax of $\mathsf{QPL}(\Phi)$ by the grammar rule $\phi ::= (p_1, \ldots, p_n) \subseteq (q_1, \ldots, q_n)$ for every $n \geq 0$, where $p_1, \ldots, p_n, q_1, \ldots, q_n \in \Phi$. The semantics for propositional inclusion atoms is defined as follows: $X \models \overline{p} \subseteq \overline{q}$ iff $\forall s \in X \, \exists t \in X : s(\overline{p}) = t(\overline{q})$.

It is easy to check that QPLInc is not downward closed (cf. Proposition 4.3). However, analogously to FO-inclusion-logic [3], QPLInc is closed w. r. t. unions:

Proposition 4.4 (Closure under unions)**.** *Let $\phi \in$ QPLInc and let X_i, for $i \in I$, be teams. Suppose that $X_i \models \phi$ for each $i \in I$. Then $\bigcup_{i \in I} X_i \models \phi$.*

Definition 4.5. Let L be a propositional logic with team semantics. Recall that a sentence $\phi \in \mathsf{L}$ is *true* if $\{\emptyset\} \models \phi$. A formula $\phi \in \mathsf{L}$ is *satisfiable* if there exists a non-empty team X such that $X \models \phi$. A formula $\phi \in \mathsf{L}$ is *valid* if $X \models \phi$ holds for all teams X such that the proposition symbols in $\mathrm{Fr}(\phi)$ are in the domain of X. The problems TRUE(L), SAT(L), and VAL(L) are defined in the obvious way: Given a formula $\phi \in \mathsf{L}$, decide whether the formula is true, satisfiable or valid, respectively.

The following results for PLInc and MInc are implicitly shown by Hella et al. [6]. They state the results using **PSPACE**-reductions, but in fact their reductions run in polynomial time.

Proposition 4.6 ([6, 16])**.** SAT(PLInc) *and* SAT(MInc) *are* **EXPTIME***-complete w. r. t.* $\leq_{\mathrm{m}}^{\mathrm{P}}$*-reductions.* VAL(PD) *is* **NEXPTIME***-complete w. r. t.* $\leq_{\mathrm{m}}^{\log}$*-reductions.*

The following lemma is a direct consequence of a result of Galliani et al. [4, Lemma 14], where an analogous claim is proven in the first-order setting over structures with universe size at least 2. The result follows by the obvious back-and-forth translations between propositional logic and first-order logic where truth of a propositional formula is replaced with satisfaction by the first-order structure that has universe $\{0, 1\}$ and two constants interpreted as 0 and 1.

Lemma 4.7 ([4])**.** *Any formula ϕ in L, where $\mathsf{L} \in \{$QPD, QPLInc$\}$, is logically equivalent to a polynomial size formula $\partial_1 p_1 \ldots \partial_k p_k \psi$ in L where ψ is quantifier-free and $\{\partial_1, \ldots, \partial_k\} \subseteq \{\exists, \forall\}$ for $i = 1, \ldots, n$.*

4.2 Complexity of quantified propositional logics

In this section we consider the complexity of quantified propositional dependence and inclusion logic. In the latter case, we reduce the problem to the satisfiability problem of *modal inclusion logic*, MInc, as defined by Hella et al. [6].

Proposition 4.8. TRUE(QPD) *is* **NEXPTIME***-complete w. r. t.* $\leq_{\mathrm{m}}^{\log}$*-reductions.*

Proof. We show a reduction from VAL(PD) to TRUE(QPD). By Proposition 4.6, the former is **NEXPTIME**-hard and thus the latter is as well. Let ϕ be a PD-formula and let $\overline{p}$ be the tuple of proposition symbols that occur in ϕ. Note first that, since PD is downward closed, it follows that ϕ is valid if and only if $2^{\overline{p}} \models \phi$, where $2^{\overline{p}}$ is the team that contains exactly all propositional assignments with domain $\overline{p}$. Thus it follows that the PD-formula ϕ is valid if and only if the QPD-formula $\forall \overline{p}\phi$ is true.

The fact that TRUE(QPD) is in **NEXPTIME** follows from the obvious brute force algorithm that uses non-determinism to guess the witnessing teams for existential quantifiers and disjunctions. □

Theorem 4.9. TRUE(QPLInc) *is* **EXPTIME***-complete w. r. t.* $\leq_{\mathrm{m}}^{\mathrm{P}}$*-reductions.*

Proof. We give a $\leq_{\mathrm{m}}^{\mathrm{P}}$-reduction from SAT(PLInc) to TRUE(QPLInc). Since, by Proposition 4.6, SAT(PLInc) is **EXPTIME**-hard under $\leq_{\mathrm{m}}^{\mathrm{P}}$-reductions, it follows that TRUE(QPLInc) is as well. Let ϕ be a formula of PLInc and let $\overline{p}$ be the tuple of proposition symbols that occur in ϕ. Clearly there exists a nonempty propositional team X such that $X \models \phi$ if and only if $\{\emptyset\} \models \exists \overline{p}\phi$.

We will next show that TRUE(QPLInc) is in **EXPTIME**. We do this via a polynomial time translation $\phi \mapsto \phi^*$ from QPLInc to MInc. The translation is designed such that ϕ is true if and only if ϕ^* is satisfied by a non-empty team in a Kripke structure. Since, by Proposition 4.6, SAT(MInc) is in **EXPTIME**, it follows that TRUE(QPLInc) is as well. In our construction, the idea is that points in a *Kripke* model will correspond to propositional assignments, and existential and universal quantifiers are simulated by diamonds and boxes, respectively.

First we will enforce a binary (assignment) tree in our structure. Branching in the tree will correspond to quantification of proposition variables. The binary tree is forced in the standard way by modal formulae: The formula $\mathsf{branch}(p_i) := \Diamond p_i \wedge \Diamond \neg p_i$ forces that there are ≥ 2 successor states which disagree on a proposition p_i. The formula $\mathsf{store}(p_i) := (p_i \wedge \Box p_i) \vee (\neg p_i \wedge \Box \neg p_i)$ is used to propagate chosen values for p_i to successors in the tree. Now define

$$\mathsf{tree}(p,n) := \mathsf{branch}(p_1) \wedge \bigwedge_{i=1}^{n-1} \Box^i \Big(\mathsf{branch}(p_{i+1}) \wedge \bigwedge_{j=1}^{i} \mathsf{store}(p_j) \Big),$$

where $\Box^i \phi := \overbrace{\Box \cdots \Box}^{i \text{ many}} \phi$. The formula $\mathsf{tree}(p,n)$ forces a complete binary assignment tree of depth n for proposition symbols $p_1, \ldots, p_n$. Notice that $\mathsf{tree}(p,n)$ is an ML-formula and hence has the flatness property, analogously to Proposition 4.2 [17]. When ϕ is a QPLInc-formula, we denote by ϕ' the MInc formula that is obtained from ϕ by substituting each existential quantifier $\exists p$ by $\Diamond$ and each universal quantifier $\forall p$ by $\Box$.

We are now ready to state our reduction. Let ϕ be an arbitrary QPLInc-formula in the normal form of Lemma 4.7. W.l.o.g. we may assume that $\phi = \partial_1 p_1 \ldots \partial_n p_n \psi$, where $\{\partial_1, \ldots, \partial_n\} \subseteq \{\exists, \forall\}$ and ψ is quantifier-free. Define $\phi^* := \mathsf{tree}(p,n) \wedge \phi'$. It is straightforward to check that, indeed, ϕ is true if and only if ϕ^* is satisfiable. □

4.3 Propositional team logic and ADQBF

In [5] it was established that the validity and satisfiability problem of PTL extended with either inclusion or independence atom is complete for **AEXPTIME**(poly). In the extended version of the paper it is shown that, in fact, this holds this holds already for PTL. Here we generalize this result by establishing connections between fragments of team-based logics and ADQBF.

First observe that sentences of ADQBF can be equivalently interpreted as sentences of QPTL extended with dependence atoms, denoted by QPTL(dep). This translation is analogous to the translation from SO to first-order team logic (see [8, 12]). Let

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

be a simple alternating qBf with constraints $(C_1^1, \ldots, C_{j_k}^k)$. Recall that for a set of variables C, we denote by $\bar{c}$ the canonically ordered tuple consisting of the variables in C. Let ϕ^* denote the following QPTL(dep)-sentence:

$$\begin{aligned} &\forall p_1 \cdots \forall p_n \, (\exists q_1^1 \cdots \exists q_{j_1}^1)\, (\mathsf{U}\, q_1^2 \cdots \mathsf{U}\, q_{j_2}^2)\, (\exists q_1^3 \cdots \exists q_{j_3}^3) \ldots (Q q_1^k \cdots Q q_{j_k}^k) \\ &\qquad \sim \left[\sim (p \wedge \neg p) \wedge \bigwedge_{\substack{1 \le i \le k \\ i \text{ is even} \\ 1 \le l \le j_i}} \mathrm{dep}\left(\bar{c}_l^i, y_l^i\right) \right] \vee \left[\left(\bigwedge_{\substack{1 \le i \le k \\ i \text{ is odd} \\ 1 \le l \le j_i}} \mathrm{dep}\left(\bar{c}_l^i, y_l^i\right) \right) \wedge \theta \right] \end{aligned} \tag{3}$$

Above the quantifier $\mathsf{U}\, q$ is treated as a shorthand for the expression $\sim\exists q\sim$.[1] It is straightforward to check that ϕ is true under the constraint $(C_1^1, \ldots, C_{j_k}^k)$ if and only if ϕ^* is true. Thus we obtain fragments of QPTL(dep) that express complete problems for levels of the exponential hierarchy, see Theorem 3.11. For $k = 1$, we obtain a translation from DQBF to QPD. It is noteworthy that, in fact, by the above translation we obtain a close connection between the classes Σ_k^{E} and Π_k^{E}, and the fragment of QPTL(dep) of sentences with $\le k$ nested $\sim$s ($\mathsf{deg}_\sim(\phi)$); formally defined as follows:

$$\begin{aligned} &\mathsf{deg}_\sim(\forall p\phi) := \mathsf{deg}_\sim(\exists p\phi) := \mathsf{deg}_\sim(\phi), \mathsf{deg}_\sim(\phi \vee \psi) := \mathsf{deg}_\sim(\phi \wedge \psi) := \max\{\mathsf{deg}_\sim(\phi), \mathsf{deg}_\sim(\psi)\}, \\ &\mathsf{deg}_\sim(\neg\phi) := \mathsf{deg}_\sim(\phi), \mathsf{deg}_\sim(\sim\phi) := \mathsf{deg}_\sim(\phi) + 1, \mathsf{deg}_\sim(\mathrm{dep}(\overline{p}, q)) := \mathsf{deg}_\sim(p) := 0. \end{aligned}$$

Note that the relationship given by this translation is not strict. It is easy to show, by a brute-force algorithm, that TRUE(L) is in $\Sigma_{k+1}^{\mathrm{E}}$, where L is the fragment of QPTL(dep) with formulae with $\mathsf{deg}_\sim(\phi) \le k$. Moreover, from the above translation together with Theorem 3.11 we obtain hardness for $\Sigma_{k-2}^{\mathrm{E}}$.

Proposition 4.10. *Every* ADQBF*-instance* (DQBF*-instance*) $(\psi, \overline{C})$ *can be translated in polynomial time to a* QPTL(dep)*-sentence* (QPD*-sentence*) ϕ *s. t.* $(\psi, \overline{C})$ *is true iff* ϕ *is true.*

Using the ideas of [5], we may eliminate the quantifiers in (3) and relate the truth of ϕ and ϕ^* with the satisfiability of a certain formula of PTL extended with dependence atoms, denoted by PTL(dep). Define a shorthand $\max(p_1, \ldots, p_n) := \sim \bigvee_{1 \le i \le n} \mathrm{dep}(p_i)$. It was noted in [5] that X satisfies $\max(p_1, \ldots, p_n)$ if and only if for each assignment s with domain $\{p_1, \ldots, p_n\}$

[1] The syntax is the same as in Definition 3.4. However, for ADQBF, U refers to the universal quantification of Skolem functions, while in team semantics, it refers to the universal quantification of supplementing functions. These notions are not the same, but easily translatable into each other, as we show.

there is an expansion s' of s in X. Let $\phi' := \max(p_1,\dots,p_n,q_1^1,\dots,q_k^k) \wedge \psi$ denote the $\mathsf{PTL}(\mathrm{dep})$-formula, where ψ is obtained by using the following recursive translation to eliminate every quantifier from ϕ starting from left to right. Each quantifier of type $\exists q_j^i$ is recursively translated as $\Big(\mathrm{dep}\Big(\overline{c}_j^i, q_j^i\Big) \vee (\mathrm{dep}\Big(\overline{c}_j^i, q_j^i\Big) \wedge \psi)\Big)$. Each quantifier of type $\mathsf{U}\, q_j^i$ is recursively translated as $\sim\Big(\mathrm{dep}\Big(\overline{c}_j^i, q_j^i\Big) \vee (\mathrm{dep}\Big(\overline{c}_j^i, q_j^i\Big) \wedge \sim\psi)\Big)$. For the right most quantifier in the recursive translation, we set $\psi := \theta$. It is quite straightforward to prove (cf. [5, Theorem 7]) that ϕ is true under the constraint $(C_1^1,\dots,C_{j_k}^k)$ if and only if ϕ' is satisfiable. Here the connection between the classes Σ_k^{E} and Π_k^{E}, and the fragment of $\mathsf{PTL}(\mathrm{dep})$ of sentences with $\mathsf{deg}_{\sim}(\phi) \leq k$ is even more tighter than above. We obtain Σ_k^{E}-hardness for $\mathrm{SAT}(\mathsf{L})$, where L is the fragment of $\mathsf{PTL}(\mathrm{dep})$ with formulae with $\mathsf{deg}_{\sim}(\phi) \leq k$. Note that using the above recursive translation and by setting $\max(p_1,\dots,p_n,q_1^1,\dots,q_k^k) := \forall p_1 \cdots \forall p_n \forall q_1^1 \dots \forall q_k^k$, we obtain Σ_k^{E}-hardness for $\mathrm{TRUE}(\mathsf{L})$, where L is the fragment of $\mathsf{QPTL}(\mathrm{dep})$ with formulae with $\mathsf{deg}_{\sim}(\phi) \leq k$.

Finally note that dependence atoms of type $\mathrm{dep}(p_1,\dots,p_n,q)$ can be expressed via unary atoms as follows

$$\sim\big((r \vee \neg r) \vee \bigwedge_{1\leq n} \mathrm{dep}(p_i) \wedge \sim\mathrm{dep}(q)\big),$$

while unary atoms $\mathrm{dep}(p)$ can be rewritten as $\sim(\sim p \wedge \sim\neg p)$. As a summary, we obtain the following results.

Proposition 4.11. *Every* ADQBF*-instance* $(\psi, \overline{C})$ *can be translated in polynomial time to a* QPTL*-sentence* ϕ *s. t.* $(\psi, \overline{C})$ *is true iff* ϕ *is true.*

Proposition 4.12. *For a logic* L *let* L_k *denote the fragment of* L *with formulae* ϕ *for which* $\mathsf{deg}_{\sim}(\phi) \leq k$*. Then* $\mathrm{TRUE}(\mathsf{QPTL}_k(\mathrm{dep}))$ *and* $\mathrm{SAT}(\mathsf{PTL}_k(\mathrm{dep}))$ *are in* $\Sigma_{k+1}^{\mathrm{E}}$ *and* Σ_k^{E}*-hard w. r. t.* $\leq_{\mathrm{m}}^{\mathrm{P}}$*-reductions. Moreover* $\mathrm{TRUE}(\mathsf{QPTL}_k)$ *and* $\mathrm{SAT}(\mathsf{PTL}_k)$ *are in* $\Sigma_{k+1}^{\mathrm{E}}$ *and* $\Sigma_{k-2}^{\mathrm{E}}$*-hard w. r. t.* $\leq_{\mathrm{m}}^{\mathrm{P}}$*-reductions.*

5 Generalized dependence atoms

In this section we study extensions of QPL and QPTL by the so-called generalized dependence atoms. In the context of first-order dependence logic, generalized atoms were introduced by Kuusisto [9].

An *n-ary generalized dependence atom* (n-GDA) is a set G of n-ary relations over the Boolean domain $\{0,1\}$. For each n-GDA G, we introduce an atomic expression $A_G(p_1,\dots,p_n)$ that takes n proposition symbols as parameters. Let X be a team with $\{p_1,\dots,p_n\} \subseteq \mathsf{Dom}(X)$. The satisfaction relation $X \models A_G(p_1,\dots,p_n)$ is given as follows:

$$X \models A_G(p_1,\dots,p_n) \Leftrightarrow \mathrm{rel}(X,(p_1,\dots,p_n)) \in G,$$

where $\mathrm{rel}(X,(p_1,\dots,p_n)) := \{(s(p_1),\dots,s(p_n)) \mid s \in X\}$. We say that an SO_2-formula $\phi(f)$ with free function variable f *defines* G if

$$X \models A_G(p_1,\dots,p_n) \Leftrightarrow \emptyset^f_{\chi(X,(p_1,\dots,p_n))} \models \phi(f),$$

where $\chi(X,(p_1,\dots,p_n))$ is the characteristic function of $\mathrm{rel}(X,(p_1,\dots,p_n))$, and $\emptyset^f_{\chi(X,(p_1,\dots,p_n))}$ is the assignment that maps f to $\chi(X,(p_1,\dots,p_n))$. Moreover, we call G SO_2*-definable* (ESO_2*-definable*) if there exists an SO_2-formula (ESO_2-formula) $\phi(f)$ that defines G. For a set $\mathcal{G}$ of

GDAs, let us denote by $\mathsf{QPL}(\mathcal{G})$ ($\mathsf{QPTL}(\mathcal{G})$) the logic obtained by extending QPL (QPTL) with the atoms in $\mathcal{G}$. For a set $\mathcal{G} = \{G_i \mid i \in \mathbb{N}\}$ of atoms and respective defining sentences ϕ_i, the set $\mathcal{G}$ is said to be *polynomial time translatable* if the function $1^n \mapsto \langle \phi_n \rangle$, where $\langle \phi_n \rangle$ is the binary encoding of ϕ_n, is polynomial-time computable. The following theorem relates the logics $\mathsf{QPL}(\mathcal{G})$ and $\mathsf{QPTL}(\mathcal{G})$ to ESO_2 and SO_2, respectively.

As an example, we consider the dependence atom introduced in Section 4.

Example 5.1: The n-ary dependence atom $\mathrm{dep}(p_1,\ldots,p_{n-1},q)$ corresponds to the n-GDA that is defined as

$$\left\{ R \subseteq \{0,1\}^n \;\middle|\; \forall (s_1,\ldots,s_n),(t_1,\ldots,t_n) \in R : \bigwedge_{i=1}^{n-1} s_i = t_i \text{ implies } s_n = t_n \right\}$$

It is definable (even without second-order quantifiers) by the ESO_2-formula

$$\phi(f) := \forall x_1 \ldots \forall x_n \forall y_1 \ldots \forall y_n \left(f(x_1,\ldots,x_n) \wedge f(y_1,\ldots,y_n) \wedge \bigwedge_{i=1}^{n-1} x_i \leftrightarrow y_i \right) \rightarrow (x_n \leftrightarrow y_n)$$

Theorem 5.2. *Let $\mathcal{G}$ be a set of ESO_2-definable (SO_2-definable), polynomial time translatable generalized dependence atoms. Then every sentence in $\mathsf{QPL}(\mathcal{G})$ ($\mathsf{QPTL}(\mathcal{G})$) can be translated to an equivalent ESO_2 (SO_2) sentence in polynomial time.*

These translations are analogously presented by Väänänen in the first-order setting [15]. As we restrict ourselves to propositional logics, the difference is that only the domain $\{0,1\}$ and therefore the logic SO_2 are considered for the resulting formulae. The idea is to encode teams of assignments as their Boolean "characteristic functions". We start the proof with a slightly more general lemma.

Lemma 5.3. *Let $\mathcal{G}$ be a set of SO_2-definable, polynomial time translatable generalized dependence atoms. Then for every formula $\phi \in \mathsf{QPTL}(\mathcal{G})$ and every set of proposition symbols $\{p_1,\ldots,p_n\} \supseteq \mathsf{Fr}(\phi)$ there is an SO_2-sentence $\psi(f)$ computable in polynomial time s. t. for all Boolean teams X with $\mathsf{Dom}(X) \supseteq \{p_1,\ldots,p_n\}$,*

$$X \models \phi \Leftrightarrow \emptyset^f_{\chi(X,(p_1,\ldots,p_n))} \models \psi(f).$$

Proof. Analogously to the textbook translation from dependence logic to ESO [15], we show how to transform an open $\mathsf{QPTL}(\mathcal{G})$-formula ϕ, whose free proposition symbols are from $\{p_1,\ldots,p_n\}$, to an SO_2-sentence $\phi^*(f)$. One can easily verify by induction that for any team X with $\mathsf{Dom}(X) \supseteq \{p_1,\ldots,p_n\}$, $X \models \phi$ if and only if $\emptyset^f_{\chi(X,(p_1,\ldots,p_n))} \models \phi^*(f)$. The construction proceeds recursively as follows.

1. Assume $\phi = A_G(p_{i_1},\ldots,p_{i_k})$ for $G \in \mathcal{G}$, and let $\overline{p} = (p_1,\ldots,p_n)$, $\overline{p}_0 = (p_{i_1},\ldots,p_{i_k})$. Moreover, let $\overline{p}_1$ be any sequence listing $\{p_1,\ldots,p_n\} \setminus \{p_{i_1},\ldots,p_{i_k}\}$. Then $\phi^*(f)$ is defined as

$$\exists g\, \psi(g) \wedge \pi_{\overline{p},\overline{p}_0}(f,g),$$

where g has arity k, $\psi(g)$ is the SO_2-translation of G and $\pi_{\overline{p},\overline{p}_0}(f,g) := \forall \overline{p}(f(\overline{p}) \rightarrow g(\overline{p}_0)) \wedge \forall \overline{p}_0 \exists \overline{p}_1 (g(\overline{p}_0) \rightarrow f(\overline{p}))$ expresses that the team encoded in g is the projection of the team encoded in f onto the variables $\overline{p}_0$.

2. If $\phi = p_i$, then $\phi^*(f)$ is defined as $\forall\overline{p}(f(\overline{p}) \to p_i)$.
3. If $\phi = \neg p_i$, then $\phi^*(f)$ is defined as $\forall\overline{p}(f(\overline{p}) \to \neg p_i)$.
4. If $\phi = \psi_0 \wedge \psi_1$, then $\phi^*(f)$ is defined as $\psi_0^*(f) \wedge \psi_1^*(f)$.
5. If $\phi = \psi_0 \vee \psi_1$, then $\phi^*(f)$ is defined as

$$\exists f_0 \exists f_1 (\psi_0^*(f_0) \wedge \psi_1^*(f_1) \wedge \forall\overline{p}(f(\overline{p}) \to (f_0(\overline{p}) \vee f_1(\overline{p}))) \wedge \forall\overline{q}((f_0(\overline{q}) \vee f_1(\overline{q})) \to f(\overline{q}))).$$

6. If $\phi = \forall q\psi$, then $\phi^*(f)$ is defined as

$$\exists g(\psi^*(g) \wedge \forall\overline{p}\,\forall p'\,(f(\overline{p}) \to g(\overline{p},p')) \wedge \forall q \forall q'(g(\overline{q},q') \to f(\overline{q}))).$$

7. If $\phi = \exists q\psi$, then $\phi^*(f)$ is defined as

$$\exists g(\psi^*(g) \wedge \forall\overline{p}\,\exists p'(f(\overline{p}) \to g(\overline{p},p')) \wedge \forall\overline{q}\,\forall q'(g(\overline{q},q') \to f(\overline{q}))).$$

8. If $\phi = {\sim}\psi$, then $\phi^*(f)$ is defined as $\neg\psi^*(f)$. □

Note that, if the atoms in $\mathcal{G}$ are ESO_2-definable and if the $\sim$-case is dropped from the above translation, then the resulting formula itself is in ESO_2.

Corollary 5.4. *Let $\mathcal{G}$ be a set of ESO_2-definable, polynomial time translatable generalized dependence atoms. Then for every formula $\phi \in \mathsf{QPL}(\mathcal{G})$ and every set of proposition symbols $\{p_1,\ldots,p_n\} \supseteq \mathsf{Fr}(\phi)$ there is an ESO_2-sentence $\psi(f)$ computable in polynomial time s. t. for all Boolean teams X with $\mathsf{Dom}(X) \supseteq \{p_1,\ldots,p_n\}$,*

$$X \models \phi \Leftrightarrow \emptyset^f_{\chi(X,(p_1,\ldots,p_n))} \models \psi(f).$$

We are now ready to prove Theorem 5.2.

Proof of Theorem 5.2. Let ϕ be a $\mathsf{QPL}(\mathcal{G})$ resp. $\mathsf{QPTL}(\mathcal{G})$ sentence. First translate it to an ESO_2-sentence resp. SO_2-sentence $\psi(f)$ in polynomial time according to Corollary 5.4 resp. Lemma 5.3. It holds that $X \models \phi$ iff $\emptyset^f_{\chi(X,(p_1,\ldots,p_n))} \models \psi(f)$, if $\mathsf{Dom}(X) \supseteq \{p_1,\ldots,p_n\}$, and in particular that ϕ is true (i.e., satisfied by a non-empty team) iff $\psi' := \exists f \exists\overline{p}(f(\overline{p}) \wedge \psi(f))$ is true. □

Hence, we may conclude this section with the following complexity results.

Theorem 5.5. *(i) Assume that $\mathcal{G}$ is a polynomial time translatable set of SO_2-definable generalized dependence atoms. Then* TRUE($\mathsf{QPTL}(\mathcal{G})$) *is* **AEXPTIME**(poly)*-complete w. r. t. $\leq^{\mathrm{P}}_{\mathrm{m}}$-reductions.*

(ii) Assume that $\mathcal{G}$ is a polynomial time translatable set of ESO_2-definable generalized dependence atoms, and assume that dependence atoms translate into $\mathsf{QPL}(\mathcal{G})$ in polynomial time. Then TRUE($\mathsf{QPL}(\mathcal{G})$) *is* **NEXPTIME***-complete w. r. t. $\leq^{\mathrm{P}}_{\mathrm{m}}$-reductions.*

Proof. Both upper bounds follow from Theorem 5.2 and Proposition 2.3. The lower bound for TRUE($\mathsf{QPTL}(\mathcal{G})$) (TRUE($\mathsf{QPL}(\mathcal{G})$)) follows from Proposition 4.11 (Proposition 4.8). □

6 Summary

In this article we compared different approaches to function quantification, with the logics depicted in Figure 1. We showed that, while some of the logics can express the quantification of functions only in a restricted way, like only in form of Skolem functions, they all can be efficiently translated into each other. It was shown in Lemma 3.10 that the "uniqueness" property of function symbols occurring in SO_2 and ESO_2 formulae can be obtained and hence (as depicted in the proof of Theorem 3.11) these formulae have a natural translation into ADQBF and DQBF. Proposition 4.10 established that ADQBF and DQBF can easily be translated into team semantics, i.e., into $\mathsf{QPTL}(\mathrm{dep})$ and QPD. The point is that the dependence atom can be used in team semantics to model the constraints of Skolem functions. Finally we showed in Theorem 5.5 that propositional team logic, even when augmented with generalized dependence atoms, can efficiently be translated back into SO_2 resp. ESO_2 when teams are modeled as Boolean functions. Thus all these formalisms capture the same complexity classes: the class **AEXPTIME**(poly) by unbounded quantifier alternation and the class **NEXPTIME** by the existential fragment. Since $\mathsf{QPTL}(\mathcal{G})$ can express the dependence atom, it is complete for **AEXPTIME**(poly) for any set $\mathcal{G}$ of polynomial time translatable SO_2-definable generalized dependence atoms. For $\mathsf{QPL}(\mathcal{G})$ the matter is more complicated: If the dependence atom can be efficiently expressed in $\mathsf{QPL}(\mathcal{G})$ and $\mathcal{G}$ is a set of polynomial time translatable ESO_2-definable generalized dependence atoms, then $\mathsf{QPL}(\mathcal{G})$ is **NEXPTIME**-complete, but for instance for QPLInc the complexity drops down to **EXPTIME**, as shown in Theorem 4.9. Higher levels of the exponential hierarchy are not only captured by fragments of SO_2 (see Proposition 2.3), but also (with more or less sharp bounds) by the corresponding fragments of ADQBF (see Theorem 3.11) and (Q)PTL (Proposition 4.12).

Logic	Ex. fragment	Method of function quantification	Example
SO_2	ESO_2	Explicit, Second-order interpretation	$\exists f_y \forall \overline{x}\, \phi$
ADQBF	DQBF	Constraints, Skolem functions	$\forall \overline{x} \exists y\, \phi$, $C_y = \{x_1\}$
QPTL	QPD	Dep. atoms, Supplemented teams	$\forall \overline{x} \exists y\, \phi \wedge \mathrm{dep}(x_1, y)$
PTL	PD	Dep. atoms, Splitting of teams	$\mathrm{dep}(x_1, y) \vee (\phi \wedge \mathrm{dep}(x_1, y))$

Figure 1: Different formalisms of Boolean function quantification

Acknowledgements

We wish to thank the anonymous referees for their helpful suggestions. Miika Hannula was supported by the FRDF grant of the University of Auckland (project 3706751). Juha Kontinen and Jonni Virtema were supported by grant 292767 of the Academy of Finland.

References

[1] Ashok K. Chandra, Dexter Kozen & Larry J. Stockmeyer (1981): *Alternation. J. ACM* 28(1), pp. 114–133, doi:10.1145/322234.322243.

[2] Arnaud Durand, Juha Kontinen & Heribert Vollmer (2016): *Expressivity and Complexity of Dependence Logic.* Springer, In Press, doi:10.1007/978-3-319-31803-5_2.

[3] Pietro Galliani (2012): *Inclusion and exclusion dependencies in team semantics - On some logics of imperfect information*. *Ann. Pure Appl. Logic* 163(1), pp. 68–84, doi:10.1016/j.apal.2011.08.005.

[4] Pietro Galliani, Miika Hannula & Juha Kontinen (2013): *Hierarchies in independence logic*. In: *Computer Science Logic 2013 (CSL 2013)*, *Leibniz International Proceedings in Informatics (LIPIcs)* 23, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, pp. 263–280, doi:10.4230/LIPIcs.CSL.2013.263.

[5] Miika Hannula, Juha Kontinen, Jonni Virtema & Heribert Vollmer (2015): *Complexity of Propositional Independence and Inclusion Logic*. In: *Mathematical Foundations of Computer Science 2015 - 40th International Symposium, MFCS 2015, Milan, Italy, August 24-28, 2015, Proceedings, Part I*, pp. 269–280, doi:10.1007/978-3-662-48057-1_21.

[6] Lauri Hella, Antti Kuusisto, Arne Meier & Heribert Vollmer (2015): *Modal Inclusion Logic: Being Lax is Simpler than Being Strict*. In: *Mathematical Foundations of Computer Science 2015 - 40th International Symposium, MFCS 2015, Milan, Italy, August 24-28, 2015, Proceedings, Part I*, pp. 281–292, doi:10.1007/978-3-662-48057-1_22.

[7] Jaakko Hintikka & Gabriel Sandu (1989): *Informational Independence as a Semantical Phenomenon*. In Ivan T. Frolov Jens Erik Fenstad & Risto Hilpinen, editors: *Logic, Methodology and Philosophy of Science VIIIProceedings of the Eighth International Congress of Logic, Methodology and Philosophy of Science*, *Studies in Logic and the Foundations of Mathematics* 126, Elsevier, pp. 571 – 589, doi:10.1016/S0049-237X(08)70066-1.

[8] Juha Kontinen & Ville Nurmi (2011): *Team Logic and Second-Order Logic*. *Fundam. Inform.* 106(2-4), pp. 259–272, doi:10.3233/FI-2011-386.

[9] Antti Kuusisto (2015): *A Double Team Semantics for Generalized Quantifiers*. *Journal of Logic, Language and Information* 24(2), pp. 149–191, doi:10.1007/s10849-015-9217-4.

[10] Markus Lohrey (2012): *Model-checking hierarchical structures*. *Journal of Computer and System Sciences* 78(2), pp. 461–490, doi:10.1016/j.jcss.2011.05.006.

[11] Martin Lück (2016): *Complete Problems of Propositional Logic for the Exponential Hierarchy*. *CoRR* abs/1602.03050.

[12] Ville Nurmi (2009): *Dependence Logic: Investigations into Higher-Order Semantics Defined on Teams*. Ph.D. thesis, University of Helsinki.

[13] Gary L. Peterson, John H. Reif & Salman Azhar (2001): *Lower bounds for multiplayer noncooperative games of incomplete information*. *Computers & Mathematics with Applications* 41(7-8), pp. 957–992, doi:10.1016/S0898-1221(00)00333-3.

[14] Larry J. Stockmeyer & Albert R. Meyer (1973): *Word Problems Requiring Exponential Time: Preliminary Report*. In: *Proceedings of the 5th Annual ACM Symposium on Theory of Computing, April 30 - May 2, 1973, Austin, Texas, USA*, pp. 1–9, doi:10.1145/800125.804029.

[15] Jouko A. Väänänen (2007): *Dependence Logic - A New Approach to Independence Friendly Logic*. *London Mathematical Society student texts* 70, Cambridge University Press, doi:10.1017/CBO9780511611193.

[16] Jonni Virtema (2014): *Complexity of validity for propositional dependence logics*. In: *Proceedings Fifth International Symposium on Games, Automata, Logics and Formal Verification, GandALF 2014, Verona, Italy, September 10-12, 2014.*, pp. 18–31, doi:10.4204/EPTCS.161.5.

[17] Jouko Väänänen (2008): *Modal dependence logic*. *New Perspectives on Games and Interaction* 4, p. 237.