Propositional Fragments for Knowledge Compilation and Quantified Boolean Formulae

Sylvie Coste-Marquis and **Daniel Le Berre** and **Florian Letombe** and **Pierre Marquis**
CRIL/CNRS, Université d'Artois,
F-62307 Lens, France
{coste,leberre,letombe,marquis}@cril.univ-artois.fr

Abstract

Several propositional fragments have been considered so far as target languages for knowledge compilation and used for improving computational tasks from major AI areas (like inference, diagnosis and planning); among them are the (quite influential) ordered binary decision diagrams, prime implicates, prime implicants, "formulae" in decomposable negation normal form. On the other hand, the validity problem QBF for Quantified Boolean Formulae (QBF) has been acknowledged for the past few years as an important issue for AI, and many solvers have been designed for this purpose. In this paper, the complexity of restrictions of QBF obtained by imposing the matrix of the input QBF to belong to such propositional fragments is identified. Both tractability and intractability results (PSPACE-completeness) are obtained.

Introduction

Compiling "knowledge" has been used for the past few years to improve (from the computational point of view) basic tasks from major AI areas, like inference (both classical and nonmonotonic, see among others (Selman & Kautz 1996; del Val 1994; Schrag 1996; Boufkhad *et al.* 1997; Coste-Marquis & Marquis 2001; 2004; Darwiche & Marquis 2004)), diagnosis (see e.g. (Darwiche 1999)) and planning (see e.g. (Cimmati *et al.* 1997; Geffner 2004)). These approaches typically consist in turning, during an off-line phase, some pieces of information encoded as a propositional formula into a formula from a "more tractable" fragment. "More tractable" means that tasks required by the application under consideration becomes computationally easier, and if possible, feasible in polynomial time (Darwiche & Marquis 2002). Such tasks usually contain deciding satisfiability, the famous SAT problem (determining whether the formula has or not a model), which is NP-complete for propositional formulae. Among the " tractable" fragments considered so far are the (quite influential) ordered binary decision diagrams, prime implicates, prime implicants, " formulae" in decomposable negation normal form.

On the other hand, QBF, the validity problem for QBFs, has a growing importance in AI. This can be explained by the fact that, as the canonical PSPACE-complete problem, many AI problems can be polynomially reduced to QBF (see e.g., (Fargier, Lang, & Marquis 2000; Rintanen 1999a; Pan, Sattler, & Vardi 2002; Pan & Vardi 2003; Besnard *et al.* 2005)); in particular, it includes SAT as a specific case; furthermore, there is some empirical evidence from various AI fields (including among others planning, nonmonotonic reasoning, paraconsistent inference) that a translation-based approach can prove more "efficient" than domain-dependent algorithms dedicated to such AI tasks. Accordingly, many QBF solvers have been designed and evaluated for the past few years (see among others (Cadoli, Giovanardi, & Schaerf 1998; Rintanen 1999b; Feldmann, Monien, & Schamberger 2004; Giunchiglia, Narizzano, & Tacchella 2001; Letz 2002; Zhang & Malik 2002; Pan & Vardi 2004; Audemard & Saïs 2004)).

In this paper, we consider several tractable fragments for SAT, used as target languages for knowledge compilation. For each fragment $\mathcal{C}$ under consideration, we focus on the restriction of the QBF problem obtained by imposing the matrix of the input formula to belong to the fragment. A similar investigation has already been done w.r.t. some *incomplete* propositional fragments (Schaefer 1978; Creignou, Khanna, & Sudan 2001). Thus, in his well-known paper where a dichotomy theorem for SAT is presented (Schaefer 1978), Schaefer also gave an analogue dichotomy theorem for QBF (Theorem 6.1); roughly, this theorem shows that the only tractable classes for the restrictions of QBF among those "characterized locally" (i.e., by the nature of the "clauses" from the matrix) are the Krom one (binary clauses), the Horn one, the reverse Horn one and the affine one (sets of linear equations over the field $\{0,1\}$, or equivalently, conjunctions of XOR-clauses). Accordingly, several polytime algorithms for the restriction of QBF to such incomplete fragments can be found in the literature (see (Aspvall, Plass, & Tarjan 1979; Kleine-Büning, Karpinski, & Flögel 1995; Gent & Rowley 2002)).

In the following, the complexity of QBF is investigated for *complete* propositional fragments, where a propositional fragment $\mathcal{C}$ is complete if and only if every propositional formula has an equivalent from $\mathcal{C}$. We mainly focus on fragments considered in (Darwiche & Marquis 2002): DNF, d-DNNF, DNNF, $\text{OBDD}_<$, FBDD, PI, IP, MODS whose significance for many AI tasks (as well as for problems pertaining to other fields) is acknowledged. We complete the results given in (Darwiche & Marquis 2002) by focusing on

an additional query, the QBF one. We draw the complexity picture for QBF when restricted to those fragments.

Both tractability and intractability results have been derived. Like for the DNF fragment and its supersets including the DNNF fragment and the disjunctions of Horn CNF formulae, the QBF problem for the $\mathsf{OBDD}_<$ fragment (and its superset, the FBDD fragment and the d-DNNF fragment) is PSPACE-complete in the general case, while in P whenever the prefix of the instance is compatible with the total, strict ordering $<$ associated with the $\mathsf{OBDD}_<$ "formula"; we also show that the QBF problem for the MODS fragment is in P as well. We finally show that the QBF problem for prime implicates formulae and (resp. prime implicants formulae) is PSPACE-complete as well, while the complexity falls down to P when the prefix is of the form $\forall X\ \exists Y$ (resp. $\exists X\ \forall Y$).

Formal Preliminaries

Definition 1 (syntax of a quantified boolean formula) *Let PS be a finite set of propositional symbols. The set $QPROP_{PS}$ of* quantified boolean formulae *(QBFs) over PS is the smallest set of words defined inductively as follows:*[1]

1. *$true$, $false$ and every variable from PS belong to $QPROP_{PS}$.*
2. *if ϕ and ψ belong to $QPROP_{PS}$, then $\neg(\phi)$, $(\phi \wedge \psi)$, $(\phi \vee \psi)$, $(\phi \Rightarrow \psi)$, $(\phi \Leftrightarrow \psi)$, $(\phi \oplus \psi)$ belong to $QPROP_{PS}$.*
3. *if ϕ belongs to $QPROP_{PS}$ and x belongs to PS, then $\forall x(\phi)$ and $\exists x(\phi)$ belong to $QPROP_{PS}$.*

$Var(\Sigma)$ is the set of all symbols from PS occurring in a QBF Σ. A QBF Σ is said to be quantifier-free if and only if it does not contain any quantification (obviously enough, such formulae can easily be considered as "standard" propositional formulae). The subset of $QPROP_{PS}$ containing only quantifier-free formulae is noted $PROP_{PS}$.

Before defining the semantics of QBFs, we need the following notation. For every QBF Σ and every variable $x \in PS$, $\Sigma_{x \leftarrow 0}$ (resp. $\Sigma_{x \leftarrow 1}$) denotes the formula obtained by replacing every free occurrence of x in Σ by $false$ (resp. $true$).

Definition 2 (semantics of a quantified boolean formula) *Let I be an interpretation over PS (i.e., a total function from PS to $BOOL = \{0, 1\}$). The* semantics *of a QBF Σ in I is the truth value $[\![\Sigma]\!](I)$ from $BOOL$ defined inductively as for propositional formulae, except that the inductive definition contains in addition the two rules:*

- *if $\Sigma = \forall x(\phi)$,*
  then $[\![\Sigma]\!](I) = min(\{[\![\phi_{x \leftarrow 0}]\!](I), [\![\phi_{x \leftarrow 1}]\!](I)\})$.
- *if $\Sigma = \exists x(\phi)$,*
  then $[\![\Sigma]\!](I) = max(\{[\![\phi_{x \leftarrow 0}]\!](I), [\![\phi_{x \leftarrow 1}]\!](I)\})$.

An interpretation I is a said to be a *model* of Σ, noted $I \models \Sigma$, if and only if $[\![\Sigma]\!](I) = 1$. If Σ has a model, it is *satisfiable*; otherwise, it is *unsatisfiable*. If every interpretation I over PS is a model of Σ, Σ is *valid*. If every model of a QBF Σ is a model of a QBF μ, then μ is a *logical consequence* of Σ, noted $\Sigma \models \mu$. Finally, when both $\Sigma \models \mu$ and $\mu \models \Sigma$ hold, Σ and μ are *equivalent*, noted $\Sigma \equiv \mu$. Because every connective is truth-functional and truth is captured by a unique truth value (1), a replacement theorem holds for QBFs: if a QBF ϕ is equivalent to a QBF ψ and ϕ is a subformula of a QBF Σ, replacing occurrences of ϕ by ψ in Σ leads to a QBF equivalent to Σ.

The QBF problem is concerned with the validity of prenex, closed and polite formulae from $QPROP_{PS}$; formally, the language of its instances is the set of all QBFs Σ of the form $Qx_1 \ldots Qx_n\phi$ where each occurrence of Q stands for a quantifier $\forall$ or $\exists$, ϕ is a quantifier-free QBF and $Var(\phi) = \{x_1, \ldots, x_n\}$. The sequence $Qx_1 \ldots Qx_n\phi$ is called the *prefix* of Σ, while ϕ is its *matrix*. The set of positive instances of QBF contains the set of valid prenex, closed and polite QBFs.

Tractable vs. Intractable Classes for QBF

In the following, the complexity of several restrictions of QBF is investigated. A propositional fragment is said to be tractable for QBF if and only if the membership to the fragment can be decided in polynomial time, and there also exists a polytime decision algorithm for the validity problem of (closed, polite, prenex) quantified boolean formulae whose matrix is from the fragment.

Let us start with intractability results. First, it is well-known that the restriction of QBF obtained by imposing the matrix to be a CNF formula is still PSPACE-complete. Indeed, every propositional formula Σ over $\{x_1, \ldots, x_n\}$ can be associated in linear time to a CNF formula Σ' over $\{x_1, \ldots, x_n, y_1, \ldots, y_m\}$ s.t. $\Sigma \equiv \exists\{y_1, \ldots, y_m\}\Sigma'$.[2] Such a reduction which preserves satisfiability (and much more) is typically used to show that CIRCUIT-SAT can be reduced to SAT restricted to CNF formulae (the idea is to introduce a new variable y_i per gate).

Let us now consider the restriction of QBF when matrices belong to a target fragment for knowledge compilation; many such fragments have been identified in the literature: DNF, d-DNNF, DNNF, FBDD, $\mathsf{OBDD}_<$, MODS, PI, IP, ... As we will see, QBF remains typically intractable under such restrictions.

First, since PSPACE is closed under complementation and the negation of a QBF with a CNF matrix is a QBF with a DNF matrix, it follows directly that the restriction of QBF where the matrix is a DNF formula also is PSPACE-complete. This prevents many tractable fragments for SAT to be considered as interesting candidates for QBF. Among them are all the supersets of DNF including the DNNF fragment and the disjunctions of Horn CNF formulae which are

[1] In order to simplify the syntax, we feel free to omit some parentheses when this does not affect equivalence.

[2] Since it is possible to switch two successive quantifications of the same nature in a QBF without affecting equivalence, for every finite, non empty subset $S = \{x_1, \ldots, x_n\}$ of PS, we note $\forall S(\phi)$ (resp. $\exists S(\phi)$) as a short for $\forall x_1(\ldots \forall x_n(\phi)\ldots)$ (resp. $\exists x_1(\ldots \exists x_n(\phi)\ldots)$).

target classes for knowledge compilation (see (Schrag 1996; Boufkhad *et al.* 1997; Darwiche & Marquis 2002)).

Let us now turn to complete DAG-based propositional fragments. Abusing words, a "formula" in NNF_{PS} is a rooted, directed acyclic graph where each leaf node is labeled with $true$, $false$, x or $\neg x$, $x \in PS$; and each internal node is labeled with $\wedge$ or $\vee$ and can have arbitrarily many children. If C is a node in an NNF_{PS} formula, then $Var(C)$ denotes the set of all variables that label the descendants of node C. Moreover, if ϕ is an NNF_{PS} formula rooted at C, then $Var(\phi)$ is defined as $Var(C)$. Interesting fragments of NNF_{PS} are obtained by imposing some of the following requirements (Darwiche 2001):

- **Decomposability**: An NNF_{PS} formula satisfies this property if for each conjunction C in the formula, the conjuncts of C do not share variables. That is, if $C_1, \ldots, C_n$ are the children of and-node C, then $Var(C_i) \cap Var(C_j) = \emptyset$ for $i \neq j$.
- **Determinism**: An NNF_{PS} formula satisfies this property if for each disjunction C in the formula, each two disjuncts of C are logically contradictory. That is, if $C_1, \ldots, C_n$ are the children of or-node C, then $C_i \wedge C_j \models false$ for $i \neq j$.
- **Decision**: A decision node N in an NNF_{PS} formula is one which is labeled with $true$, $false$, or is an or-node having the form $(x \wedge \alpha) \vee (\neg x \wedge \beta)$, where x is a variable, α and β are decision nodes. In the latter case, $dVar(N)$ denotes the variable x.
- **Ordering**: Let $<$ be a total, strict ordering over the variables PS. A NNF_{PS} formula satisfies the ordering property w.r.t. $<$ if and only if the following condition is satisfied: if N and M are or-nodes, and if N is an ancestor of node M, then $dVar(N) < dVar(M)$.
- **Smoothness**: An NNF formula satisfies this property if for each disjunction C in the formula, each disjunct of C mentions the same variables. That is, if $C_1, \ldots, C_n$ are the children of or-node C, then $Var(C_i) = Var(C_j)$ for $i \neq j$.

We consider the following propositional fragments[3] (Darwiche & Marquis 2002):

Definition 3 (propositional fragments)

- *The language* DNNF *is the subset of* NNF_{PS} *of formulae satisfying decomposability.*
- *The language* d-DNNF *is the subset of* NNF_{PS} *of formulae satisfying decomposability and determinism.*
- *The language* FBDD *is the subset of* NNF_{PS} *of formulae satisfying decomposability and decision.*
- *The language* $\mathsf{OBDD}_<$ *is the subset of* NNF_{PS} *of formulae satisfying decomposability, decision and ordering.*
- *The language* MODS *is the subset of* DNF $\cap$ d-DNNF *of formulae satisfying smoothness.*

The FBDD language corresponds to *free binary decision diagrams (FBDDs)*, as known in formal verification (Gergov & Meinel 1994), while its subset obtained by imposing the ordering property w.r.t. a given variable ordering contains the *ordered binary decision diagrams (OBDDs)* (Bryant 1986).

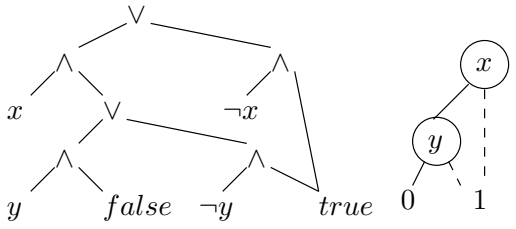
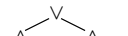


Figure 1: On the left, a formula in the $\mathsf{OBDD}_<$ language. On the right, a more standard notation for it.

Binary decision diagrams are usually depicted using a more compact notation: labels $true$ and $false$ are denoted by 1 and 0, respectively; and each decision node  denoted by  The $\mathsf{OBDD}_<$ formula on the left of Figure 1 corresponds to the binary decision diagram on the right of Figure 1.

The MODS encoding of a propositional formula mainly consists in the explicit representation of the set of its models.

Eliminating a single quantification within an $\mathsf{OBDD}_<$ formula can be achieved in time quadratic in the input size (an $\mathsf{OBDD}_<$ formula equivalent to $\exists x.\Sigma$ (resp. $\forall x.\Sigma$) is computed as $\Sigma_{x\leftarrow 0} \vee \Sigma_{x\leftarrow 1}$, (resp. $\Sigma_{x\leftarrow 0} \wedge \Sigma_{x\leftarrow 1}$), see e.g. (Bryant 1986)). Since the size of the resulting formula may be quadratic in the size of the input Σ, there is no guarantee that such an elimination process leads to a formula of size polynomial in the input size when iterated so as to eliminate more than a preset number of variables. Hence, there is no guarantee that the time needed by such an elimination algorithm will remain polynomial in the input size. Actually, the next proposition shows that whatever the approach to solving QBF for $\mathsf{OBDD}_<$ formulae, a polytime algorithm is very unlikely:

Proposition 1 *The* QBF *problems for* DNNF, d-DNNF, FBDD *and* $\mathsf{OBDD}_<$ *formulae are* PSPACE*-complete.*

Sketch of proof: Membership directly comes from the fact that QBF for CNF formulae is in PSPACE, the fact that the circuit language associated to $PROP_{PS}$ includes NNF_{PS} as a proper subset, the fact that every circuit (encoding a boolean function over $\{x_1, \ldots, x_n\}$) can be mapped in polynomial time to a CNF formula over an extended set of vari-

[3] It must be noted that the five languages below are not *stricto sensu* subsets of $PROP_{PS}$ in the sense that its elements are rooted DAGs, not standard tree-like formulae. Considering DAG-based representations is just a way to enable subformulae sharing; while this is important for the spatial efficiency point of view, this has no impact on the semantical issue, so the definitions and properties reported in the previous Section can be easily extended to DAG-based formulae.

ables, whilst equivalent to the circuit whenever the new variables are forgotten (i.e., existentially quantified), and the fact that PSPACE is closed under polynomial reductions.

As to hardness, since the following inclusions hold

$$\mathtt{OBDD}_< \subset \mathtt{FBDD} \subset \mathtt{d\text{-}DNNF} \subset \mathtt{DNNF}$$

it is sufficient to prove that the QBF problem for $\mathtt{OBDD}_<$ formulae is PSPACE-hard. The proof is by reduction from the QBF problem for DNF formulae. The main step is to show that every DNF formula $\phi = \gamma_1 \vee \ldots \gamma_n$ can be associated in polynomial time to an equivalent QBF of the form $\exists X \psi$ where $X \cap Var(\phi) = \emptyset$ and ψ is from $\mathtt{OBDD}_<$ (whatever $<$ over $Var(\phi)$). First, let us note $obdd(\gamma_i)$ the $\mathtt{OBDD}_<$ formula equivalent to the term γ_i ($i \in 1 \ldots n$); clearly enough, every $obdd(\gamma_i)$ can be computed in time polynomial in $|\gamma_i|$. Let $Var(\phi) = \{y_1, \ldots, y_m\}$ and let $X = \{x_1, \ldots, x_{n-1}\}$ be a set of new variables; let $\psi = \psi^1$, where the formulae ψ^i ($i \in 1 \ldots n$) are defined by:

- $\psi^n = obdd(\gamma_n)$, and
- $\psi^i = (obdd(\gamma_i) \wedge x_i) \vee (\psi^{i+1} \wedge \neg x_i)$, for $i = 1, \ldots, n-1$.

From such definitions, ψ – which can be read as an $\mathtt{OBDD}_<$ formula where the new ordering $<$ is the extension of the previous ordering $y_1 < \ldots < y_m$ such that $x_1 < \ldots < x_{n-1} < y_1 < \ldots < y_m$ – can be computed in time polynomial in the size of ϕ. Now, since for every pair of propositional formulae α, β and every variable x, we have that $\exists x(\alpha \vee \beta) \equiv (\exists x \alpha) \vee (\exists x \beta)$, and $\exists x(\alpha \wedge x) \equiv \exists x(\alpha \wedge \neg x) \equiv \alpha$ whenever $x \notin Var(\alpha)$, it immediately follows that $\phi \equiv \exists Y \psi$. Finally, the replacement theorem for QBFs shows that for any prefix P, the QBF $P\phi$ is equivalent to the QBF $P\ \exists X \psi$, and this completes the proof. ■

Based on this reduction, one can also show that QBF for $\mathtt{OBDD}_<$ formulae spans all the polynomial hierarchy when restrictions are put on the prefix of the input: if no alternations of quantifiers occur, the problem reduces to the satisfiability problem or to the validity problem, and both of them are in P for $\mathtt{OBDD}_<$ formulae; if the prefix is of the form $\forall S_1\ \exists S_2$, the problem is Π_1^p-complete (= coNP-complete); if the prefix is of the form $\exists S_1\ \forall S_2\ \exists S_3$, the problem is Σ_2^p-complete, and so on. Since the negation of an $\mathtt{OBDD}_<$ formula can be computed as an $\mathtt{OBDD}_<$ formula in constant time, we also obtain that if the prefix is of the form $\exists S_1\ \forall S_2$, the problem is Σ_1^p-complete (= NP-complete), if the prefix is of the form $\forall S_1\ \exists S_2\ \forall S_3$, the problem is Π_2^p-complete, and so on.

Nevertheless, it is interesting to note that the restriction of QBF to $\mathtt{OBDD}_<$ formulae is tractable for the subset of instances whose prefixes are compatible with the total, strict ordering $<$ associated with the $\mathtt{OBDD}_<$ fragment:

Proposition 2 *Let $\Sigma = QS_1 \ldots QS_n.\phi$ be a prenex, polite, closed QBF where each Q stands for a quantifier and $\{S_1, \ldots, S_n\}$ is a partition of $Var(\phi)$ which does not contain the empty set. The prefix $QS_1 \ldots QS_n$ of Σ is said to be* compatible *with a total, strict ordering $<$ over $Var(\phi)$ if and only if for each $x, y \in Var(\phi)$ s.t. $x < y$, if $x \in S_i$, then $y \in S_j$ with $j \geq i$. The* QBF *problem for $\mathtt{OBDD}_<$ formulae when the prefix is compatible with $<$ is in* P.

Sketch of proof: The proof relies on the correctness of a polytime algorithm for eliminating quantifiers as an internal law for such $\mathtt{OBDD}_<$ "formulae", so it is sufficient to apply it from the most internal quantifier to the most external one until all quantifiers have been eliminated; at each step the size of the resulting $\mathtt{OBDD}_<$ formula once a quantifier has been eliminated is bounded by the size of the input $\mathtt{OBDD}_<$ formula. ■

The picture is similar when the MODS fragment is considered:

Proposition 3 *The* QBF *problem for* MODS *formulae is in* P.

Again, we have a polytime algorithm for eliminating quantifiers as an internal law for MODS formulae.

Let us now turn to two additional, important propositional fragments in AI: the prime implicates one and the (dual) prime implicants one (see e.g., (Marquis 2000) for a survey of their applications in abduction, assumption-based reasoning, closed world reasoning and other AI areas). Formally, a *prime implicate* of $\phi \in PROP_{PS}$ is a clause δ s.t. $\phi \models \delta$ and for every clause δ' s.t. $\phi \models \delta'$ and $\delta' \models \delta$, we have $\delta \equiv \delta'$.

Definition 4 (prime implicates formulae) *A prime implicates formula (or Blake formula) from $PROP_{PS}$ is a CNF formula Σ where every prime implicate of Σ appears as a conjunct.* PI *is the language of all prime implicates formulae (a proper subset of* CNF*).*

For instance, the following is a prime implicates formula:

$$(a \vee b) \wedge (\neg b \vee c \vee d) \wedge (a \vee c \vee d).$$

The set of prime implicates formulae is a tractable fragment for SAT since (1) a CNF formula Σ is a prime implicate one if and only no clause of it is properly entailed by another clause of Σ and every resolvent from two clauses from Σ is entailed by a clause of Σ (this shows that the problem of deciding whether a propositional formula is a prime implicate one can be decided in polynomial time), and (2) a prime implicate formula Σ is satisfiable if and only if it does not reduce to the empty clause.

Proposition 4 *The* QBF *problem for prime implicates formulae is* PSPACE*-complete.*

Sketch of proof: Membership comes directly from the fact that QBF is in PSPACE for CNF formulae, and every PI formula also is CNF. As to hardness, let us give a polytime reduction from QBF for CNF formulae $QS_1 \ldots QS_k\phi$ to QBF for PI formulae. Let ϕ be a CNF formula over $\{x_1, \ldots, x_n\}$, viewed as the set S of its clauses. We take advantage of the following property, which results directly from the correctness of resolution-based prime implicates algorithms (like Tison's one (Tison 1967)): a set S of clauses contains all its prime implicates if and only if

whenever two clauses from S have a resolvent δ, there exists a clause $\mathtt{PI} \in S$ s.t. $\mathtt{PI} \models \delta$. Let δ_i and δ_j be two clauses from S with $i < j$; when δ_i and δ_j have a resolvent, replace δ_1 by $\delta_1 \vee y_{i,j}$ and δ_2 by $\delta_2 \vee \neg y_{i,j}$; doing it in a systematic way for every ordered pair of clauses from S leads to generate in polynomial time a CNF formula ψ over an extended vocabulary $\{x_1, \ldots, x_n\} \cup Y$ where $\mathcal{O}(n^2)$ new variables $y_{i,j}$ are introduced. By construction, every binary resolvent from clauses of ψ is tautologous, hence implied by any clause of ψ. As a consequence, ψ contains all its prime implicates, and a prime implicates formula equivalent to ψ can be computed in time polynomial in $|\psi|$, just by removing every clause of ψ which is properly implied. Now, for every pair of formulae α and β and every variable $x \in PS$, we have $\forall x(\alpha \wedge \beta) \equiv (\forall x\alpha) \wedge (\forall x\beta)$; furthermore, for every nontautologous clause δ (viewed as the set of its literals) and every variable $x \in PS$, $\forall x\delta$ is equivalent to the clause $\delta \setminus \{x, \neg x\}$. As a consequence, we have $\phi \equiv \forall Y \psi$. Finally, the replacement theorem for QBFs shows that $QS_1 \ldots QS_k\phi$ is equivalent to $QS_1 \ldots QS_k \forall Y \psi$, and this concludes the proof. ■

Based on this reduction, one can also show that QBF for PI formulae hits every level from the polynomial hierarchy when restrictions are put on the prefix: if no alternations of quantifiers occur, the problem is in P, if the prefix is of the form $\exists S_1\ \forall S_2$, the problem is Σ_1^p-complete (= NP-complete), if the prefix is of the form $\forall S_1\ \exists S_2\ \forall S_3$, the problem is Π_2^p-complete, and so on.

Now, what's about QBF for PI formulae when the rightmost quantification of the prefix is existential? Contrariwise to $\mathtt{OBDD}_<$ formulae, the negation of PI formula cannot be computed in polynomial time (and even in *polynomial space*) as a PI formula, hence the same argument cannot be used again. It is easy to show that a rightmost existential quantification does not lead to a complexity shift:

Proposition 5 *The* QBF *problem for prime implicates formulae with prefixes of the form $\forall S_1\ \exists S_2$ is in* P.

Sketch of proof: The proof comes from the fact that variable forgetting (i.e., eliminating existential quantifiers) can be achieved in polynomial time as an internal law in the PI fragment (Darwiche & Marquis 2002). ■

From the previous reduction, we obtain that if the prefix is of the form $\exists S_1\ \forall S_2\ \exists S_3$, the QBF problem for PI formulae is Σ_1^p-complete, if the prefix is of the form $\forall S_1\ \exists S_2\ \forall S_3\ \exists S_4$, the problem is Π_2^p-complete, and so on.

The following dual class also is interesting. Let $\phi \in PROP_{PS}$. A *prime implicant* of ϕ is a term γ s.t. $\gamma \models \phi$ and for every term γ' s.t. $\gamma' \models \phi$ and $\gamma \models \gamma'$, we have $\gamma \equiv \gamma'$.

Definition 5 (prime implicants formulae) *A prime implicants formula from $PROP_{PS}$ is a DNF formula Σ where every prime implicant of Σ appears as a disjunct.* IP *is the language of all prime implicants formulae (a proper subset of* DNF*).*

Prime implicants formulae are duals of prime implicates formulae in the sense that every prime implicant of a formula ϕ is (up to logical equivalence) the negation of a prime implicate of $\neg\phi$. Taking advantage of duality, we also obtain that:

Proposition 6 *The* QBF *problem for prime implicants formulae is* PSPACE*-complete.*

Proposition 7 *The* QBF *problem for prime implicants formulae with prefixes of the form $\exists S_1\ \forall S_2$ is in* P.

Exploiting duality, it is easy to show that the classes Σ_i^p and Π_i^p of the polynomial hierarchy that were not "hit" by restrictions of QBF for PI formulae are "hit" by restrictions of QBF for IP formulae: if no alternations of quantifiers occur, the problem QBF for IP formulae is in P, if the prefix is of the form $\forall S_1\ \exists S_2$ or $\forall S_1\ \exists S_2\ \forall S_3$, the problem is Π_1^p-complete, if the prefix is of the form $\exists S_1\ \forall S_2\ \exists S_3$ or $\exists S_1\ \forall S_2\ \exists S_3\ \forall S_4$, the problem is Σ_2^p-complete, and so on.

Conclusion

In this paper, we have presented new tractability and new intractability results for the validity problems for QBFs whose matrices belong to a target class for knowledge compilation. In the light of our study, the complexity landscape for QBF can be completed as reported on Table 1.

Fragment	Complexity
$PROP_{PS}$ (general case)	PSPACE-c
CNF	PSPACE-c
DNF	PSPACE-c
d-DNNF	PSPACE-c
DNNF	PSPACE-c
FBDD	PSPACE-c
$\mathtt{OBDD}_<$	PSPACE-c
$\mathtt{OBDD}_<$ (compatible prefix)	$\in$ P
PI	PSPACE-c
IP	PSPACE-c
MODS	$\in$ P

Table 1: Complexity results for QBF.

In (Darwiche & Marquis 2002), the authors have also investigated the spatial efficiency of many complete propositional fragments, including those considered in this paper. A given fragment $\mathcal{C}_1$ is considered at least as concise than a second fragment $\mathcal{C}_2$ whenever there exists a polynomial $p(.)$ s.t. for every formula $\alpha \in \mathcal{C}_2$, there exists an equivalent formula $\beta \in \mathcal{C}_1$ s.t. $|\beta| \leq p(|\alpha|)$. Our results show QBF difficult even when limited to instances whose matrices belong to fragments which are not efficient from the spatial point of view (i.e., the $\mathtt{OBDD}_<$ one, the PI fragment and the IP fragment). Tractability is achieved without restrictions only for the MODS fragment which is among the least efficient one (as to spatial efficiency). Under the compatibility assumption, tractability is also achieved for the more concise $\mathtt{OBDD}_<$ fragment; this fragment appears as the best

candidate among the classes considered in this paper which enable tractable QBF queries.

Acknowledgements

The authors have been partly supported by the Région Nord/Pas-de-Calais through the IRCICA Consortium and the COCOA Project, by the European Community FEDER Program and by the IUT de Lens. Many thanks to the anonymous reviewers for their helpful comments.

References

Aspvall, B.; Plass, M.; and Tarjan, R. 1979. A linear-time algorithm for testing the truth of certain quantified boolean formulas. *Information Processing Letters* 8:121–123. Erratum: Information Processing Letters 14(4): 195 (1982).

Audemard, G., and Saïs, L. 2004. SAT based BDD solver for Quantified Boolean Formulas. In *ICTAI'04*, 82–89.

Besnard, P.; Schaub, T.; Tompits, H.; and Woltran, S. 2005. *Inconsistency Tolerance*, volume 3300 of *LNCS State-of-the-Art Survey*. Springer-Verlag. chapter Representing Paraconsistent Reasoning via Quantified Propositional Logic, 84–118.

Boufkhad, Y.; Grégoire, E.; Marquis, P.; Mazure, B.; and Saïs, L. 1997. Tractable cover compilations. In *IJCAI'97*, 122–127.

Bryant, R. E. 1986. Graph-based algorithms for boolean function manipulation. *IEEE Trans. on Computers* C-35(8):677–692.

Cadoli, M.; Giovanardi, A.; and Schaerf, M. 1998. An algorithm to Evaluate Quantified Boolean Formulae. In *AAAI'98*, 262–267.

Cimmati, A.; Giunchiglia, E.; Giunchiglia, F.; and Traverso, P. 1997. Planning via model checking: a decision procedure for AR. In *ECP'97*, 130–142.

Coste-Marquis, S., and Marquis, P. 2001. Knowledge compilation for circumscription and closed world reasoning. *J. of Logic and Computation* 11(4):579–607.

Coste-Marquis, S., and Marquis, P. 2004. On Stratified Belief Base Compilation. *Annals of Mathematics and Artificial Intelligence* 42(4):399–442.

Creignou, N.; Khanna, S.; and Sudan, M. 2001. Complexity classification of boolean constraint satisfaction problems. In *SIAM Monographs on Discrete Mathematics and Applications*, volume 7. SIAM.

Darwiche, A., and Marquis, P. 2002. A Knowledge Compilation Map. *J. of Artificial Intelligence Research* 17:229–264.

Darwiche, A., and Marquis, P. 2004. Compiling propositional weighted bases. *Artificial Intelligence* 157(1–2):81–113.

Darwiche, A. 1999. Compiling devices into decomposable negation normal form. In *IJCAI'99*, 284–289.

Darwiche, A. 2001. Decomposable negation normal form. *J. of the ACM* 48(4):608–647.

del Val, A. 1994. Tractable databases: how to make propositional unit resolution complete through compilation. In *KR'94*, 551–561.

Fargier, H.; Lang, J.; and Marquis, P. 2000. Propositional Logic and One-stage Decision Making. In *KR'00*, 445–456.

Feldmann, R.; Monien, B.; and Schamberger, S. 2004. A distributed algorithm to evaluate quantified boolean formulas. In *AAAI'00*, 285–290.

Geffner, H. 2004. Planning graphs and knowledge compilation. In *KR'04*, 662–672.

Gent, I. P., and Rowley, A. G. D. 2002. Solving 2-CNF Quantified Boolean Formulae using Variable Assignment and Propagation. In *QBF wks at SAT'02*, 17–25.

Gergov, J., and Meinel, C. 1994. Efficient analysis and manipulation of OBDDs can be extended to FBDDs. *IEEE Trans. on Computers* 43(10):1197–1209.

Giunchiglia, E.; Narizzano, M.; and Tacchella, A. 2001. Backjumping for Quantified Boolean Logic Satisfiability. In *IJCAI'01*, 275–281.

Kleine-Büning, H.; Karpinski, M.; and Flögel, A. 1995. Resolution for quantified boolean formulas. *Information and Computation* 117(1):12–18.

Letz, R. 2002. Lemma and Model Caching in Decision Procedures for Quantified Boolean Formulas. In *Tableaux'02*, 160–175.

Marquis, P. 2000. Consequence finding algorithms. In Gabbay, D., and Smets, P., eds., *Algorithms for Uncertain and Defeasible Reasoning*, volume 5 of *Handbook of Defeasible Reasoning and Uncertainty Management Systems*. Kluwer Academic Publishers. 41–145.

Pan, G., and Vardi, M. 2003. Optimizing a BDD-Based Modal Solver. In *CADE'02*, 75–89.

Pan, G., and Vardi, M. 2004. Symbolic Decision Procedures for QBF. In *CP'04*, 453–467.

Pan, G.; Sattler, U.; and Vardi, M. 2002. BDD-Based Decision Procedures for K. In *CADE'02*, 16–30.

Rintanen, J. 1999a. Constructing Conditional Plans by a Theorem-Prover. *J. of Artificial Intelligence Research* 10:323–352.

Rintanen, J. 1999b. Improvements to the Evaluation of Quantified Boolean Formulae. In *IJCAI'99*, 1192–1197.

Schaefer, T. J. 1978. The complexity of satisfiability problems. In *STOC'78*, 216–226.

Schrag, R. 1996. Compilation for critically constrained knowledge bases. In *AAAI'96*, 510–515.

Selman, B., and Kautz, H. 1996. Knowledge compilation and theory approximation. *J. of the ACM* 43:193–224.

Tison, P. 1967. Generalization of consensus theory and application to the minimization of boolean functions. *IEEE Trans. on Electronic Computers* EC–16:446–456.

Zhang, L., and Malik, S. 2002. Towards a symmetric treatment of satisfaction and conflicts in quantified boolean formula evaluation. In *CP'02*, 200–215.