International Journal on Artificial Intelligence Tools

Enhanced Unsatisfiable Cores for QBF: Weakening Universal to Existential Quantifiers

Viktor Schuppan
Email: Viktor.Schuppan@gmx.de, *URL:* http://schuppan.de/viktor/

Received (Day Month Year)
Revised (Day Month Year)
Accepted (Day Month Year)

We introduce an enhanced notion of unsatisfiable cores for QBF in prenex CNF that allows to weaken universal quantifiers to existential quantifiers in addition to the traditional removal of clauses. The resulting unsatisfiable cores can be different from those of the traditional notion in terms of syntax, standard semantics, and proof-based semantics. This not only gives rise to explanations of unsatisfiability but, via duality, also leads to diagnoses and repairs of unsatisfiability that are not obtained with traditional unsatisfiable cores. We use a source-to-source transformation on QBF in PCNF such that the weakening of universal quantifiers to existential quantifiers in the original formula corresponds to the removal of clauses in the transformed formula. This makes any tool or method for the computation of unsatisfiable cores of the traditional notion available for the computation of unsatisfiable cores of our enhanced notion. We implement our approach as an extension to the QBF solver `DepQBF`, and we perform an extensive experimental evaluation on a subset of `QBFLIB`. We illustrate with several case studies that helpful information can be provided by unsatisfiable cores of our enhanced notion.

Keywords: QBF; unsatisfiable cores; quantifier weakening.

1. Introduction

1.1. *Motivation and contributions*

Many important problems have natural encodings as QBF (quantified Boolean formulas). Examples include two-player games,[1] variants of planning,[2] satisfiability of formulas in the modal logic K,[3] and a number of problems in knowledge representation[4] and formal methods;[5] for a more extensive list see Ref. 6. Unsatisfiable cores have been established as a fundamental concept in logic with significant applications in AI and formal methods. For example, in various logics unsatisfiable cores are used to represent causes of unsatisfiability and to serve as explanations of unsatisfiability,[7–11] as building blocks to obtain advanced explanations of unsatisfiability,[12] to diagnose unsatisfiability,[13] and to repair unsatisfiability.[14] Previous work on unsatisfiable cores for QBF in PCNF (prenex conjunctive normal form) removes clauses to weaken formulas.[10,15–17]

In this paper we present an enhanced notion of unsatisfiable cores for QBF in

PCNF that not only removes clauses from a QBF but also weakens universal quantifiers to existential quantifiers (Section 3). We show that our enhanced notion of unsatisfiable cores can represent causes and lead to explanations of unsatisfiability that differ in terms of syntax as well as both standard and proof-based semantics from any cause and explanation that can be obtained from an unsatisfiable core of the traditional notion (Section 4). Moreover, via duality this gives rise to diagnoses and repairs for unsatisfiability that are different from those obtained when using the traditional notion of unsatisfiable cores (Section 7). On a more practical rather than rigorously formal note, if a user finds that the set of quantifiers that has been weakened from universal to existential in an unsatisfiable core exhibits some unexpected characteristics, then this may provide her with the initial hunch that there might be something off in the QBF under consideration. In Section 5 we prove that if it is not possible to remove any clause from an unsatisfiable QBF in PCNF without making the result satisfiable, then it is also not possible to weaken any universal quantifier to an existential quantifier without losing unsatisfiability. We then show that the PSPACE-completeness result for minimally unsatisfiable cores of the traditional notion[15] can be extended to our enhanced notion (Section 6). In Sections 8 and 9 we present a transformation of QBF in PCNF such that weakening of universal quantifiers to existential quantifiers in the original formula can be achieved by removing clauses in the transformed formula. Using this transformation we can obtain unsatisfiable cores of our enhanced notion in three steps: (i) apply the transformation; (ii) use existing tools and methods to compute an unsatisfiable core by removing clauses; and (iii) map back the result to an unsatisfiable core of the enhanced notion. In Section 10 we provide hints that can help to interpret unsatisfiable cores of our enhanced notion, and in Section 11 we classify universal quantifications into non-trivially, trivially, and not ∀-to-∃ reducible. We then describe the implementation of our approach in `DepQBF`[18] in Section 12. We illustrate with case studies including two-player games,[1] conformant planning,[19] and satisfiability of modal logic K[3] what kind of helpful information our enhanced notion of unsatisfiable cores can provide to a user (Section 13). We experimentally evaluate our approach on a subset of `QBFLIB`[20] (Section 14). Our experiments show that it is feasible to compute unsatisfiable cores of our enhanced notion on `QBFLIB` instances and that weakening of universal quantifications to existential quantifications does indeed occur.

A preliminary version of this paper appeared at ICTAI 2018.[21] This extended version contains the following major additions. (i) A discussion of related work from QCSP (part of Section 1.2); (ii) an extension of some of our results to the dual notion of enhanced satisfiable cores (Section 7); (iii) an argument that the transformation that we suggest in Section 8.1 does not push a formula into higher levels of the polynomial hierarchy despite potentially significantly increasing the alternation depth of a formula (Section 8.2); and (iv) a classification of universal quantifications into non-trivially, trivially, and not ∀-to-∃ reducible including two algorithms to underapproximate the set of non-trivially ∀-to-∃ reducible quantifications and a corresponding experimental evaluation (Section 11 and parts of Section 14).

1.2. *Related work*

QBF. Reimer et al.[22] propose soft variables, which — subject to a preference function — may take different positions in the prefix of a QBF. They then define the following optimization problem: find a placement for the soft variables that makes the resulting QBF satisfiable and, among all such placements, maximizes the valuation of the preference function. Reimer et al. reduce this optimization problem to a weighted partial MaxQBF problem[23] with a transformation that can be seen as a generalized version of the transformation that we propose in Section 8. (We discovered our transformation independently.) The authors implement their ideas in the tool `quantom`.[24] The main differences between our work and that of Reimer et al.[22] are as follows. Reimer et al.[22] are interested in satisfiable results, while we are mostly concerned with unsatisfiable results. While the two are related via hitting set duality, the approaches are complementary, and often one is used as part of a method to obtain the other (for an example see Ref. 16). Reimer et al.[22] make no connection to unsatisfiable cores. Reimer et al.[22] search for a maxim*um* solution, while we (optionally) search for a minim*al* solution. Reimer et al.[22] do not modify the matrix, whereas we (optionally) also remove clauses from the matrix.

Weakening universal to existential quantifiers has been called "quantifier abstraction" in a work on failed literal detection for QBF[25] and "existential abstraction" in the context of generalizing Q-resolution.[26] `QBFDD`[27] allows quantifier manipulations when minimizing failure-inducing input.

QCSP. Ferguson and O'Sullivan[28] define a number of weakening operations for QCSP (quantified constraint satisfaction problems). For universal quantifications they suggest to weaken a universal quantifier to an existential quantifier, to shrink the domain a universal quantification is ranging over, and to move a universal quantification to the left in the sequence of quantifications. They then extend an insertion-based algorithm to find minimal unsatisfiable cores[29] to handle lattices of weakening operations. Clearly, the idea of weakening universal quantifiers to existential quantifiers by Ferguson and O'Sullivan is the same as the main idea in this paper; in addition, they propose two more weakening operations related to universal quantifications. An obvious difference is that their work deals with QCSP, while we work on QBF (for a comparative survey of the quantifier-free fragments see Ref. 30). However, more importantly, their work remains at a fairly abstract level, both conceptually and algorithmically; in particular, they do not report on an implementation or experimental evaluation. Notice that for QBF in PCNF shrinking the domain of a universal quantification can be achieved by removing clauses of the appropriate polarity, i.e., by the traditional notion of unsatisfiable cores. In subsequent work[31] Mehta et al. extend Ferguson and O'Sullivan's work to take user preferences between different minimal unsatisfiable cores into account; here, too, no practical results are reported.

Bordeaux et al.[32] generalize a number of properties from CSP to QCSP. They investigate the relation between the validities of these properties in the quantified case and in the case in which all universal quantifications have been weakened to existential quantifications. They also mention other work in QCSP that uses universal quantifications weakened to existential quantifications.

Various. Shlyakhter et al.[33] support the debugging of unsatisfiable Alloy models by pointing out values of bound variables that are not relevant to the unsatisfiability. Let p be a Boolean variable in some formula $\forall p.f[p]$. The approach by Shlyakhter et al.[33] corresponds to weakening $f[\bot/p] \wedge f[\top/p]$ either to $f[\bot/p]$ or to $f[\top/p]$. Notice that this can be achieved by the traditional notion: it suffices to simply remove clauses with occurrences of p of the suitable polarity. We, on the other hand, can additionally weaken to $f[\bot/p] \vee f[\top/p]$.

Finally, our work shares the spirit of investigating the aspect of granularity in various notions including: unsatisfiable cores for propositional logic,[12,34] temporal logic,[11,35] and constraint programming;[28,36] equivalent formulas;[37] unrealizable cores;[11] vacuity;[38,39] justifications;[40–42] diagnoses;[43] and repair.[44,45] For a uniform treatment of some such notions and their relationships see Ref. 46.

2. Preliminaries

We consider QBF in PCNF;[6,47] using standard techniques any QBF can be turned into an equivalent QBF in PCNF.[47]

Let V be a set of *variables*; we use the letter p to denote variables. $\bot$ and $\top$ are the *Boolean constants false* and *true*. A variable, a Boolean constant, or their *negation* (denoted $\neg$) is a *literal*; literals are written as the letter l. A *disjunction* of literals $(l_1 \vee \ldots \vee l_n)$ is a *clause*, which we denote by the letter c. We use *implication* $\rightarrow$ as syntactic sugar within clauses as usual. A *conjunction* of clauses $c_1 \wedge \ldots \wedge c_n$ is a CNF (conjunctive normal form) formula; we write CNF formulas with the letter C. We treat clauses as sets of literals and CNF formulas as sets of clauses when this is convenient. A variable p that occurs only non-negated or only negated in a CNF formula C is *pure* in C. $\mathbb{B} = \{0, 1\}$ are the *Booleans*. A mapping v from V to $\mathbb{B}$ is an *assignment* for C. We define the evaluation of a CNF formula under an assignment and the (un)satisfiability of a CNF formula as usual.

$\forall$ denotes the *universal quantifier*, and $\exists$ denotes the *existential quantifier*, respectively. We represent quantifiers with the letter Q. If $Q_1, \ldots, Q_n \in \{\forall, \exists\}$ are quantifiers, if $p_1, \ldots, p_n \in V$ are pairwise different variables, and if C is a CNF formula whose variables are contained in $p_1, \ldots, p_n$, then $Q_1 p_1 \ldots Q_n p_n.C$ is a *QBF in PCNF*. $Q_1 p_1 \ldots Q_n p_n$ is called the *prefix*, and C is called the *matrix* of the QBF. We write prefixes as the letter Π. Let $\Pi.C$ be a a QBF in PCNF. Its *alternation depth* $ad(\Pi.C)$ is defined as one plus the number of alternations between $\forall$ and $\exists$ in Π. For $p \in V$ $(\Pi.C)[\bot/p]$ (resp. $(\Pi.C)[\top/p]$) denotes the QBF in PCNF in which Π

is unchanged and every occurrence of p in C is replaced with $\bot$ (resp. $\top$). We can now define the satisfiability of a QBF in PCNF as follows. $\forall p \Pi.C$ (resp. $\exists p \Pi.C$) is satisfiable iff $(\Pi.C)[\bot/p]$ and (resp. or) $(\Pi.C)[\top/p]$ are satisfiable. The satisfiability problem for QBF in PCNF is PSPACE-complete;[48] deciding the satisfiability of a QBF in PCNF with alternation depth at most $i \in \mathbb{N}$ and $\forall$ (resp. $\exists$) as the first quantifier is a Π_i^P-complete (resp. Σ_i^P-complete) problem,[49,50] where Π_i^P and Σ_i^P denote the i-th level of the polynomial hierarchy.

3. Enhanced Unsatisfiable Cores for QBF

In this section we introduce our enhanced notions of unsatisfiable cores for QBF. We complement the traditional notion of cores for QBF in PCNF (from now on called c-cores), which weakens only the matrix by removing clauses, with the notions of q-cores, which weakens only the prefix by turning universal quantifiers into existential quantifiers, and of qc-cores, which combines both kinds of weakening. First, we formally define c-, q-, and qc-cores (Definition 3.1). Then, we naturally extend the definitions of proper cores and unsatisfiable cores to q- and qc-cores (Definitions 3.2 and 3.3). Finally, we add the criterion of quantifier-minimal unsatisfiability to the traditional criterion of clause-minimal unsatisfiability (Definition 3.4). Let $\Pi.C$ be a QBF in PCNF.

Definition 3.1. (Core) (1) Let $C' \subseteq C$. Then $\Pi.C'$ is a *c-core* of $\Pi.C$. (2) Let $\Pi = Q_1 p_1 \ldots Q_n p_n$, $\Pi' = Q'_1 p_1 \ldots Q'_n p_n$ be prefixes such that, $\forall 1 \leq i \leq n$: if Q_i is $\exists$, then Q'_i is $\exists$; otherwise, $Q'_i \in \{\forall, \exists\}$. Then $\Pi'.C$ is a *q-core* of $\Pi.C$. (3) Let $\Pi.C'$ be a c-core of $\Pi.C$, and let $\Pi'.C'$ be a q-core of $\Pi.C'$. Then $\Pi'.C'$ is a *qc-core* of $\Pi.C$.

Some authors (e.g., Lonsing and Egly[17]) remove quantifications from the prefix of a c-core if the quantified variables cease to occur in the matrix of the c-core. In our implementation we do this as a generic postprocessing step and, therefore, we opt to keep our exposition simple and omit this step from Definition 3.1.

Definition 3.2. (Proper Core) Let $\Pi'.C'$ be a qc-core (resp. c-core, q-core) of $\Pi.C$ such that $\Pi' \neq \Pi$ or $C' \neq C$. Then $\Pi'.C'$ is a *proper qc-core* (resp. proper c-core, q-core) of $\Pi.C$.

Definition 3.3. (Unsatisfiable Core) Let $\Pi'.C'$ be a qc-core (resp. c-core, q-core) of $\Pi.C$ such that $\Pi'.C'$ is unsatisfiable. Then $\Pi'.C'$ is an *unsatisfiable qc-core* (resp. unsatisfiable c-core, q-core) of $\Pi.C$.

Definition 3.4. (Minimal Unsatisfiability) Let $\Pi.C$ be unsatisfiable such that there is no proper unsatisfiable c-core (resp. q-core) of $\Pi.C$. Then $\Pi.C$ is *c-minimally unsatisfiable* (resp. q-minimally unsatisfiable).

Example 3.1. As an example consider $\Pi.C = \forall p.(p) \wedge (\neg p)$. $\Pi.C$ is obviously unsatisfiable. It has four c-cores: $\Pi.C$, $\forall p.(p)$, $\forall p.(\neg p)$, and $\forall p.\top$. $\Pi.C$, $\forall p.(p)$, and

$\forall p.(\neg p)$ are unsatisfiable c-cores of $\Pi.C$; $\forall p.(p)$, $\forall p.(\neg p)$, and $\forall p.\top$ are proper c-cores of $\Pi.C$; and $\forall p.(p)$ and $\forall p.(\neg p)$ are both q- and c-minimally unsatisfiable.

$\Pi.C$ has two q-cores: $\Pi.C$ and $\exists p.(p) \wedge (\neg p)$. Both are unsatisfiable. $\exists p.(p) \wedge (\neg p)$ is the only proper q-core of $\Pi.C$ and the only q-minimally unsatisfiable q-core of $\Pi.C$. $\exists p.(p) \wedge (\neg p)$ is also c-minimally unsatisfiable.

Any c-core and any q-core is also a qc-core. $\Pi.C$ has three qc-cores that are both proper c-cores and proper q-cores of $\Pi.C$: $\exists p.(p)$, $\exists p.(\neg p)$, and $\exists p.\top$. All of them are satisfiable. □

4. QC-Cores Can Be Different From C-Cores

Unsatisfiable cores are commonly taken to be causes and/or explanations of unsatisfiability.[7–11,33] Some authors prefer minimally or minimum cardinality unsatisfiable cores,[7,9,51,52] and some authors use unsatisfiable cores as building blocks of more advanced explanations.[12] In this paper we take the view that a minimally unsatisfiable core represents a cause of unsatisfiability and gives rise to an explanation of unsatisfiability. We now show that our enhanced notion of unsatisfiable qc-cores for QBF in PCNF can identify additional causes of unsatisfiability (giving rise to additional explanations of unsatisfiability) that differ not only in terms of syntax but also in terms of both standard and proof-based semantics from the causes of unsatisfiability that are identified by the traditional notion of unsatisfiable c-cores.

We continue with $\forall p.(p) \wedge (\neg p)$ from Example 3.1. $\forall p.(p) \wedge (\neg p)$ has three q- and c-minimally unsatisfiable qc-cores $\forall p.(p)$, $\forall p.(\neg p)$, and $\exists p.(p) \wedge (\neg p)$. Obviously, the unsatisfiable q-core $\exists p.(p) \wedge (\neg p)$ differs syntactically from both unsatisfiable c-cores $\forall p.(p)$ and $\forall p.(\neg p)$. However, in general, the significance of syntactic differences may be limited; therefore, in the following we discuss differences in terms of semantics.

A standard semantics for unsatisfiable QBF are tree refutations.[53,54] Let $\Pi.C$ be an unsatisfiable QBF. Intuitively, a tree refutation for $\Pi.C$ shows which values to assign to the universally quantified variables in Π in order to falsify $\Pi.C$. A tree refutation for $\Pi.C$ is a tree with the following properties. (i) The labels of non-leaf nodes are variables in Π; the labels of leaf nodes are irrelevant. (ii) The labels of edges are Booleans; they represent assignments to the variables that are labeling their source nodes. (iii) If a node is labeled with a universally quantified variable, then it has one outgoing edge; it is labeled with either 0 or 1. (iv) If a node is labeled with an existentially quantified variable, then it has two outgoing edges; one is labeled with 0, the other with 1. (v) On every path from the root to a leaf node the sequence of labels on the non-leaf nodes matches the sequence of variables in the prefix Π. (vi) On every path from the root to a leaf node the assignment to the variables in Π that is induced by the path falsifies C.

$\forall p.(p)$ has a single tree refutation. Its root node is labeled p. The root node has a single outgoing edge labeled 0. $\exists p.(p) \wedge (\neg p)$ has a single tree refutation as well. Its root node is also labeled p. Here, the root node has two outgoing edges, labeled 0 and 1. Clearly, the tree refutations for $\forall p.(p)$ and $\exists p.(p) \wedge (\neg p)$ are different. They correspond to different ways to explain why $\forall p.(p) \wedge (\neg p)$ is unsatisfiable. For $\forall p.(p)$ setting p to 0 falsifies (p). For $\exists p.(p) \wedge (\neg p)$ setting p to 0 falsifies (p), while setting p to 1 falsifies $(\neg p)$. The reasoning for $\forall p.(\neg p)$ is analogous.

Let C_1 and C_2 be two matrices that are different but have the same sets of satisfying assignments. For any prefix Π, if $\Pi.C_1$ and $\Pi.C_2$ are unsatisfiable, then their sets of tree refutations are identical. I.e., tree refutations cannot always distinguish unsatisfiable cores. In that case we can use proof-theoretic semantics instead, which can be more discriminating.[55] For example, we can assign to each unsatisfiable QBF in PCNF the set of its Q-resolution proofs of unsatisfiability.[56] Then we can compare two unsatisfiable cores in terms of their sets of Q-resolution proofs of unsatisfiability. Q-resolution essentially allows for two operations (we omit some details and assume a working knowledge of resolution): (i) resolve two clauses on an existentially quantified literal; and (ii) remove a universally quantified literal l from a clause c if there is no existentially quantified literal in c that occurs to the right of the variable of l in the prefix. Then a QBF in PCNF is unsatisfiable iff the empty clause can be derived via Q-resolution.[56]

Using Q-resolution, $\forall p.(p)$ is proved to be unsatisfiable by removing p from (p). Also using Q-resolution, $\exists p.(p) \wedge (\neg p)$ is proved to be unsatisfiable by resolving (p) with $(\neg p)$. Hence, $\forall p.(p)$ and $\exists p.(p) \wedge (\neg p)$ have different sets of Q-resolution proofs of unsatisfiability. The reasoning for $\forall p.(\neg p)$ is analogous.

5. C-Minimal Unsatisfiability Implies Q-Minimal Unsatisfiability

In this section we show that any c-minimally unsatisfiable core is also q-minimally unsatisfiable.

Theorem 5.1. Let $\Pi.C$ be a c-minimally unsatisfiable QBF in PCNF such that every universally quantified variable in Π occurs in some clause in C. Then $\Pi.C$ is also q-minimally unsatisfiable. The converse is not true.

Proof. The first part directly follows from Lemma 5.1 below. A counterexample to disprove the converse is

$$\exists p_1 \exists p_2 \forall p_3 \,.\, (p_1 \rightarrow p_3) \wedge (p_3 \rightarrow p_1) \wedge (p_2 \rightarrow p_3) \wedge (p_3 \rightarrow p_2). \qquad \square$$

Lemma 5.1. *Let*

$$\Pi.C = Q_1 p_1 \dots Q_{l-1} p_{l-1} \forall p_l Q_{l+1} p_{l+1} \dots Q_n p_n \,.\, c_1 \wedge \dots \wedge c_{i-1} \wedge c_i \wedge c_{i+1} \wedge \dots \wedge c_m$$

be a QBF in PCNF such that p_l *occurs in* c_i*, let* $\Pi'.C'$ *be obtained from* $\Pi.C$ *by changing* $\forall p_l$ *to* $\exists p_l$ *in* Π*, and let* $\Pi''.C''$ *be obtained from* $\Pi.C$ *by removing* c_i *from* C*. If* $\Pi'.C'$ *is unsatisfiable, then so is* $\Pi''.C''$*.*

Proof. By induction over l. For the base case let $l - 1 = 0$. By assumption $\Pi'.C' = \exists p_l Q_{l+1} p_{l+1} \dots Q_n p_n.C'$ is unsatisfiable. Let $\Pi'_{l+1} = Q_{l+1} p_{l+1} \dots Q_n p_n$.

By the semantics of $\exists p_l$ both $(\Pi'_{l+1}.C')[\bot/p_l]$ and $(\Pi'_{l+1}.C')[\top/p_l]$ are unsatisfiable. W.l.o.g. let p_l occur non-negated in c_i. Hence, $(\Pi'_{l+1}.C'')[\top/p_l]$ is also unsatisfiable. Finally, by the semantics of $\forall p_l$, $\forall p_l \Pi'_{l+1}.C'' = \Pi''.C''$ is unsatisfiable as desired.

For the inductive case let $l - 1 > 0$. First let $Q_1 = \exists$. By assumption $\Pi'.C' = \exists p_1 Q_2 p_2 \ldots Q_{l-1} p_{l-1} \exists p_l Q_{l+1} p_{l+1} \ldots Q_n p_n.C'$ is unsatisfiable. Let $\Pi'_{2,\exists} = Q_2 p_2 \ldots Q_{l-1} p_{l-1} \exists p_l Q_{l+1} p_{l+1} \ldots Q_n p_n$ and $\Pi'_{2,\forall} = Q_2 p_2 \ldots Q_{l-1} p_{l-1} \forall p_l Q_{l+1} p_{l+1} \ldots Q_n p_n$. By the semantics of $\exists p_1$ both $(\Pi'_{2,\exists}.C')[\bot/p_1]$ and $(\Pi'_{2,\exists}.C')[\top/p_1]$ are unsatisfiable. With the inductive assumption both $(\Pi'_{2,\forall}.C'')[\bot/p_1]$, and $(\Pi'_{2,\forall}.C'')[\top/p_1]$ are unsatisfiable as well. Finally, by the semantics of $\exists p_1$, $\exists p_1 \Pi'_{2,\forall}.C'' = \Pi''.C''$ is unsatisfiable as desired. $Q_1 = \forall$ is similar. □

It is tempting to think that Theorem 5.1 would call into question the usefulness of unsatisfiable q- or qc-cores, because it proves that essentially any c-minimally unsatisfiable c-core is already a q-minimally unsatisfiable c-core (and qc-core). However, Theorem 5.1 does not preclude the existence of additional (possibly c- and q-minimally) unsatisfiable q- and qc-cores that are different from any unsatisfiable c-core; in fact, the existence of such cores has been shown in Section 4.

6. Complexity

Let CMF denote the set of c-minimally unsatisfiable QBF in PCNF, let QMF denote the set of q-minimally unsatisfiable QBF in PCNF, and let QCMF denote CMF ∩ QMF. Kleine-Büning and Zhao established PSPACE-completeness of CMF.[15] Here we extend this result to QMF and QCMF.

Theorem 6.1. QMF and QCMF are PSPACE-complete.

Proof. Clearly, QMF and QCMF are in PSPACE. We show PSPACE-hardness of QMF by a reduction from CMF. Let $\Pi.C = Q_1 p_1 \ldots Q_m p_m.c_1 \wedge \ldots \wedge c_n$ be a QBF in PCNF. Construct $\Pi'.C$ from $\Pi.C$ by removing those universal quantifications from Π whose variables do not occur in C. Let

$$\Pi''.C'' = \Pi' \forall p'_1 \ldots \forall p'_n.(c_1 \vee p'_1) \wedge \ldots \wedge (c_n \vee p'_n)$$

where $p'_1 \ldots p'_n$ are fresh. The size of $\Pi''.C''$ is obviously linear in the size of $\Pi.C$. We show that $\Pi.C$ is in CMF iff $\Pi''.C''$ is in QMF. First assume that $\Pi.C$ is in CMF. Then $\Pi'.C$ is also in CMF and, by Theorem 5.1, in QMF. Hence, $\Pi''.C''$ is in QMF as well. Now assume that $\Pi.C$ is not in CMF. If $\Pi.C$ is satisfiable, then so is $\Pi''.C''$; thus, $\Pi''.C'' \notin$ QMF. Let $\Pi.C$ be unsatisfiable. Clearly, $\Pi'.C$ is also not in CMF. For some $0 \leq i \leq n$ let c_i be a clause that can be removed from C without making the resulting QBF satisfiable. Then

$$\Pi' \forall p'_1 \ldots \forall p'_{i-1} \exists p'_i \forall p'_{i+1} \ldots \forall p'_n.(c_1 \vee p'_1) \wedge \ldots \wedge (c_n \vee p'_n),$$

which is a proper q-core of $\Pi''.C''$, is unsatisfiable. Hence, $\Pi''.C''$ is not in QMF. Thus, we have PSPACE-hardness of QMF. The proof for QCMF is similar. □

7. The Dual Notion: Enhanced Satisfiable Cores

We now briefly discuss the dual notion of satisfiable cores. As stated before, unsatisfiable cores help to explain the unsatisfiability of a formula. While that already is very useful, it is often necessary to modify the unsatisfiable formula such that it becomes satisfiable, e.g., when the formula is part of a system description and its unsatisfiability indicates the presence of contradictory requirements. Our definition of enhanced unsatisfiable cores for QBF in PCNF can easily be extended for this purpose by aiming for cores that are satisfiable rather than unsatisfiable. [a] [b]

We begin by supplying straightforward definitions of satisfiable cores (Definition 7.1) and maximally satisfiable cores (Definition 7.2). The latter differs from the corresponding definition of minimally unsatisfiable cores in Definition 3.4 in that it has to explicitly limit the strengthening to those quantifications that were universal and those clauses that were present in the original QBF. The following Example 7.1 then shows that indeed semantically different satisfiable cores can be obtained using our enhanced notion of satisfiable cores. Hence, not only can our enhanced notion of cores for QBF in PCNF produce strictly larger sets of explanations for unsatisfiability than the traditional notion, but it can also generate strictly larger sets of diagnoses, repairs, and repaired formulas. We conclude this section by showing in Theorem 7.1 a dual result to Theorem 5.1. While in the next Section 8 we still consider satisfiable cores as well as unsatisfiable cores, extending the remainder of this paper to satisfiable cores is left as future work. Let $\Pi.C$ be a QBF in PCNF.

Definition 7.1. (Satisfiable Core) Let $\Pi'.C'$ be a qc-core (resp. c-core, q-core) of $\Pi.C$ such that $\Pi'.C'$ is satisfiable. Then $\Pi'.C'$ is a *satisfiable qc-core* (resp. satisfiable c-core, q-core) of $\Pi.C$.

Definition 7.2. (Maximally Satisfiable Core) Let $x \in \{c, q, qc\}$. Let $\Pi'.C'$ be a satisfiable x-core of $\Pi.C$. Let there be no satisfiable qc-core $\Pi''.C''$ of $\Pi.C$ such that $\Pi'.C'$ is a proper c-core (resp. q-core) of $\Pi''.C''$. Then $\Pi'.C'$ is a *c-maximally satisfiable* (resp. q-maximally satisfiable) x-core of $\Pi.C$.

Example 7.1. We continue Example 3.1 with $\Pi.C = \forall p.(p) \wedge (\neg p)$. $\Pi.C$ has one satisfiable c-core $\forall p.\top$ and three satisfiable qc-cores $\exists p.(p)$, $\exists p.(\neg p)$, and $\exists p.\top$. $\forall p.\top$, $\exists p.(p)$, and $\exists p.(\neg p)$ are both q- and c-maximally satisfiable cores of $\Pi.C$. The q-

[a] In parts of the literature in a set-based setting the complements of satisfiable subsets (cores) are referred to as diagnoses[13] and sometimes as repair (solutions);[44] in that sense our satisfiable cores constitute repair*ed* formulas.

[b] It is well known that unsatisfiable cores and satisfiable cores are also connected via hitting set duality (e.g., Ref. 13; for a generic formulation see Slaney[57]). Roughly speaking, in a set-based setting a hitting set of the set of all unsatisfiable subsets of some set S is the complement of a satisfiable subset of S. This provides an additional avenue to obtain the enhanced notions of satisfiable q- and qc-cores from the enhanced notions of unsatisfiable q- and qc-cores: For QBF in PCNF the matrix already is a set of clauses. The prefix can easily be treated as a set by considering non-weakened universal quantifications as present in the set and universal-weakened-to-existential quantifications as absent from the set; alternatively, the transformation in Section 8 can be applied.

and c-maximally satisfiable qc-cores $\exists p.(p)$ and $\exists p.(\neg p)$ of $\Pi.C$ can be shown to be semantically different from the only c-maximally satisfiable c-core $\forall p.\top$ of $\Pi.C$ in a similar fashion as has been done for unsatisfiable cores in Section 4. □

Theorem 7.1. Let $\Pi'.C'$ be a satisfiable qc-core of $\Pi.C$ such that every variable that occurs universally quantified in Π and existentially quantified in Π' also occurs in some clause in $C \setminus C'$. If $\Pi'.C'$ is c-maximally satisfiable, then it is also q-maximally satisfiable. The converse is not true.

Proof. The first part is easily obtained by repeated application of the following Lemma 7.1. As a counterexample for the second part consider

$$\Pi.C = \exists p_1 \exists p_2 \forall p_3 \,.\, (p_1 \rightarrow p_3) \wedge (p_3 \rightarrow p_1) \wedge (p_2 \rightarrow p_3) \wedge (p_3 \rightarrow p_2)$$

with satisfiable qc-core $\Pi'.C' = \exists p_1 \exists p_2 \exists p_3 \,.\, (p_1 \rightarrow p_3) \wedge (p_3 \rightarrow p_1)$. □

Lemma 7.1. *Let* $\Pi.C = Q_1 p_1 \dots Q_{l-1} p_{l-1} \forall p_l Q_{l+1} p_{l+1} \dots Q_n p_n.C$ *be a QBF in PCNF. Let* $\Pi'.C' = Q'_1 p_1 \dots Q'_{l-1} p_{l-1} \exists p_l Q'_{l+1} p_{l+1} \dots Q'_n p_n.C'$ *be a c-maximally satisfiable qc-core of* $\Pi.C$ *such that* p_l *occurs in some clause in* $C \setminus C'$. *Then* $\Pi''.C' = Q'_1 p_1 \dots Q'_{l-1} p_{l-1} \forall p_l Q'_{l+1} p_{l+1} \dots Q'_n p_n.C'$ *is unsatisfiable.*

Proof. Corollary of Lemma 5.1. □

8. A2AECC: Q- and QC-Cores as C-Cores

We now present a source-to-source transformation on QBF in PCNF that allows to cast q- and qc-cores of the original formula as c-cores of the transformed formula.

8.1. *Definition and correctness*

Let $\Pi.C$ be a QBF in PCNF. For each universally quantified variable p_i in $\Pi.C$ the transformation replaces the quantification $\forall p_i$ in the prefix Π with $\forall p'_i \exists p_i$, where p'_i is a fresh variable, and conjoins the matrix C with two clauses $(p_i \rightarrow p'_i)$ and $(p'_i \rightarrow p_i)$. Hence, the acronym A2AECC.

Definition 8.1. (A2AECC) Let $\Pi.C = Q_1 p_1 \dots Q_n p_n.C$. Let $p'_1, \dots, p'_n$ be fresh. Let, for all $1 \leq i \leq n$,

$$a2ae(Q_i p_i) = \begin{cases} \forall p'_i \exists p_i & \text{if } Q_i = \forall \\ \exists p_i & \text{otherwise,} \end{cases}$$

and

$$a2cc(Q_i p_i) = \begin{cases} (p_i \rightarrow p'_i) \wedge (p'_i \rightarrow p_i) & \text{if } Q_i = \forall \\ \top & \text{otherwise.} \end{cases}$$

Then

$$a2aecc(\Pi.C) = a2ae(Q_1 p_1) \dots a2ae(Q_n p_n).(\bigwedge_{1 \leq i \leq n} a2cc(Q_i p_i)) \wedge C.$$

Let $\Pi.C$ be an unsatisfiable QBF in PCNF. Definition 8.1 allows to reduce the computation of an unsatisfiable q- or qc-core $\Pi'.C'$ of $\Pi.C$ to the computation of an unsatisfiable c-core of $a2aecc(\Pi.C)$ as follows.

(1) Apply the A2AECC-transformation: let $\Pi_{a2aecc}.C_{a2aecc} = a2aecc(\Pi.C)$.
(2) Compute an unsatisfiable c-core $\Pi_{a2aecc}.C'_{a2aecc}$ of $\Pi_{a2aecc}.C_{a2aecc}$.
(3) Compute the matrix C': let $C' = \begin{cases} C & \text{if a q-core is desired,} \\ C \cap C'_{a2aecc} & \text{if a qc-core is desired.} \end{cases}$
(4) Compute the prefix Π': take Π and replace each quantification $Q_i p_i$ in Π with $Q'_i p_i$ where

$$Q'_i = \begin{cases} \exists & \text{if } (Q_i = \exists) \text{ or } (Q_i = \forall \text{ and } C'_{a2aecc} \cap \{(p_i \rightarrow p'_i), (p'_i \rightarrow p_i)\} = \emptyset), \\ \forall & \text{otherwise.} \end{cases}$$

For the dual case of satisfiable cores it its sufficient to compute a satisfiable c-core in step (2) and to replace the comparison "$= \emptyset$" with "$\neq \{(p_i \rightarrow p'_i), (p'_i \rightarrow p_i)\}$" in step (4). I.e., the case of one clause remaining in the c-core $\Pi_{a2aecc}.C'_{a2aecc}$ of the two clauses introduced by the A2AECC-transformation for a universal quantification is decided in favor of a universal quantification for unsatisfiable cores and in favor of an existential quantification for satisfiable cores. In Theorem 8.1 below we prove the correctness of the above procedure. The proof uses the following Lemma 8.1, which directly follows from the semantics of QBF.

Lemma 8.1. *Let*

$$\Pi.C = Q_1 p_1 \ldots Q_{l-1} p_{l-1} \forall p_l Q_{l+1} p_{l+1} \ldots Q_m p_m . c_1 \wedge \ldots \wedge c_n$$

be a QBF in PCNF. Let p'_l be fresh. Let

$$\Pi'.C' = Q_1 p_1 \ldots Q_{l-1} p_{l-1} \forall p'_l \exists p_l Q_{l+1} p_{l+1} \ldots Q_m p_m . (p_l \rightarrow p'_l) \wedge (p'_l \rightarrow p_l) \wedge c_1 \wedge \ldots \wedge c_n .$$

Then $\Pi.C$ is satisfiable iff $\Pi'.C'$ is satisfiable.

Theorem 8.1. Let $\Pi.C$ be a QBF in PCNF. Let P be a subset of the universally quantified variables in Π. Let Π' be obtained from Π by weakening $\forall p$ to $\exists p$ for all $p \in P$. Let $\Pi_{a2aecc}.C_{a2aecc} = a2aecc(\Pi.C)$ and let

$$C'_{a2aecc} = C_{a2aecc} \setminus \bigcup_{p \in P} \{(p \rightarrow p'), (p' \rightarrow p)\}.$$

Then (1) $\Pi'.C$ is a q-core of $\Pi.C$. (2) $\Pi_{a2aecc}.C'_{a2aecc}$ is a c-core of $\Pi_{a2aecc}.C_{a2aecc}$. (3) $\Pi'.C$ is satisfiable iff $\Pi_{a2aecc}.C'_{a2aecc}$ is satisfiable.

Proof. Claims (1), (2) are directly obtained from Definition 3.1. To prove claim (3) proceed by induction on the cardinality of P. Establish the base case $|P| = 0$ by repeated application of Lemma 8.1. For the inductive case assume that the claim is true for every P such that $|P| = n$. Now let $P = \{p_1, \ldots, p_{n+1}\}$; i.e.,

$|P| = n+1$. Obtain Π'' from Π by weakening $\forall p_{n+1}$ in Π to $\exists p_{n+1}$ in Π''. Let $\Pi''_{a2aecc}.C''_{a2aecc} = a2aecc(\Pi''.C)$. By the inductive assumption $\Pi'.C$ and

$$\Pi''_{a2aecc}.C''_{a2aecc} \setminus \bigcup_{p\in\{p_1,\ldots,p_n\}} \{(p \rightarrow p'), (p' \rightarrow p)\}$$

are equisatisfiable. By the construction of C'_{a2aecc} and C''_{a2aecc} we have

$$C'_{a2aecc} = C''_{a2aecc} \setminus \bigcup_{p\in\{p_1,\ldots,p_n\}} \{(p \rightarrow p'), (p' \rightarrow p)\}.$$

Hence, $\Pi'.C$ and $\Pi''_{a2aecc}.C'_{a2aecc}$ are equisatisfiable. Notice that Π_{a2aecc} only differs from Π''_{a2aecc} by having $\forall p'_{n+1}\exists p_{n+1}$ instead of $\exists p_{n+1}$. Moreover, p'_{n+1} does not occur in C'_{a2aecc}. Hence, $\Pi''_{a2aecc}.C'_{a2aecc}$ and $\Pi_{a2aecc}.C'_{a2aecc}$ are equisatisfiable. Finally, with transitivity $\Pi'.C$ is satisfiable iff $\Pi_{a2aecc}.C'_{a2aecc}$ is satisfiable as desired. □

8.2. *Complexity-theoretic considerations*

Remember that determining the satisfiability of QBF in PCNF with alternation depth m is a complete problem for the m-th level of the polynomial hierarchy.[49,50] The following proposition is immediate from Definition 8.1.

Proposition 8.1. *Let $\Pi.C$ be a QBF in PCNF with m universally quantified variables. Then $a2aecc(\Pi.C)$ has alternation depth $2m$ or $2m+1$.*

Notice that Lemma 8.1 works in both directions, i.e., it can also be used to turn $\Pi'.C'$ in Lemma 8.1 into $\Pi.C$ while preserving (un)satisfiability. Hence, we can define a reverse transformation that, given a QBF in PCNF $\Pi.C$, checks once[c] for each universal quantifier in Π, whether that quantification is an instance of the reverse direction of Lemma 8.1 and, if yes, replaces $\forall p'_i \exists p_i$ with $\forall p_i$ in Π and removes $(p_i \rightarrow p'_i)$ and $(p'_i \rightarrow p_i)$ from C; we call the resulting reverse transformation $aecc2a$. It is easy to see that $aecc2a$ can be performed in deterministic polynomial time and that, for any QBF $\Pi.C$, we have $ad(aecc2a(\Pi.C)) \leq ad(\Pi.C)$ and $ad(aecc2a(a2aecc(\Pi.C))) \leq ad(\Pi.C)$. For $m \geq 1$ let $QBF_{m,\forall}$ (resp. $QBF_{m,\exists}$) denote the set of all QBF in PCNF that either have alternation depth less than m, or that have alternation depth m and $\forall$ (resp. $\exists$) as the first quantifier. Then we have

Proposition 8.2. *Let $m \geq 1$. (1) The satisfiability problem for $QBF_{m,\forall} \cup \{a2aecc(\Pi.C) \mid \Pi.C \in QBF_{m,\forall}\}$ is in Π^P_m. (2) The satisfiability problem for $QBF_{m,\exists} \cup \{a2aecc(\Pi.C) \mid \Pi.C \in QBF_{m,\exists}\}$ is in Σ^P_m.*

Hence, while the A2AECC-transformation potentially significantly increases the alternation depth of a QBF in PCNF, from a complexity-theoretic point of view this does not push determining the satisfiability of QBF in PCNF into higher levels of the polynomial hierarchy. In Section 9 we discuss a variant of the transformation that does not affect alternation depth but has different semantics.

[c] I.e., if $\forall p'_i \exists p_i$ has been replaced with $\forall p_i$, then $\forall p_i$ is not checked for replacement again.

8.3. *Optimizations*

Let a variable p be universally quantified in a prefix Π and pure in a matrix C. If p occurs only non-negated (resp. negated) in C, then $(p' \rightarrow p)$ (resp. $(p \rightarrow p')$) is a quantified blocked clause[58] in $a2aecc(\Pi.C)$ and can be eliminated from $a2aecc(\Pi.C)$.

If a solver for QBF in PCNF allows to group clauses for the computation of unsatisfiable c-cores,[59,60] as does `DepQBF`,[17] then placing each pair of clauses $(p_i \rightarrow p_i'), (p_i' \rightarrow p_i)$ introduced by Definition 8.1 in a separate clause group ensures that either none or both of $(p_i \rightarrow p_i'), (p_i' \rightarrow p_i)$ are present in a c-core of $a2aecc(\Pi.C)$.

8.4. *Example*

Example 8.1. We use $\Pi.C = \forall p.(p) \wedge (\neg p)$ from Example 3.1 again. We have

$$a2aecc(\Pi.C) = \forall p' \exists p.(p \rightarrow p') \wedge (p' \rightarrow p) \wedge (p) \wedge (\neg p).$$

The unsatisfiable c-core $\Pi'.C_1' = \forall p' \exists p.(p \rightarrow p') \wedge (p' \rightarrow p) \wedge (p)$ of $a2aecc(\Pi.C)$ corresponds to the unsatisfiable c-core $\forall p.(p)$ of $\Pi.C$; the unsatisfiable c-core $\Pi'.C_2' = \forall p' \exists p.(p \rightarrow p') \wedge (p' \rightarrow p) \wedge (\neg p)$ of $a2aecc(\Pi.C)$ corresponds to the unsatisfiable c-core $\forall p.(\neg p)$ of $\Pi.C$; and the unsatisfiable c-core $\Pi'.C_3' = \forall p' \exists p.(p) \wedge (\neg p)$ of $a2aecc(\Pi.C)$ corresponds to the unsatisfiable q-core $\exists p.(p) \wedge (\neg p)$ of $\Pi.C$. $\Pi'.C_3'$ is c-minimally unsatisfiable, while $\Pi'.C_1'$ and $\Pi'.C_2'$ are not; however, when using a clause group for $(p \rightarrow p'), (p' \rightarrow p)$ as discussed above, then $\Pi'.C_1'$ and $\Pi'.C_2'$ are c-minimally unsatisfiable as well under a suitable definition of c-minimality that takes clause groups into account. □

8.5. *Discussion*

The A2AECC-transformation in Definition 8.1, Theorem 8.1 is also of theoretical interest in that it may enable to extend a result for clauses to include universal quantifiers. For example, besides directly extending our enhanced notion of unsatisfiable cores to satisfiable cores in Section 7, the A2AECC-transformation provides an additional avenue to obtain the enhanced notions of satisfiable q- and qc-cores from the enhanced notions of unsatisfiable q- and qc-cores via hitting set duality[57] on the sets of matrices of unsatisfiable c-cores of A2AECC-transformed formulas.

9. A Variant of A2AECC: Reducing Alternation Depth by Reducing Precision

In this section we discuss a variant of the A2AECC-transformation. It avoids the potentially large increase in alternation depth between $\Pi.C$ and $a2aecc(\Pi.C)$ (see Proposition 8.1). However, it underapproximates the set of quantifiers that can be weakened from universal to existential in an unsatisfiable q- or qc-core of $\Pi.C$.

Assume a QBF in PCNF $\Pi.C$ that has n universal quantifiers and alternation depth m. Assume further that $\forall p_{i,1} \ldots \forall p_{i,n_i}$ is a maximal sequence (called *block*) of universal quantifications in Π. The A2AECC-transformation turns $\forall p_{i,1} \ldots \forall p_{i,n_i}$ into $\forall p'_{i,1} \exists p_{i,1} \ldots \forall p'_{i,n_i} \exists p_{i,n_i}$. Overall, with Proposition 8.1, the increase in alternation depth caused by the A2AECC-transformation, $ad(a2aecc(\Pi.C)) - ad(\Pi.C)$, is $2 \cdot n - m$ if $\Pi.C$ starts with $\forall$ and $1 + 2 \cdot n - m$ otherwise.

Let $a2aecc'$ denote the variant of Definition 8.1 that turns each block of universal quantifications $\forall p_{i,1} \ldots \forall p_{i,n_i}$ into $\forall p'_{i,1} \ldots \forall p'_{i,n_i} \exists p_{i,1} \ldots \exists p_{i,n_i}$. Here, the increase in alternation depth $ad(a2aecc'(\Pi.C)) - ad(\Pi.C)$ is 0 or 1. Moreover, using tree refutations (see Section 4), it is easy to see that $a2aecc(\Pi.C)$ and $a2aecc'(\Pi.C)$ are equisatisfiable. As shown in Theorem 8.1, the removal of $(p_{i,i'} \rightarrow p'_{i,i'}) \wedge (p'_{i,i'} \rightarrow p_{i,i'})$ from $a2aecc(\Pi.C)$ corresponds to weakening $\forall p_{i,1} \ldots \forall p_{i,i'-1} \forall p_{i,i'} \forall p_{i,i'+1} \ldots \forall p_{i,n_i}$ to $\forall p_{i,1} \ldots \forall p_{i,i'-1} \exists p_{i,i'} \forall p_{i,i'+1} \ldots \forall p_{i,n_i}$ in $\Pi.C$. It is straightforward to show that the removal of $(p_{i,i'} \rightarrow p'_{i,i'}) \wedge (p'_{i,i'} \rightarrow p_{i,i'})$ from $a2aecc'(\Pi.C)$ instead corresponds to weakening $\forall p_{i,1} \ldots \forall p_{i,i'-1} \forall p_{i,i'} \forall p_{i,i'+1} \ldots \forall p_{i,n_i}$ to $\forall p_{i,1} \ldots \forall p_{i,i'-1} \forall p_{i,i'+1} \ldots \forall p_{i,n_i} \exists p_{i,i'}$ in $\Pi.C$.

By the semantics of QBF moving an existential quantification in the prefix to the right weakens the QBF under consideration. Therefore, the unsatisfiability of a c-core of $a2aecc'(\Pi.C)$ implies the unsatisfiability of the corresponding c-core of $a2aecc(\Pi.C)$. The converse is not true, as can be seen by considering $\Pi.C = \forall p_1 \forall p_2.(p_1 \rightarrow p_2) \wedge (p_2 \rightarrow p_1)$. Weakening $\forall p_1$ to $\exists p_1$ in $\Pi.C$ leads to the unsatisfiable $\exists p_1 \forall p_2.(p_1 \rightarrow p_2) \wedge (p_2 \rightarrow p_1)$. Correspondingly, removing $(p_1 \rightarrow p'_1) \wedge (p'_1 \rightarrow p_1)$ from $a2aecc(\Pi.C)$ produces

$$\forall p'_1 \exists p_1 \forall p'_2 \exists p_2.(p_2 \rightarrow p'_2) \wedge (p'_2 \rightarrow p_2) \wedge (p_1 \rightarrow p_2) \wedge (p_2 \rightarrow p_1),$$

which, in line with Theorem 8.1, is unsatisfiable as well. On the other hand, removing $(p_1 \rightarrow p'_1) \wedge (p'_1 \rightarrow p_1)$ from $a2aecc'(\Pi.C)$ produces

$$\forall p'_1 \forall p'_2 \exists p_1 \exists p_2.(p_2 \rightarrow p'_2) \wedge (p'_2 \rightarrow p_2) \wedge (p_1 \rightarrow p_2) \wedge (p_2 \rightarrow p_1),$$

which is satisfiable, as is $\forall p_2 \exists p_1.(p_1 \rightarrow p_2) \wedge (p_2 \rightarrow p_1)$.

To conclude this section we discuss an alternative perspective on the semantics of $a2aecc'$. $a2aecc$ takes the positions of quantifications within a quantifier block as fixed; in other words, it regards a block of universal quantifications as an (ordered) *list* of quantifications. Notice that this is by no means mandatory: by the semantics of QBF arbitrarily shuffling the quantifications within a quantifier block does not affect the satisfiability of the resulting QBF. Hence, in an alternative approach a quantifier block could also be regarded as an (unordered) *set* of quantifications. In the light of that, $a2aecc'$ can be interpreted as making use of the view of a quantifier block as a set rather than as a list of quantifications and pushing those quantifications that have been weakened from universal to existential to the right of their quantifier block (i.e., towards the inside of the QBF). We call the semantics induced by $a2aecc$ *list semantics* and the semantics induced by $a2aecc'$ *set-inner semantics*. List semantics acts conservatively by assigning maximal meaning to the order of the quantifications in a quantifier block, whereas set-inner semantics acts relaxed by assigning no meaning at all to the order of quantifications in a quantifier block.

Finally, remember that, as discussed above, while shuffling quantifications within a block of quantifiers preserves satisfiability, weakening universal to existential quantifications is not the same in list and in set-inner semantics.

10. Interpreting Unsatisfiable Q- and QC-Cores

We now explain that a universal quantifier may be weakened to an existential quantifier in an unsatisfiable core for two quite different reasons and that it is easier to see which of the two reasons caused a weakening if the core is c-minimally unsatisfiable.

Let $\Pi.C$ be an unsatisfiable QBF in PCNF. Let $\Pi'.C'$ be an unsatisfiable q- or qc-core of $\Pi.C$. Let $\forall p$ be a universal quantification in Π that has been weakened to $\exists p$ in Π'. Finally, let $C'' \subseteq C'$ such that $\Pi'.C''$ is c-minimally unsatisfiable (clearly, such C'' exists). We distinguish two cases. In the first case p occurs in some clause c in C''. Then $\Pi'.C''$ represents a cause of the unsatisfiability of $\Pi.C$ in which c, including its occurrence of p, is required but in which p only needs to be existentially quantified (as it is in Π') rather than universally quantified (as it is in Π). In the second case p does not occur in any clause of C''. Then $\Pi'.C''$ represents a cause of the unsatisfiability of $\Pi.C$ in which p is not required at all; i.e., the quantification of p could be removed from Π' entirely.

Notice that in a q- or qc-core that is unsatisfiable but not c-minimally unsatisfiable both cases may apply for different choices of C''. Hence, if $\forall p$ has been weakened to $\exists p$ in a non-c-minimally unsatisfiable q- or qc-core $\Pi'.C'$ of $\Pi.C$, then the weakening of $\forall p$ to $\exists p$ should be interpreted with some care. If, on the other hand, $\forall p$ has been weakened to $\exists p$ in a c-minimally unsatisfiable q- or qc-core $\Pi'.C'$ of $\Pi.C$, then it should be checked whether C' contains p or not (if not, our implementation removes $\exists p$ from Π' during postprocessing) to determine which of the two cases above applies.

Example 10.1. As an example consider

$$\Pi.C = \forall p_1 \forall p_2 \forall p_3 \exists p_4.(p_1 \to p_2) \land (p_2 \to p_1) \land (p_3 \to p_4)$$

with a non-c-minimally unsatisfiable qc-core

$$\Pi'.C' = \exists p_1 \forall p_2 \exists p_3 \exists p_4.(p_1 \to p_2) \land (p_2 \to p_1) \land (p_3 \to p_4).$$

We can see by inspection that the unsatisfiability of $\Pi'.C'$ is caused by $\exists p_1 \forall p_2.(p_1 \to p_2) \land (p_2 \to p_1)$ and that, obviously, for the unsatisfiability of $\Pi'.C'$ it is sufficient that p_1 is existentially quantified. Hence, the weakening of $\forall p_1$ in $\Pi.C$ to $\exists p_1$ in $\Pi'.C'$ gives us useful additional information about the unsatisfiability of $\Pi.C$. On the other hand, $\exists p_3 \exists p_4.(p_3 \to p_4)$ plays no role in the unsatisfiability of $\Pi'.C'$. Hence, the weakening of $\forall p_3$ in $\Pi.C$ to $\exists p_3$ in $\Pi'.C'$ gives us little to no information about the unsatisfiability of $\Pi.C$. $\Pi'.C'$ has only one c-minimally unsatisfiable core: $\Pi'.C'' = \exists p_1 \forall p_2 \exists p_3 \exists p_4.(p_1 \to p_2) \land (p_2 \to p_1)$. Remember that by definition every clause in a c-minimally unsatisfiable core is essential for unsatisfiability. As we can see, p_1 does and p_3 does not occur in the matrix C''. $\square$

11. ∀-to-∃ Reducibility

We now lift the discussion of Section 10 from a single unsatisfiable core to the entire formula $\Pi.C$ by partitioning the set of universally quantified variables in Π into three sets as follows. The first set contains those universally quantified variables p of Π for which a c-minimally unsatisfiable qc-core $\Pi'.C'$ of $\Pi.C$ exists such that p is existentially quantified in Π' and occurs in C'; these are the variables that can actually still be relevant for the unsatisfiability of $\Pi.C$ when weakened from universally to existentially quantified. The second set contains those universally quantified variables of Π that can be weakened to existentially quantified variables without making the result satisfiable, but for which no c-minimally unsatisfiable qc-core $\Pi'.C'$ of $\Pi.C$ exists in which they are existentially quantified in Π' and occur in C'; these are the variables that cannot be relevant for the unsatisfiability of $\Pi.C$ when weakened from universally to existentially quantified. Finally, the third set contains those universally quantified variables of Π that cannot be weakened to existentially quantified variables without making the result satisfiable.

Definition 11.1. (∀-to-∃ Reducibility) Let $\Pi.C$ be unsatisfiable, and let $\forall p$ occur in Π. (1) If there exists a c-minimally unsatisfiable qc-core $\Pi'.C'$ of $\Pi.C$ such that $\forall p$ in Π has been weakened to $\exists p$ in Π' and such that p occurs in C', then $\forall p$ is *non-trivially* ∀*-to-*∃ *reducible* in $\Pi.C$. (2) If $\forall p$ is not non-trivially ∀-to-∃ reducible in $\Pi.C$ but there exists an unsatisfiable q-core $\Pi'.C$ of $\Pi.C$ such that $\forall p$ in Π has been weakened to $\exists p$ in Π', then $\forall p$ is *trivially* ∀*-to-*∃ *reducible* in $\Pi.C$. (3) If there exists no unsatisfiable q-core $\Pi'.C$ of $\Pi.C$ in which $\forall p$ has been weakened to $\exists p$, then $\forall p$ is *not* ∀*-to-*∃ *reducible* in $\Pi.C$.

Let p be a universally quantified variable in Π. If p is pure in C, then — because of the pure literal rule for existentially quantified variables[6] — $\forall p$ is either trivially or not ∀-to-∃ reducible in $\Pi.C$.

Example 11.1. We continue Example 10.1. In

$$\Pi.C = \forall p_1 \forall p_2 \forall p_3 \exists p_4.(p_1 \rightarrow p_2) \wedge (p_2 \rightarrow p_1) \wedge (p_3 \rightarrow p_4)$$

p_1 is non-trivially ∀-to-∃ reducible, p_2 is not ∀-to-∃ reducible, and p_3 is trivially ∀-to-∃ reducible. □

To better understand the potential for weakening universal to existential quantifiers we are interested in computing which variables in an unsatisfiable QBF $\Pi.C$ are non-trivially ∀-to-∃ reducible. A precise result might require finding all c-minimally unsatisfiable qc-cores of $\Pi.C$. We suggest two methods to underapproximate the set of non-trivially ∀-to-∃ reducible variables. We start with the second method. For each universal quantification $\forall p$ in Π it performs the following steps.

(1) $\Pi'_{(i)}.C'_{(i)}$ is obtained from $\Pi.C$ by weakening $\forall p$ to $\exists p$.
(2) If $\Pi'_{(i)}.C'_{(i)}$ is satisfiable, then $\forall p$ is not ∀-to-∃ reducible in $\Pi.C$ and the method moves on to the next universal quantification in Π.

(3) $\Pi'_{(iii)}.C'_{(iii)}$ is obtained from $\Pi'_{(i)}.C'_{(i)}$ by weakening a maximal set of universal to existential quantifiers in $\Pi'_{(i)}$ and by removing a maximal set of clauses without occurrences of p from $C_{(i)}$ such that the result is still unsat.
(4) $\Pi'_{(iv)}.C'_{(iv)}$ is obtained from $\Pi'_{(iii)}.C'_{(iii)}$ by removing all clauses with occurrences of p from $C'_{(iii)}$.
(5) If $\Pi'_{(iv)}.C'_{(iv)}$ is satisfiable, then $\forall p$ is non-trivially $\forall$-to-$\exists$ reducible in $\Pi.C$; otherwise, $\forall p$ is trivially or non-trivially $\forall$-to-$\exists$ reducible in $\Pi.C$.

The first method, which is cheaper but reports "trivially or non-trivially" $\forall$-to-$\exists$ reducible more often, omits step (3).

12. Implementation

We implemented the ideas presented in this paper as an extension of `DepQBF`[18] version 6.03, which we call `DepQBF-a2aecc`. Given a QBF in PCNF $\Pi.C$, `DepQBF--a2aecc` can compute an — optionally q- and c-minimally — unsatisfiable c-core, q-core, or qc-core of $\Pi.C$. Alternatively, `DepQBF-a2aecc` can act as a preprocessor to transform $\Pi.C$ into $a2aecc(\Pi.C)$. In both cases the variant of the A2AECC-transformation discussed in Section 9 can be enabled as an option. `DepQBF` supports the computation of unsatisfiable cores by permitting to place clauses in clause groups and, for unsatisfiable formulas, indicating which clause groups were used to establish unsatisfiability.[17] We utilize this to obtain an initial unsatisfiable c-core $\Pi'.C'$ of $\Pi.C$ (for c-cores) or of $a2aecc(\Pi.C)$ (for q- and qc-cores). If an unsatisfiable c-core is desired, then we output $\Pi'.C'$ directly. If an unsatisfiable q-core or qc-core is desired, then we translate $\Pi'.C'$ back into an unsatisfiable q- or qc-core of $\Pi.C$ according to Theorem 8.1. If, in addition, a user requests a minimally unsatisfiable core, then we employ a deletion-based algorithm[61] with CSR (clause set refinement)[62] to minimize C'; we use the `DepQBF` API[17] to dis- or enable clause groups as needed in the repeated checks for satisfiability. Because of Theorem 5.1 we first minimize the clauses introduced by the A2AECC-transformation and only after that the clauses of C; optionally, during the first phase of minimization, we also restrict CSR to the clauses introduced by the A2AECC-transformation.

13. Case Studies

In this section we discuss four case studies from `QBFLIB`,[20] which we encountered during our experimental evaluation, that illustrate how the weakening of universal quantifiers to existential quantifiers in unsatisfiable cores can cause improved understanding of unsatisfiable QBF.

Winning strategies in two-player games. The `Gent-Rowley` suite models variants of the well-known Connect-4 game.[1] The parameters of an instance include the length of a winning line and the width and the height of the game board. A subset of instances model whether player 1 has a strategy to enforce a draw. Some of these instances with winning lines of length 2 on boards with at least two rows and two columns have unsatisfiable cores in which all universal quantifiers have been weakened to existential quantifiers. This means that player 1 would not be able to enforce a draw even if she were given full control over the moves of player 2. This is clear, because eventually two pieces of the same color will end up next to each other, either horizontally, vertically, or diagonally, and, hence, form a winning line for one of the two players. The corresponding unsatisfiable cores confirmed this.

Moreover, for instances with longer winning lines and on larger boards we obtained unsatisfiable cores in which only one universal quantifier remained. This seemed odd, as larger board sizes give rise to larger maximal numbers of moves, which in turn induce larger numbers of universal quantifiers in the input formula. Inspection of the unsatisfiable cores helped to understand that in the model of the game in Ref. 1 player 2 can prevent a draw if she plays an illegal move at her first turn, thereby ending the game with a win for player 1. This seems to be an aspect of this model of the game that a user of this model of the game should be aware of.

Finally, another subset of instances model whether player 2 has a strategy to win. Again, we obtained an unsatisfiable core in which only one universal quantifier remained. The unsatisfiable core revealed that player 1 caused the unsatisfiability by playing an illegal first move; while this should imply a win for player 2, that is ruled out by Eqn. 12. in Ref. 1. This raises the question of whether this part of the model of the game is indeed as intended.

Conformant planning. The `Rintanen/Sorting_networks` family contains instances, parameterized by d and l, which are satisfiable iff there exists a sorting network of depth d that, for all input sequences of length l, produces a sorted output sequence.[19] The instance for depth 3 and input sequences of length 6 is unsatisfiable. In the resulting unsatisfiable core the universal quantification over the first number of the input sequence has been turned into an existential quantification. This means that there would be no such sorting network even if the "planner" were able to freely choose the first number of the input sequence. This is an interesting fact to know in itself; moreover, it implies that there is already no sorting network of depth 3 for input sequences of length 5.

Satisfiability of modal logic K. The `Pan` suite of examples encodes formulas in the modal logic K as equisatisfiable QBF.[3] In the QBF encoding universal quantification ranges over the values of an index variable. Each value of the index variable activates a different part of the encoding, which corresponds to a different $\Diamond$-subformula of the K formula. This avoids the repetition of certain subformulas in the resulting QBF, which is needed to keep the complexity of the translation from K to QBF polynomial instead of exponential.[3] In an unsatisfiable core that we obtained for the instance `k_branch_p-2` a universal quantifier had been turned into an existential quantifier. This signals that it is sufficient to retain either one of two $\Diamond$-subformulas in the input formula to obtain unsatisfiability.

Table 1. Statistics about structural properties of the benchmark set.

	min.	1st quartile	median	3rd quartile	max.	mean
number of ∀	0	19.25	90	213	55,022	325.8
number of ∃	1	477.5	2,239	7,215	2,202,774	18,980.3
alternation depth	1	2	3	6	1,141	17.7
number of clauses	1	2,000	9,126.5	29,861.75	5,534,890	80,410.1
max. variable index	1	558.25	2,556.5	8,556.75	2,202,778	33,383.3

Answer set programming. The `Faber-Leone-Maratea-Ricca/Strategic_-Companies` family of examples encodes the question of whether two selected companies from a set of companies are strategic.[63] Instance `x25.17` turned out to be unsatisfiable. This means that the two companies under consideration are indeed strategic. In the corresponding unsatisfiable core the universal quantifier for the variable for a third company had been weakened to an existential quantifier. This indicates that that company is strategic as well.

14. Experimental Evaluation

14.1. *Setup and benchmarks*

We used a single machine equipped with a Xeon E3-1245v5 CPU, 32 GB of RAM, and Ubuntu 16.04 as the operating system. We limited run time and memory usage to 300 s and 8 GB. Our set of benchmarks consists of 5342 instances from `QBFLIB`.[20] We chose instances randomly from the set of all `QBFLIB` instances such that the same number of instances was taken from each benchmark suite (subject to availability) and, recursively within benchmark suites, the same number of instances from each subfamily. As a result, if a benchmark suite had fewer than 193 available instances, then we included all instances; otherwise, we used at least 193 instances. We did not employ any other selection criteria. In Table 1 we show some statistics about structural properties of our benchmark set. We did not apply a preprocessor such as `bloqqer`[58] to the instances, because we were interested in determining the potential for weakening universal to existential quantifiers in the instances as they were originally included in `QBFLIB`. Our implementation, our experimental data, an extended version of this paper including, e.g., a significantly more detailed experimental evaluation, and more tables and plots (some of which are partitioned by benchmark family or structural properties such as number of universal quantifications or alternation depth) are available from `http://schuppan.de/viktor/ijait20/`.

14.2. *Extracting unsatisfiable cores*

In our first set of experiments we extracted unsatisfiable cores with `DepQBF-a2aecc` from the 2528 instances that were found to be unsatisfiable.

In the columns labeled “q” to “q minsepcsr” of Table 2 we show how many universal quantifiers could be weakened to existential quantifiers as a share of the

Table 2. Number of instances whose number of weakened universal quantifiers in the unsatisfiable core (resp. non-trivially ∀-to-∃ reducible universal quantifiers) divided by the number of universal quantifiers in the original formula lies within a range. For reference, in line 2 the corresponding numbers for unsat. c-cores and c-minimally unsat. c-cores are 1830 and 1682.

	q	q min	qc	qc min	qc minsepcsr	enuma2e1	enuma2e2
solved	1649	1139	1551	1441	927	986	657
no ∀ in input	21	21	21	21	21	21	21
0	465	195	1528	1356	580	831	385
[0.002, 0.004[	1	—	1	5	1	—	1
[0.004, 0.006[	4	—	—	3	3	1	2
[0.006, 0.008[	4	—	—	37	9	—	9
[0.008, 0.02[	13	5	1	10	42	5	29
[0.02, 0.04[	46	5	—	5	101	9	38
[0.04, 0.06[	11	6	—	2	37	8	28
[0.06, 0.08[	8	—	—	1	9	1	22
[0.08, 0.2[	95	37	—	1	20	21	36
[0.2, 0.4[	159	82	—	—	44	15	19
[0.4, 0.6[	305	250	—	—	27	4	7
[0.6, 0.8[	96	96	—	—	22	10	10
[0.8, 1[	291	266	—	—	11	7	5
1	130	176	—	—	—	53	45

number of universal quantifiers in the original formula. In line 1 we state the kind of unsatisfiable cores that were extracted. "q" (resp. "qc") stands for unsatisfiable q-cores (resp. qc-cores), "min" stands for q-minimality for unsatisfiable q-cores and for both q- and c-minimality for unsatisfiable qc-cores, and "minsepcsr" stands for q- and c-minimality with separate CSR. In line 2 we list how many instances of each kind were solved. In line 3 we provide the number of solved instances that had no universal quantifiers to begin with. In the remaining lines we show for how many instances we obtained unsatisfiable q- or qc-cores whose share of weakened universal quantifiers lies in the range that is stated in column 1. Notice that the numerator of this fraction includes only weakened universal quantifications whose variables still occur in some clause of the matrix of the unsatisfiable core, because our implementation removes quantifications from the prefix whose variables have no occurrences in the matrix during postprocessing. For example, for q- and c-minimally unsatisfiable qc-cores with separate CSR we found 22 instances such that the number of weakened universal quantifiers in the unsatisfiable core divided by the number of universal quantifiers in the original formula is between 0.6 (inclusive) and 0.8 (exclusive). For a number of instances we obtained unsatisfiable q-cores in which the share of universal quantifiers that had been weakened to existential quantifiers is quite large; we remark, though, that these cores need not be c-minimally unsatisfiable (cf. Section 10). Finding an unsatisfiable qc-core in which a significant share of universal quantifiers has been weakened to existential quantifiers apparently requires to enable minimization with separate CSR. Then also here we found instances in which a fairly large share of universal quantifiers had been weakened to existential quantifiers (these cores are c-minimally unsatisfiable). Un-

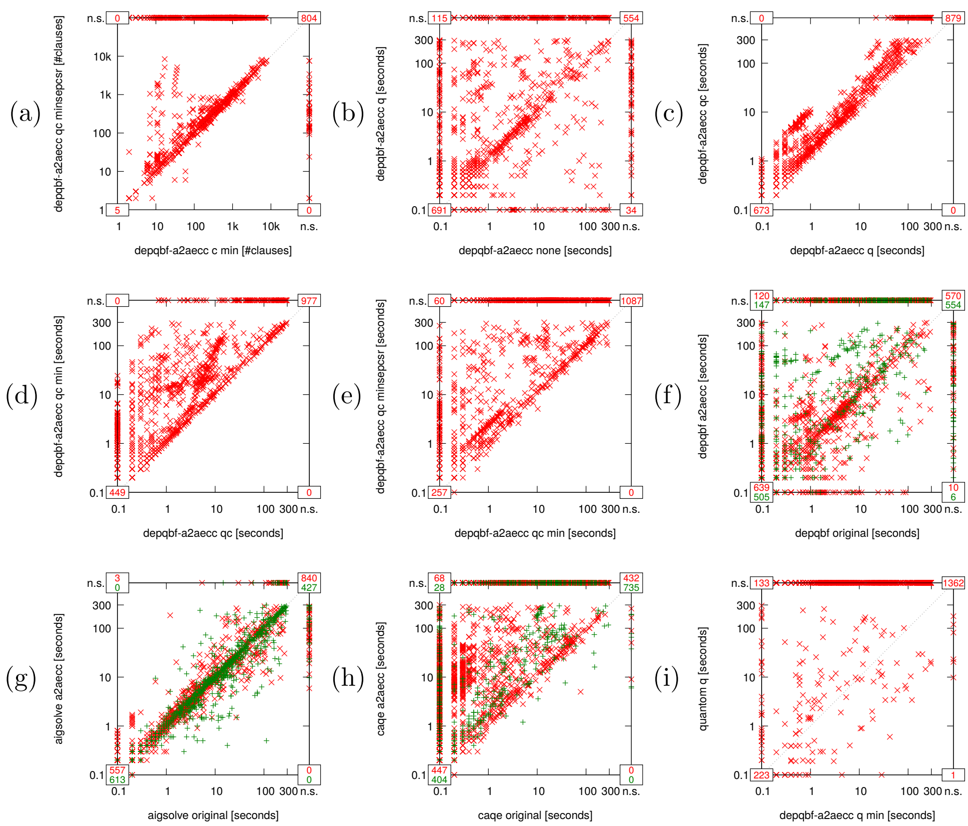

Fig. 1. (a) Comparing the sizes of unsatisfiable cores [number of clauses]: x-axis: c-minimally unsatisfiable c-cores, y-axis: q- and c-minimally unsatisfiable qc-cores with separate CSR. (b)–(e) Comparing the run times for extracting unsatisfiable cores [seconds]: (b) x-axis: no unsatisfiable cores, y-axis: unsatisfiable q-cores; (c) x-axis: unsatisfiable q-cores, y-axis: unsatisfiable qc-cores; (d) x-axis: unsatisfiable qc-cores, y-axis: q- and c-minimally unsatisfiable qc-cores; (e) x-axis: q- and c-minimally unsatisfiable qc-cores, y-axis: q- and c-minimally unsatisfiable qc-cores with separate CSR. (f)–(h): Comparing the run times for solving the A2AECC-transformed (y-axis) versus the original (x-axis) instances [seconds]: (f) `DepQBF`; (g) `AIGSolve`; (h) `CAQE`. (i) Comparing the run times for finding q-minimally unsatisfiable q-cores with `DepQBF-a2aecc` (x-axis) with finding minimum cardinality sets of universal quantifiers whose weakening to existential quantifiers results in satisfiability (y-axis) [seconds]. Red diagonal crosses represent unsatisfiable and green horizontal-/vertical crosses represent satisfiable instances. "n.s." abbreviates not solved. Scatter plots may be subject to overplotting, when different benchmark instances are assigned the same x- and y-coordinates and cannot be distinguished in the plot. In our case the effect tends to be worst in the corners of the plot. We therefore replace the crosses *in the corners* with the numbers of instances exhibiting the corresponding x- and y-coordinates. When two values are given, then the red, upper value stands for unsatisfiable and the green, lower value stands for satisfiable instances. For example, in (b) there are 115 instances in the upper left corner, i.e. they were solved in 0.1 seconds by the method on the x-axis and remained unsolved by the method on the y-axis.

surprisingly, our data show that for unsatisfiable q-cores, q-minimally unsatisfiable q-cores, and q- and c-minimally unsatisfiable qc-cores with separate CSR we tend to obtain higher numbers of weakened universal quantifiers from original instances with higher numbers of universal quantifiers.

In the columns labeled "enuma2e1" and "enuma2e2" of Table 2 we show how many universal quantifiers were found to be non-trivially ∀-to-∃ reducible relative to the number of universal quantifiers in the original formula, where "enuma2e1" refers to the first and "enuma2e2" to the second method from Section 11. Inspection of our data shows that, as expected, the second method finds more non-trivially ∀-to-∃ reducible quantifiers than the first method.

In Figure 1 (a) we compare the sizes of q- and c-minimally unsatisfiable qc-cores obtained with separate CSR with the sizes of c-minimally unsatisfiable c-cores. We find that, for the same input formula, the former can be significantly larger than the latter. This is not surprising: turning a universal quantification into an existential quantification amounts to turning a conjunction into a disjunction, and establishing the unsatisfiability of a disjunction requires both disjuncts while establishing the unsatisfiability of a conjunction requires only one conjunct. Keep in mind (cf. Section 13) that already the mere fact that a certain universal quantifier has been weakened to an existential quantifier may convey valuable information. Our data also show that large increases in unsatisfiable core size tend to coincide with large numbers of weakened universal quantifiers, which is expected.

In Figure 1 (b)–(e) we show the run time overhead that is incurred by each step when going from no unsatisfiable core extraction via unsatisfiable q-core extraction, unsatisfiable qc-core extraction, and q- and c-minimally unsatisfiable qc-core extraction to q- and c-minimally unsatisfiable qc-core extraction with separate CSR. The relation of the run times between no unsatisfiable core extraction and unsatisfiable q-core extraction is quite variable (b). Moving from unsatisfiable q-core extraction to unsatisfiable qc-core extraction incurs only a moderate overhead (c). In contrast, additionally requiring q- and c-minimality (d) and, on top of that, using separate CSR (e) are quite costly. Notice that (b) involves solving the original versus solving the A2AECC-transformed instance; although the A2AECC-transformation essentially increases the alternation depth by twice the number of universal quantifiers minus the alternation depth in the original instance, we did not observe a clear corresponding dependence of the overhead in (b).

We repeated the above experiments with set-inner instead of list semantics. As expected, when using set-inner semantics, often fewer universal quantifiers were weakened to existential quantifiers. However, despite the potentially lower alternation depth of the transformed formula with set-inner semantics, we did not find an unambiguous performance advantage for set-inner semantics.

14.3. *Solving A2AECC-transformed versus original instances*

In our second set of experiments we used `DepQBF-a2aecc` as a preprocessor. We then ran the following QBF solvers on both the original and the A2AECC-transformed instances: `DepQBF` v. 6.03,[18] `AIGSolve`,[64] `CAQE` v. qbfeval 2017,[65] `GhostQ` v. 2017-

07-26,[66] `QESTO` v. 1.0,[67] and `RAReQS` v. 1.1.[66] This allows to partially evaluate our proposed methodology beyond `DepQBF-a2aecc`. In Figure 1 (f)–(h) we compare the run times for solving the A2AECC-transformed versus the original instances with `DepQBF` (f), `AIGSolve` (g), and `CAQE` (h). We omit the plot for `GhostQ`, which is similar to that for `AIGSolve`, and the plots for `QESTO` and `RAReQS`, which are largely similar to that for `CAQE`. We observe that (i) the A2AECC-transformed instances can be solved in many cases, (ii) the overhead for solving the A2AECC-transformed instances depends on the solver, and (iii) some of the A2AECC-transformed instances are solved faster than the original instances by some solvers. For `CAQE`, `QESTO`, and, to a lesser extent, `RAReQS` our data indicate a dependence of the overhead of solving the A2AECC-transformed versus the original instance on twice the number of universal quantifiers minus the alternation depth in the original instance.

We repeated the experiments with set-inner instead of list semantics. Only for `RAReQS` set-inner semantics resulted in a fairly unambiguous performance advantage. `AIGSolve` and `GhostQ` were affected comparatively little by the choice of semantics, while for the remaining solvers no clear picture arose.

14.4. *quantom*

In our last set of experiments we performed a preliminary comparison of `DepQBF-a2aecc` with `quantom`, which, despite its differences, is the most closely related tool. We used `quantom` to obtain a minimum cardinality set of universal quantifiers such that the weakening of all quantifiers in this set from universal to existential makes the QBF under consideration satisfiable. We then compared the performance of `quantom` on this task with the performance of `DepQBF-a2aecc` on extracting a q-minimally unsatisfiable q-core. Note that this compares finding minimum cardinality diagnoses with finding minimal unsatisfiable cores, which are quite different tasks! In Figure 1 (i) we show the results. `DepQBF-a2aecc` was faster on 835 instances, while `quantom` was faster on 81 instances, with some large differences both ways.

15. Conclusions

We introduced a notion of q- and qc-cores for QBF in PCNF that not only removes clauses but also weakens universal quantifiers to existential quantifiers. We showed that this leads to unsatisfiable cores and, thus, explanations, diagnoses, and repairs of unsatisfiability that cannot be obtained from traditional unsatisfiable c-cores. We used the A2AECC-transformation on QBF in PCNF to cast q- and qc-cores as c-cores. We illustrated with case studies that helpful additional information can be learned from unsatisfiable qc-cores. We demonstrated through an experimental evaluation that our approach can successfully compute unsatisfiable q- and qc-cores on examples from `QBFLIB`. Potential future work includes analyzing how the A2AECC-transformation affects different solvers, finding a method to obtain unsatisfiable q- and qc-cores without using the A2AECC-transformation such as directly from a run of the solver, and extending this work to logics with quantification beyond QBF.

Acknowledgments

I am grateful to: Sven Reimer and his coauthors for discussing their work[22] and for providing me with `quantom`;[24] Alessandro Cimatti for mentioning that proofs can be used to compare formulas; Barry O'Sullivan for pointing out related work[28,31] in QCSP; and the anonymous reviewers for their suggestions on how to improve this paper.

References

1. I. P. Gent and A. G. D. Rowley, Encoding Connect-4 Using Quantified Boolean Formulae, in *ModRef* 2003.
2. J. Rintanen, Constructing Conditional Plans by a Theorem-Prover, *J. Artif. Intell. Res.* **10** (1999).
3. G. Pan and M. Y. Vardi, Optimizing a BDD-Based Modal Solver, in *CADE LNCS* **2741**, (Springer, 2003).
4. U. Egly, T. Eiter, H. Tompits and S. Woltran, Solving Advanced Reasoning Tasks Using Quantified Boolean Formulas, in *AAAI* (AAAI Press / The MIT Press, 2000).
5. A. Ayari and D. A. Basin, Bounded Model Construction for Monadic Second-Order Logics, in *CAV LNCS* **1855**, (Springer, 2000).
6. E. Giunchiglia, P. Marin and M. Narizzano, Reasoning with Quantified Boolean Formulas, in *Handbook of Satisfiability* (IOS Press, 2009)
7. J. W. Chinneck and E. W. Dravnieks, Locating Minimal Infeasible Constraint Sets in Linear Programs, *INFORMS Journal on Computing* **3**(2) (1991).
8. R. Bruni and A. Sassano, Restoring Satisfiability or Maintaining Unsatisfiability by finding small Unsatisfiable Subformulae, *Electronic Notes in Discrete Mathematics* **9** (2001).
9. S. Schlobach and R. Cornet, Non-Standard Reasoning Services for the Debugging of Description Logic Terminologies, in *IJCAI* (Morgan Kaufmann, 2003).
10. Y. Yu and S. Malik, Validating the result of a Quantified Boolean Formula (QBF) solver: theory and practice, in *ASP-DAC* (ACM Press, 2005).
11. V. Schuppan, Towards a notion of unsatisfiable and unrealizable cores for LTL, *Sci. Comput. Program.* **77**(7-8) (2012).
12. O. Kullmann, I. Lynce and J. Marques-Silva, Categorisation of Clauses in Conjunctive Normal Forms: Minimally Unsatisfiable Sub-clause-sets and the Lean Kernel, in *SAT LNCS* **4121**, (Springer, 2006).
13. R. Reiter, A Theory of Diagnosis from First Principles, *Artif. Intell.* **32**(1) (1987).
14. S. Schlobach, Debugging and Semantic Clarification by Pinpointing, in *ESWC LNCS* **3532**, (Springer, 2005).
15. H. Kleine Büning and X. Zhao, Minimal False Quantified Boolean Formulas, in *SAT LNCS* **4121**, (Springer, 2006).
16. A. Ignatiev, M. Janota and J. Marques-Silva, Quantified Maximum Satisfiability: A Core-Guided Approach, in *SAT LNCS* **7962**, (Springer, 2013).
17. F. Lonsing and U. Egly, Incrementally Computing Minimal Unsatisfiable Cores of QBFs via a Clause Group Solver API, in *SAT LNCS* **9340**, (Springer, 2015).
18. F. Lonsing and U. Egly, DepQBF 6.0: A Search-Based QBF Solver Beyond Traditional QCDCL, in *CADE LNCS* **10395**, (Springer, 2017).
19. J. Rintanen, Asymptotically Optimal Encodings of Conformant Planning in QBF, in *AAAI* (AAAI Press, 2007).
20. E. Giunchiglia, M. Narizzano, L. Pulina and A. Tacchella, Quantified Boolean Formulas satisfiability library (QBFLIB) `http://www.qbflib.org/`.

21. V. Schuppan, Enhanced Unsatisfiable Cores for QBF: Weakening Universal to Existential Quantifiers, in *ICTAI* (IEEE, 2018).
22. S. Reimer, M. Sauer, P. Marin and B. Becker, QBF with Soft Variables, *ECEASST* **70** (2014).
23. A. Condon, J. Feigenbaum, C. Lund and P. W. Shor, Probabilistically checkable debate systems and approximation algorithms for PSPACE-hard functions, in *STOC* (ACM, 1993).
24. S. Reimer, F. Pigorsch, C. Scholl and B. Becker, Enhanced Integration of QBF Solving Techniques, in *MBMV* (Verlag Dr. Kovac, 2012).
25. F. Lonsing and A. Biere, Failed Literal Detection for QBF, in *SAT LNCS* **6695**, (Springer, 2011).
26. F. Lonsing, U. Egly and M. Seidl, Q-Resolution with Generalized Axioms, in *SAT LNCS* **9710**, (Springer, 2016).
27. R. Brummayer, F. Lonsing and A. Biere, Automated Testing and Debugging of SAT and QBF Solvers, in *SAT LNCS* **6175**, (Springer, 2010).
28. A. Ferguson and B. O'Sullivan, Quantified Constraint Satisfaction Problems: From Relaxations to Explanations, in *IJCAI* 2007.
29. U. Junker, QuickXplain: Conflict Detection for Arbitrary Constraint Propagation Algorithms, in *CONS* 2001.
30. L. Bordeaux, Y. Hamadi and L. Zhang, Propositional Satisfiability and Constraint Programming: A comparative survey, *ACM Comput. Surv.* **38**(4) (2006).
31. D. Mehta, B. O'Sullivan and L. Quesada, Extending the Notion of Preferred Explanations for Quantified Constraint Satisfaction Problems, in *ICTAC LNCS* **9399**, (Springer, 2015).
32. L. Bordeaux, M. Cadoli and T. Mancini, Generalizing consistency and other constraint properties to quantified constraints, *ACM Trans. Comput. Log.* **10**(3) (2009).
33. I. Shlyakhter, R. Seater, D. Jackson, M. Sridharan and M. Taghdiri, Debugging Overconstrained Declarative Models Using Unsatisfiable Cores, in *ASE* (IEEE Computer Society, 2003).
34. V. Schuppan, Extracting unsatisfiable cores for LTL via temporal resolution, *Acta Inf.* **53**(3) (2016).
35. V. Schuppan, Enhancing unsatisfiable cores for LTL with information on temporal relevance, *Theor. Comput. Sci.* **655, Part B** (2016).
36. É. Grégoire, B. Mazure and C. Piette, MUST: Provide a Finer-Grained Explanation of Unsatisfiability, in *CP LNCS* **4741**, (Springer, 2007).
37. S. Grimm and J. Wissmann, Elimination of Redundancy in Ontologies, in *ESWC LNCS* **6643**, (Springer, 2011).
38. R. Armoni, L. Fix, A. Flaisher, O. Grumberg, N. Piterman, A. Tiemeyer and M. Y. Vardi, Enhanced Vacuity Detection in Linear Temporal Logic, in *CAV LNCS* **2725**, (Springer, 2003).
39. A. Gurfinkel and M. Chechik, How Vacuous Is Vacuous?, in *TACAS LNCS* **2988**, (Springer, 2004).
40. A. Kalyanpur, B. Parsia and B. C. Grau, Beyond Asserted Axioms: Fine-Grain Justifications for OWL-DL Entailments, in *DL CEUR Workshop Proceedings* **189**, (CEUR-WS.org, 2006).
41. S. C. Lam, J. Z. Pan, D. H. Sleeman and W. W. Vasconcelos, A Fine-Grained Approach to Resolving Unsatisfiable Ontologies, in *WI* (IEEE Computer Society, 2006).
42. M. Horridge, B. Parsia and U. Sattler, Laconic and Precise Justifications in OWL, in *ISWC LNCS* **5318**, (Springer, 2008).
43. I. Pill and T. Quaritsch, Behavioral Diagnosis of LTL Specifications at Operator Level,

in *IJCAI* (IJCAI/AAAI, 2013).
44. A. Kalyanpur, B. Parsia, E. Sirin and B. C. Grau, Repairing Unsatisfiable Concepts in OWL Ontologies, in *ESWC LNCS* **4011**, (Springer, 2006).
45. J. Du, G. Qi and X. Fu, A Practical Fine-grained Approach to Resolving Incoherent OWL 2 DL Terminologies, in *CIKM* (ACM, 2014).
46. J. Marques-Silva and M. Janota, Computing Minimal Sets on Propositional Formulae I: Problems & Reductions, *CoRR* **abs/1402.3011** (2014).
47. H. Kleine Büning and U. Bubeck, Theory of Quantified Boolean Formulas, in *Handbook of Satisfiability* (IOS Press, 2009)
48. L. J. Stockmeyer and A. R. Meyer, Word Problems Requiring Exponential Time: Preliminary Report, in *STOC* (ACM, 1973).
49. L. J. Stockmeyer, The Polynomial-Time Hierarchy, *Theor. Comput. Sci.* **3**(1) (1976).
50. C. Wrathall, Complete Sets and the Polynomial-Time Hierarchy, *Theor. Comput. Sci.* **3**(1) (1976).
51. I. Lynce and J. P. Marques-Silva, On Computing Minimum Unsatisfiable Cores, in *SAT* 2004.
52. E. Torlak, F. S. Chang and D. Jackson, Finding Minimal Unsatisfiable Cores of Declarative Specifications, in *FM LNCS* **5014**, (Springer, 2008).
53. A. V. Gelder, Contributions to the Theory of Practical Quantified Boolean Formula Solving, in *CP LNCS* **7514**, (Springer, 2012).
54. S. Coste-Marquis, H. Fargier, J. Lang, D. Le Berre and P. Marquis, Representing Policies for Quantified Boolean Formulae, in *KR* (AAAI Press, 2006).
55. N. Francez, The Granularity of Meaning in Proof-Theoretic Semantics, in *LACL LNCS* **8535**, (Springer, 2014).
56. H. Kleine Büning, M. Karpinski and A. Flögel, Resolution for Quantified Boolean Formulas, *Inf. Comput.* **117**(1) (1995).
57. J. Slaney, Set-theoretic duality: A fundamental feature of combinatorial optimisation, in *ECAI Frontiers in Artificial Intelligence and Applications* **263**, (IOS Press, 2014).
58. A. Biere, F. Lonsing and M. Seidl, Blocked Clause Elimination for QBF, in *CADE LNCS* **6803**, (Springer, 2011).
59. A. Nadel, Boosting minimal unsatisfiable core extraction, in *FMCAD* (IEEE, 2010).
60. M. H. Liffiton and K. A. Sakallah, Algorithms for Computing Minimal Unsatisfiable Subsets of Constraints, *J. Autom. Reasoning* **40**(1) (2008).
61. J. Marques-Silva, Computing Minimally Unsatisfiable Subformulas: State of the Art and Future Directions, *Multiple-Valued Logic and Soft Computing* **19**(1-3) (2012).
62. A. Belov, I. Lynce and J. Marques-Silva, Towards efficient MUS extraction, *AI Commun.* **25**(2) (2012).
63. W. Faber, N. Leone, M. Maratea and F. Ricca, Looking Back in DLV: Experiments and Comparison to QBF Solvers, in *ASP* 2007.
64. F. Pigorsch and C. Scholl, An AIG-Based QBF-solver using SAT for preprocessing, in *DAC* (ACM, 2010).
65. L. Tentrup, On Expansion and Resolution in CEGAR Based QBF Solving, in *CAV LNCS* **10427**, (Springer, 2017).
66. M. Janota, W. Klieber, J. Marques-Silva and E. M. Clarke, Solving QBF with Counterexample Guided Refinement, in *SAT LNCS* **7317**, (Springer, 2012).
67. M. Janota and J. Marques-Silva, Solving QBF by Clause Selection, in *IJCAI* (AAAI Press, 2015).