Enhanced Unsatisfiable Cores for QBF: Weakening Universal to Existential Quantifiers

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

***Abstract*—We propose an enhanced notion of unsatisfiable cores for QBF in prenex CNF that weakens universal to existential quantifiers in addition to the traditional removal of clauses. We can thus obtain unsatisfiable cores that are semantically different from those obtained by the traditional notion; this gives rise to explanations — and, via hitting set duality, diagnoses — of unsatisfiability that are not provided by traditional unsatisfiable cores. We use a source-to-source transformation on QBF that reduces the weakening of universal to existential quantifiers to the removal of clauses. This enables any tool or method that can compute unsatisfiable cores of the traditional notion to also compute unsatisfiable cores of our enhanced notion. We implement our approach in the QBF solver `DepQBF`, and we experimentally evaluate it on a subset of `QBFLIB`. Several case studies illustrate that interesting information can be learned from our enhanced notion of unsatisfiable cores.**

I. Introduction

a) Motivation and Contributions: Many important problems can be naturally encoded as quantified Boolean formulas (QBF), e.g., two-player games (e.g., [1]), variants of planning (e.g., [2]), satisfiability of modal logic K [3], and several problems in knowledge representation (e.g., [4]) and formal methods (e.g., [5]); for a more extensive list see [6]. Unsatisfiable cores have become a fundamental concept in applied logic. For example, they are commonly taken to represent causes of and serve as explanations of unsatisfiability in various logics (e.g., [7]–[11]), and they are used as building blocks to obtain advanced explanations of unsatisfiability (e.g., [12]) and to diagnose (e.g., [13]) and repair (e.g., [14]) unsatisfiability. Existing work on unsatisfiable cores for QBF in prenex conjunctive normal form (PCNF) weakens formulas by removing clauses [10], [15]–[17].

In this paper we propose an enhanced notion of unsatisfiable cores for QBF in PCNF that, in addition to removing clauses, weakens universal to existential quantifiers (Section III). Our enhanced notion of unsatisfiable cores can represent causes and lead to explanations of unsatisfiability that are different from any one that can be obtained from an unsatisfiable core of the traditional notion (Section IV). Moreover, via the well-known hitting set duality (e.g., [13]; for a generic formulation see [18]), this induces diagnoses and repairs for unsatisfiability that cannot be obtained when using the traditional notion of unsatisfiable cores. On a less rigorous, but nevertheless practically relevant note, an unsatisfiable core, in which the set of quantifiers that has been weakened from universal to existential has some unexpected characteristics, may provide the initial "hunch" to the user that something may not be quite right in the QBF under consideration. In Section V we show that if in an unsatisfiable QBF in PCNF no clause can be removed without making the result satisfiable, then also no universal quantifier can be weakened to an existential one without losing unsatisfiability. Then we extend the PSPACE-completeness result for minimally unsatisfiable cores of the traditional notion [15] to our enhanced notion (Section VI). We describe a transformation of QBF in PCNF such that weakening of universal to existential quantifiers can be performed by removing clauses in the transformed formula (Sections VII,VIII). That allows to obtain unsatisfiable cores in our enhanced notion by first applying the transformation, then using existing tools and methods to compute an unsatisfiable core by removing clauses, and finally mapping back the result to an unsatisfiable core in the enhanced notion. Next we provide some hints on how to interpret unsatisfiable cores (Section IX). We implement our approach in `DepQBF` [19] (Section X), and we experimentally evaluate it on a subset of `QBFLIB` [20] (Section XII). Using a number of case studies including two-player games [1], conformant planning [21], and satisfiability of modal logic K [3] we illustrate that interesting information can be learned from our enhanced notion of unsatisfiable cores (Section XI). Our experiments show that on instances from `QBFLIB` unsatisfiable cores of our enhanced notion can be computed and that indeed universal quantifications are weakened to existential ones.

b) Related Work: Work on improving algorithms for solving QBF has referred to weakening universal to existential quantifiers as "quantifier abstraction" [22] and "existential abstraction" [23].

`QBFDD` [24] allows quantifier manipulations when minimizing failure-inducing input.

[25] introduces the concept of soft variables, which are variables that may be placed at different positions of the prefix of a QBF subject to a preference function. The authors then define the optimization problem of finding a placement for the soft variables that maximizes the preference function while maintaining satisfiability of the resulting QBF. They use a transformation, which can be seen as a generalized version of our transformation in Section VII, to reduce their problem to a weighted partial MaxQBF problem. (We discovered our transformation independently.) They implement their approach in `quantom` [26]. Our work differs from [25] as follows. [25] searches for a still satisfiable result, while we search for a still unsatisfiable result. 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 ([16] is an example). In [25] the authors make no connection to unsatisfiable cores. [25] finds a maxim*um* solution, while we (optionally) find a minim*al* solution. [25] leaves the matrix unchanged, whereas we (optionally) also weaken the matrix.

When debugging unsatisfiable Alloy models Shlyakhter et al. [27] point out which values of bound variables are irrelevant to the unsatisfiability. For a Boolean variable p in some formula $\forall p.f[p]$ this corresponds to weakening $f[\bot/p] \wedge f[\top/p]$ to $f[\bot/p]$ or to $f[\top/p]$ — which can be achieved by removing clauses with occurrences of p of the suitable polarity and, hence, by the traditional notion —, whereas we can additionally weaken to $f[\bot/p] \vee f[\top/p]$.

Finally, our work is in the spirit of efforts investigating the aspect of granularity in unsatisfiable cores and related notions; examples include [11], [12], [28]–[34].

II. Preliminaries

We consider QBF in PCNF (e.g., [6], [35]); any QBF can be transformed into an equivalent QBF in PCNF (e.g., [35]).

We assume a set of *variables* V; variables are denoted by the letter p. The *Boolean constants* are $\bot$ (*false*) and $\top$ (*true*). *Literals* are variables, $\bot$, $\top$, or their *negations*, denoted $\neg$; we write literals as the letter l. A *clause* $(l_1 \vee \ldots \vee l_n)$ is a *disjunction* of literals, denoted by the letter c. In clauses we use *implication* $\rightarrow$ as an abbreviation as usual. A conjunctive normal form (CNF) formula $c_1 \wedge \ldots \wedge c_n$ is a *conjunction* of clauses; CNF formulas are denoted by the letter C. When convenient we view clauses as sets of literals and CNF formulas as sets of clauses. A variable p is *pure* in a CNF formula C, if it occurs only non-negated or only negated in C. $\mathbb{B} = \{0, 1\}$ is the set of Booleans. An *assignment* v for C is a mapping from V to $\mathbb{B}$. Then the evaluation of a CNF formula under an assignment and (un)satisfiability of a CNF formula are defined as usual.

$\forall$ and $\exists$ denote *universal* and *existential quantifiers*, respectively. We use the letter Q to represent quantifiers. Let $Q_1, \ldots, Q_n \in \{\forall, \exists\}$ be quantifiers, let $p_1, \ldots, p_n \in V$ be pairwise different variables, and let C be 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* with *prefix* $Q_1 p_1 \ldots Q_n p_n$ and *matrix* C. Prefixes are written as the letter Π. The *alternation depth* of a QBF in PCNF is one plus the number of alternations between $\forall$ and $\exists$ in the prefix. If $\Pi.C$ is a QBF in PCNF and $p \in V$, then $(\Pi.C)[\bot/p]$ (resp. $(\Pi.C)[\top/p]$) denotes the QBF in PCNF that is obtained from $\Pi.C$ by replacing every occurrence of p in C with $\bot$ (resp. $\top$). Satisfiability of a QBF in PCNF is then defined as follows. $\forall p \Pi.C$ is satisfiable iff $(\Pi.C)[\bot/p]$ and $(\Pi.C)[\top/p]$ are satisfiable. $\exists p \Pi.C$ is satisfiable iff $(\Pi.C)[\bot/p]$ or $(\Pi.C)[\top/p]$ are satisfiable. Deciding the satisfiability of a QBF in PCNF is PSPACE-complete [36]; the satisfiability problems for QBF in PCNF with alternation depth at most $i \in \mathbb{N}$ and either $\forall$ or $\exists$ as the first quantifier yield complete problems for the i-th level of the polynomial hierarchy Π_i^P and Σ_i^P, respectively [37], [38].

III. Enhanced Unsatisfiable Cores for QBF

In this section we add to the traditional notion of cores for QBF in PCNF (henceforth called c-cores), which are obtained by removing clauses, the notions of q-cores, which are obtained by weakening universal to existential quantifiers, and of qc-cores, which combine c-cores and q-cores. In Definition 1 we characterize c-, q-, and qc-cores. In Definitions 2 and 3 we state natural extensions of proper cores and unsatisfiable cores to q- and qc-cores. In Definition 4, we introduce quantifier-minimally unsatisfiable cores in addition to the traditional clause-minimally unsatisfiable cores. Let $\Pi.C$ be a QBF in PCNF.

Definition 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 $\Pi.C$.

Definition 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 (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 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 1: As an example consider $\Pi.C = \forall p.(p) \wedge (\neg p)$. Clearly, $\Pi.C$ is unsatisfiable. $\Pi.C$ has four c-cores $\Pi.C$, $\forall p.(p)$, $\forall p.(\neg p)$, and $\forall p.\top$. The first three are unsatisfiable c-cores, the last three are proper c-cores, and the second and third are both q- and c-minimally unsatisfiable.

$\Pi.C$ has two q-cores $\Pi.C$ and $\exists p.(p) \wedge (\neg p)$, both of which are unsatisfiable. Only $\exists p.(p) \wedge (\neg p)$ is a proper q-core and q-minimally unsatisfiable; it is also c-minimally unsatisfiable.

Any c-core or q-core is also a qc-core. $\exists p.(p)$, $\exists p.(\neg p)$, and $\exists p.\top$ are the only qc-cores of $\Pi.C$ that are both proper c-cores and proper q-cores of $\Pi.C$. However, none of them is unsatisfiable. □

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

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 argue 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 are indeed different from the ones identified by the traditional notion of unsatisfiable c-cores.

We consider $\forall p.(p) \wedge (\neg p)$ from Example 1 with q- and c-minimally unsatisfiable qc-cores $\forall p.(p)$, $\forall p.(\neg p)$, and $\exists p.(p) \wedge (\neg p)$. Clearly, the q-core $\exists p.(p) \wedge (\neg p)$ is syntactically different from the c-cores $\forall p.(p)$ and $\forall p.(\neg p)$. However, in general, syntactic differences may carry little meaning; we therefore proceed to discuss differences based on semantics.

One semantics for unsatisfiable QBF is given by tree refutations [39], [40]. A tree refutation for an unsatisfiable QBF $\Pi.C$ is a tree such that (i) its non-leaf nodes are labeled with variables in Π (the labeling of leaf nodes is irrelevant); (ii) its edges are labeled with Booleans (representing assignments to the variables that are labeling their source nodes); (iii) every node labeled with a universally quantified variable has one outgoing edge labeled with either 0 or 1; (iv) every node labeled with an existentially quantified variable has two outgoing edges labeled with 0 and 1, respectively; (v) on every path from the root to a leaf node the sequence of labels on the non-leaf nodes is identical to the sequence of variables given by the prefix Π; and (vi) on every path from the root to a leaf node the induced assignment to the variables in Π falsifies C. Intuitively, a tree refutation shows how to choose the assignment to the universally quantified variables in order to falsify $\Pi.C$.

$\forall p.(p)$ has one tree refutation with the root node labeled p and its single outgoing edge labeled 0. $\exists p.(p) \wedge (\neg p)$ has one tree refutation with the root node labeled p and two outgoing edges labeled 0 and 1. Clearly, the tree refutation for $\exists p.(p) \wedge (\neg p)$ differs from the one for $\forall p.(p)$. The two tree refutations correspond to different ways to explain the unsatisfiability of $\forall p.(p) \wedge (\neg p)$: for $\forall p.(p)$ assigning 0 to p falsifies (p); for $\exists p.(p) \wedge (\neg p)$ each assignment to p falsifies one of the clauses (p) and $(\neg p)$. The case of $\forall p.(\neg p)$ is analogous.

Let C_1, C_2 be two different matrices that have the same sets of satisfying assignments. For any prefix Π such that $\Pi.C_1$ and $\Pi.C_2$ are unsatisfiable, the sets of tree refutations for $\Pi.C_1$ and $\Pi.C_2$ are identical. I.e., tree refutations are not always sufficient to distinguish unsatisfiable cores. In that case we may turn to proof-theoretic semantics, which can be more discriminating [41]. We can, for example, compare unsatisfiable QBFs in PCNF in terms of their sets of Q-resolution proofs of unsatisfiability [42]. This, too, can be used to show a semantic difference between $\exists p.(p) \wedge (\neg p)$, $\forall p.(p)$, and $\forall p.(\neg p)$.

V. C-Minimal Unsatisfiability Implies Q-Minimal Unsatisfiability

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

Theorem 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 is an immediate consequence of the following Lemma 1. An example that the converse is not true is $\exists p \exists p' \,.\, (p) \wedge (\neg p) \wedge (p')$. ■

Lemma 1: Let

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

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 $\Pi'.C' = \exists p_l Q_{l+1} p_{l+1} \ldots Q_n p_n . C'$ is unsatisfiable. Let $\Pi'_{l+1} = Q_{l+1} p_{l+1} \ldots Q_n p_n$. Expanding $\exists p_l$ gives unsatisfiability of both $(\Pi'_{l+1}.C')[\perp/p_l]$ and $(\Pi'_{l+1}.C')[\top/p_l]$. Without limitation of generality let p_l occur non-negated in c_i. Hence, $(\Pi'_{l+1}.C'')[\top/p_l]$ is also unsatisfiable. Finally, by the definition of $\forall$, $\forall p_l \Pi'_{l+1}.C'' = \Pi''.C''$ 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

$$\Pi'_{2,\exists} = Q_2 p_2 \ldots Q_{l-1} p_{l-1} \exists p_l Q_{l+1} p_{l+1} \ldots Q_n p_n \text{ and} \\ \Pi'_{2,\forall} = Q_2 p_2 \ldots Q_{l-1} p_{l-1} \forall p_l Q_{l+1} p_{l+1} \ldots Q_n p_n .$$

Expanding $\exists p_1$ gives unsatisfiability of both $(\Pi'_{2,\exists}.C')[\perp/p_1]$ and $(\Pi'_{2,\exists}.C')[\top/p_1]$. With the inductive assumption both $(\Pi'_{2,\forall}.C'')[\perp/p_1]$, and $(\Pi'_{2,\forall}.C'')[\top/p_1]$ are unsatisfiable as well. Finally, by the definition of $\exists$, $\exists p_1 \Pi'_{2,\forall}.C'' = \Pi''.C''$ is unsatisfiable as desired. The case of $Q_1 = \forall$ is similar. ■

One might think that Theorem 1 would cast doubt on the usefulness of q- or qc-cores, as it shows that essentially any c-minimally unsatisfiable c-core is also a q-minimally unsatisfiable c-core (and qc-core). However, as shown in Section IV, qc-cores can represent causes of unsatisfiability of a formula (and give rise to corresponding explanations) that none of the c-cores represents.

VI. Complexity

Let CMF, QMF, and QCMF denote the sets of c-minimally unsatisfiable QBF in PCNF, q-minimally unsatisfiable QBF in PCNF, and CMF $\cap$ QMF, respectively. CMF has been shown to be PSPACE-complete in [15]. In this section we extend this result to QMF and QCMF.

Theorem 2: QMF and QCMF are PSPACE-complete.

Proof: Membership of QMF and QCMF in PSPACE is obvious. For PSPACE-hardness of QMF let

$$\Pi.C = Q_1 p_1 \ldots Q_m p_m . c_1 \wedge \ldots \wedge c_n$$

be a QBF in PCNF. Let $\Pi'.C$ be obtained from $\Pi.C$ by removing those universal quantifications from Π whose variables do not occur in any clause of C. Let

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

with $p'_1 \ldots p'_n$ fresh. Clearly, the size of $\Pi''.C''$ is linear in the size of $\Pi.C$. Using Theorem 1 it is a simple exercise to show that $\Pi.C$ is in CMF iff $\Pi''.C''$ is in QMF. Thus, QMF is PSPACE-hard. The proof for QCMF is similar. ■

VII. A2AECC: Q- AND QC-CORES AS C-CORES

We now describe a source-to-source transformation on QBF in PCNF that allows to cast q- and qc-cores as c-cores. 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 5 (A2AECC): Let $\Pi.C = Q_1 p_1 \ldots Q_n p_n.C$. Let $p_1', \ldots, p_n'$ 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

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

Let $\Pi.C$ be an unsatisfiable QBF in PCNF. Definition 5 allows to compute an unsatisfiable q- or qc-core $\Pi'.C'$ of $\Pi.C$ by computing an unsatisfiable c-core of $a2aecc(\Pi.C)$ as follows.

1) Let $\Pi_{a2aecc}.C_{a2aecc} = a2aecc(\Pi.C)$.
2) Compute an unsatisfiable c-core $\Pi_{a2aecc}.C'_{a2aecc}$ of $\Pi_{a2aecc}.C_{a2aecc}$.
3) Let $C' = C$ if a q-core is desired, and let $C' = C \cap C'_{a2aecc}$ if a qc-core is desired.
4) Obtain Π' from Π by replacing each quantification $Q_i p_i$ in Π with $Q_i' p_i$ where

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

The correctness of this procedure is established in Theorem 3 below. Its proof uses the following Lemma 2, which is immediate by the semantics of QBF.

Lemma 2: 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 3: 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 follow directly from Definition 1. We prove claim 3 by induction on the cardinality of P. The base case $|P| = 0$ follows by repeated application of Lemma 2. For the inductive case assume that the claim holds for any P with $|P| = n$. Now let $P = \{p_1, \ldots, p_{n+1}\}$. Let Π'' be obtained from Π by weakening $\forall p_{n+1}$ in Π to $\exists p_{n+1}$. Let $\Pi''_{a2aecc}.C''_{a2aecc} = a2aecc(\Pi''.C)$. By inductive assumption $\Pi'.C$ is satisfiable iff

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

is satisfiable. By 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, $\Pi'.C$ is satisfiable iff $\Pi''_{a2aecc}.C'_{a2aecc}$ is satisfiable. Notice that Π_{a2aecc} only differs from Π''_{a2aecc} by having $\forall p'_{n+1} \exists p_{n+1}$ in place of $\exists p_{n+1}$. Hence, as p'_{n+1} does not occur in C'_{a2aecc}, $\Pi''_{a2aecc}.C'_{a2aecc}$ is satisfiable iff $\Pi_{a2aecc}.C'_{a2aecc}$ is satisfiable. ∎

If a prefix Π has m universal quantifiers, then the alternation depth of $a2aecc(\Pi.C)$ is either $2m$ or $2m+1$. In the next Section VIII we present a variant of the transformation that does not affect alternation depth but has different semantics.

If a universally quantified variable p is pure in a matrix C, then either $(p' \rightarrow p)$ (if p occurs only non-negated in C) or $(p \rightarrow p')$ (if p occurs only negated in C) is a quantified blocked clause [43] in $a2aecc(\Pi.C)$ and can be eliminated.

If a solver for QBF in PCNF supports grouping of clauses when extracting c-cores (e.g., [44], [45]), as does `DepQBF` [17], then a clause group for each pair of clauses $(p_i \rightarrow p_i'), (p_i' \rightarrow p_i)$ introduced by Definition 5 can be used to ensure that either none or both of $(p_i \rightarrow p_i'), (p_i' \rightarrow p_i)$ are present in a c-core of $a2aecc(\Pi.C)$.

Example 2: As an example we revisit $\Pi.C = \forall p.(p) \wedge (\neg p)$ from Example 1. We have

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

The unsatisfiable c-cores

$$\Pi'.C_1' = \forall p' \exists p.(p \rightarrow p') \wedge (p' \rightarrow p) \wedge (p) \text{ and} \\ \Pi'.C_2' = \forall p' \exists p.(p \rightarrow p') \wedge (p' \rightarrow p) \wedge (\neg p)$$

of $a2aecc(\Pi.C)$ correspond to the unsatisfiable c-cores $\forall p.(p)$ and $\forall p.(\neg p)$ of $\Pi.C$. 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 $\Pi.C$. Contrary to $\Pi'.C_3'$, neither $\Pi'.C_1'$ nor

$\Pi'.C'_2$ is c-minimally unsatisfiable; however, when treating $(p \rightarrow p'), (p' \rightarrow p)$ as a clause group, then $\Pi'.C'_1$ and $\Pi'.C'_2$ are c-minimally unsatisfiable under a suitable definition of c-minimality that takes clause groups into account. □

The transformation in Definition 5, Theorem 3 is also of theoretical interest. For example, it can be used to extend the hitting set-based relationship [18] between unsatisfiable subsets of clauses and co-satisfiable subsets of clauses (complements of satisfiable subsets [18], i.e., diagnoses [13] and repairs [14]) to a relationship between unsatisfiable q- or qc-cores and suitably defined "co-satisfiable q- or qc-cores" of QBF in PCNF. The latter induce an enhanced notion of diagnosis and repair for QBF in PCNF that diagnoses and repairs unsatisfiable QBF not only by removal of clauses but also by weakening of universal to existential quantifiers.

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

In this section we discuss a variant of the A2AECC transformation that avoids the increase in alternation depth when going from $\Pi.C$ to $a2aecc(\Pi.C)$, but underapproximates the set of universal quantifiers that can be weakened to existential ones in an unsatisfiable q- or qc-core of $\Pi.C$.

Let $\Pi.C$ be a QBF in PCNF with n universal quantifiers and alternation depth m. Let $\forall p_{i,1} \ldots \forall p_{i,n_i}$ be a maximal sequence, called a *block*, of universal quantifications in Π. Definition 5 turns this block into $\forall p'_{i,1} \exists p_{i,1} \ldots \forall p'_{i,n_i} \exists p_{i,n_i}$. Overall, this increases the alternation depth of $a2aecc(\Pi.C)$ compared to $\Pi.C$ by $2 \cdot n - m$ ($+1$, if Π starts with $\exists$).

Consider a variant of Definition 5, denoted $a2aecc'$, that instead turns each block of universal quantifications $\forall p_{i,1} \ldots \forall p_{i,n_i}$ into $\forall p'_{i,1} \ldots \forall p'_{i,n_i} \exists p_{i,1} \ldots \exists p_{i,n_i}$. Now the increase in alternation depth from $\Pi.C$ to $a2aecc'(\Pi.C)$ is at most 1. Moreover, by considering the respective tree refutations (see Section IV), it is easy to see that $a2aecc(\Pi.C)$ is unsatisfiable iff $a2aecc'(\Pi.C)$ is unsatisfiable. As shown in Theorem 3, removing $(p_{i,i'} \rightarrow p'_{i,i'}) \wedge (p'_{i,i'} \rightarrow p_{i,i'})$ 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 $\Pi.C$. In contrast, it is straightforward to prove that removing $(p_{i,i'} \rightarrow p'_{i,i'}) \wedge (p'_{i,i'} \rightarrow p_{i,i'})$ 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 $\forall p_{i,1} \ldots \forall p_{i,i'-1} \forall p_{i,i'+1} \ldots \forall p_{i,n_i} \exists p_{i,i'}$ in $\Pi.C$.

By the semantics of QBF the unsatisfiability of a c-core of $a2aecc'(\Pi.C)$ implies the unsatisfiability of the corresponding c-core of $a2aecc(\Pi.C)$. For an example that the converse is not true consider $\Pi.C = \forall p_1 \forall p_2.(p_1 \rightarrow p_2) \wedge (p_2 \rightarrow p_1)$. Weakening $\forall p_1$ to $\exists p_1$ in $\Pi.C$ results in $\exists p_1 \forall p_2.(p_1 \rightarrow p_2) \wedge (p_2 \rightarrow p_1)$, which is unsatisfiable. Correspondingly, in line with Theorem 3, removing $(p_1 \rightarrow p'_1) \wedge (p'_1 \rightarrow p_1)$ from $a2aecc(\Pi.C)$ yields the unsatisfiable

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

On the other hand, removing $(p_1 \rightarrow p'_1) \wedge (p'_1 \rightarrow p_1)$ from $a2aecc'(\Pi.C)$ leads to

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

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

We finally discuss a different perspective on the semantics of $a2aecc'$. $a2aecc$ considers the positions of quantifications within a quantifier block as fixed, i.e., a block of universal quantifications is treated as a *list* of quantifications. However, the semantics of QBF allows to arbitrarily shuffle the quantifications within a quantifier block without affecting the satisfiability of the resulting QBF. Hence, a quantifier block can also be seen as a *set* of quantifications. In the light of that, $a2aecc'$ can be interpreted as employing the set semantics of a quantifier block and push the universal quantifications that have been weakened to existential ones to the right of their quantifier block (i.e., towards the inside of the QBF). We call the semantics obtained when using $a2aecc$ *list semantics* and the semantics obtained when using $a2aecc'$ *set-inner semantics*. List semantics takes a very conservative approach in that it assigns maximal meaning to the order of the quantifications in a quantifier block, whereas set-inner semantics is very relaxed and assigns no meaning to the order of quantifications in a quantifier block at all. Keep in mind that, while — as mentioned above — shuffling quantifications inside a quantifier block is a satisfiability-preserving operation, as shown by the example in the previous paragraph weakening universal quantifications to existential ones is not the same in list and in set-inner semantics.

IX. Interpreting Unsatisfiable Q- and QC-Cores

We now explain that the weakening of a universal to an existential quantifier in an unsatisfiable core may have different reasons and that it is easier to judge the significance of a weakening in an unsatisfiable core if the core is c-minimal.

Let $\Pi.C$ be an unsatisfiable QBF in PCNF and consider an unsatisfiable q- or qc-core $\Pi'.C'$ of $\Pi.C$. Assume that some $\forall p$ in Π has been weakened to $\exists p$ in Π'. Let C'' be a subset of C' such that $\Pi'.C''$ is c-minimally unsatisfiable (such C'' obviously exists). Distinguish two cases. First, assume that p occurs in some clause c in C''. Then there is a cause of the unsatisfiability of $\Pi.C$ that requires c, including its occurrence of p, but needs p to be only existentially quantified (as it is in Π') rather than universally quantified (as it is in Π). Second, assume that there is no such clause. Then the weakening of $\forall p$ to $\exists p$ in Π' is due to the fact that the unsatisfiability of $\Pi.C$ does not require any clause that contains p or $\neg p$.

Notice that in a q- or qc-core that is unsatisfiable but not c-minimal both cases may occur simultaneously for different choices of C''. Hence, the fact that $\forall p$ has been weakened to $\exists p$ in a non-c-minimally unsatisfiable q- or qc-core $\Pi'.C'$ of $\Pi.C$ should be interpreted with some care. Moreover, if $\forall p$ has been weakened to $\exists p$ in a c-minimally unsatisfiable q- or qc-core $\Pi'.C'$, then it should be checked whether C' contains p or not (if not, our implementation removes $\exists p$ from Π' during postprocessing).

Example 3: As an example consider

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

and a (non-c-minimally) unsatisfiable qc-core of $\Pi.C$

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

Inspection of $\Pi'.C'$ shows that its unsatisfiability is caused by $\exists p_1 \forall p_2.(p_1 \rightarrow p_2) \wedge (p_2 \rightarrow p_1)$, and that for its unsatisfiability it is sufficient for p_1 to be existentially quantified. Hence, the weakening of $\forall p_1$ to $\exists p_1$ in $\Pi'.C'$ provides useful additional information about the unsatisfiability of $\Pi.C$. On the other hand, $\exists p_3 \exists p_4.(p_3 \rightarrow p_4)$ does not contribute to the unsatisfiability of $\Pi'.C'$. Hence, the fact that p_3 is existentially quantified in $\Pi'.C'$ provides little information about the unsatisfiability of $\Pi.C$. $\Pi'.C'$ has a single c-minimally unsatisfiable c-core $\Pi'.C'' = \exists p_1 \forall p_2 \exists p_3 \exists p_4.(p_1 \rightarrow p_2) \wedge (p_2 \rightarrow p_1)$. Remember that in a c-minimally unsatisfiable core every clause is required for unsatisfiability. As we can see, p_1 occurs in the matrix C'', while p_3 does not. □

X. Implementation

We implemented our ideas in `DepQBF` [19] version 6.03; we call our version `DepQBF-a2aecc`. `DepQBF-a2aecc` takes a QBF in PCNF $\Pi.C$ as input. `DepQBF-a2aecc` can either be used as a preprocessor to obtain $a2aecc(\Pi.C)$, or it can compute — optionally q- and c-minimally — unsatisfiable c-cores, q-cores, or qc-cores of $\Pi.C$. `DepQBF` allows to declare clause groups and, if a formula is found unsatisfiable, to obtain the clause groups used to establish unsatisfiability [17]. We use 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). For c-cores $\Pi'.C'$ can be output directly. For q-cores and qc-cores Theorem 3 is applied to translate $\Pi'.C'$ back into a q- or qc-core of $\Pi.C$. If minimality is desired, then C' is minimized using a deletion-based algorithm (e.g., [46]) with clause set refinement (CSR) (e.g., [47]). Because of Theorem 1 minimization is first applied to the clauses introduced by Definition 5 and then to the clauses from C; optionally, CSR can also be restricted to be applied to the clauses introduced by Definition 5 during the first phase of minimization.

XI. Case Studies

In this section we discuss some case studies, which we encountered during our experimental evaluation, that illustrate how the weakening of universal to existential quantifiers in unsatisfiable cores can trigger improved understanding of unsatisfiable QBFs. The examples are taken from `QBFLIB` [20].

a) Winning Strategies in Two-Player Games: The `Gent-Rowley` suite models variants of the well-known Connect-4 game that are parameterized by the length of a winning line and the width and height of the game board [1]. Some instances model whether player 1 can enforce a draw. For some of these instances, with winning lines of length 2 on boards with at least two rows and two columns, there exists an unsatisfiable core in which all universal quantifiers have been turned into existential ones. I.e., even if player 1 had full control over the moves of player 2, she could not enforce a draw. This is clear, because eventually there must be a winning line of length 2 for one of the two players, which is confirmed by the corresponding unsatisfiable core.

Moreover, for instances with longer winning lines and on larger boards, we obtained unsatisfiable cores with only a single universal quantifier left, which seemed odd (the number of universal quantifiers in the input formula grows with the maximal number of moves, i.e., the board size). Upon inspection of the unsatisfiable cores it turned out that the game is modeled in such a way that player 2 can spoil a draw by playing an illegal move at her first turn, thus forcing a win of player 1. This seems to be a fact that a user of the model in [1] should be aware of.

Finally, other instances model whether player 2 can enforce a win. Again, we obtained an unsatisfiable core with only one universal quantifier left. The core showed that the unsatisfiability was caused by player 1 playing an illegal first move, which should imply a win for player 2; this, however, is forbidden by Eqn. 12. in [1]. This seems to warrant an investigation of whether this way of modeling the game is indeed as intended.

b) Conformant Planning: The `Rintanen/Sorting_networks` family encodes a set of problems such that an instance with parameters d and l is satisfiable iff there exists a sorting network of depth d that, for all input sequences of length l, produces a sorted output sequence [21]. The instance with $d = 3$, $l = 6$ is unsatisfiable. It yields an unsatisfiable core in which the universal quantification over the first number of the input sequence has been weakened to an existential one. I.e., even if the ”planner” were allowed to freely choose the first number of the input sequence, there would be no sorting network. This is an interesting information in itself; it additionally implies that there is also no sorting network of depth 3 for input sequences of length 5.

c) Satisfiability of Modal Logic K*:* The `Pan` suite of examples encodes the satisfiability of formulas in the modal logic K as QBF [3]. In the QBF encoding universal quantification runs over the values of an index variable, where each value of the index variable activates a part of the encoding that corresponds to a different $\Diamond$-subformula from the original K formula. This is done to avoid repeating certain subformulas in the resulting QBF, which keeps the complexity of the translation from K to QBF polynomial rather than exponential [3]. We obtained an unsatisfiable core for the instance `k_branch_p-2` in which a universal quantifier had been weakened to an existential one. This showed that either one of two $\Diamond$-subformulas in the input formula is sufficient to obtain unsatisfiability.

d) Answer Set Programming: The `Faber-Leone-Maratea-Ricca/Strategic_Companies` family of examples encodes the question of whether two fixed companies out of a set of companies are strategic [48]. Instance `x25.17` is unsatisfiable, which indicates that the companies under consideration are indeed strategic. In the unsatisfiable core the universal quantification over the variable for a third company has been weakened to an existential one, signaling that that company, too, is strategic.

XII. Experimental Evaluation

a) Setup and Benchmarks: We used one machine with a Xeon E3-1245v5 CPU and 32 GB RAM running Ubuntu 16.04. Run time and memory limits were 300 s and 8 GB. We selected 5342 instances from QBFLIB [20]. Instances were chosen randomly such that equally many instances were taken from each benchmark suite (subject to availability) and, recursively within benchmark suites, equally many instances from each subfamily. I.e., from benchmark suites with fewer than 193 available instances all instances were included, and from each of the remaining suites at least 193 instances were used. We did not use any other selection criteria. Table I shows some statistics. As we were interested in determining the potential for weakening universal to existential quantifiers in the examples, we did not use a preprocessor such as [43]. For our implementation, experimental data, and an extended version of this paper with more tables and plots — some partitioned by benchmark family or structural properties such as number of universal quantifications or alternation depth —, see http://schuppan.de/viktor/ictai18/.

TABLE I: Statistics of structural properties of the set of instances.

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

b) Extracting Unsatisfiable Cores: In our first set of experiments we used DepQBF-a2aecc to extract unsatisfiable cores from the 2528 instances that were found to be unsatisfiable.

In Table II we show how many universal quantifiers could be weakened to existential ones relative to the number of universal quantifiers in the original formula. Column 1 states which kind of unsatisfiable cores was extracted. "q" (resp. "qc") refers to q-cores (resp. qc-cores), "min" to q-minimality for q-cores and to q-,c-minimality for qc-cores, and "minsepcsr" to minimality with separate CSR. Column 2 states the number of solved instances. (For reference, the corresponding numbers for c-cores and c-minimal c-cores are 1830 and 1682, respectively.) Column 3 lists the number of solved instances that had no universal quantifiers. The remaining columns state how many instances exhibited q- or qc-cores whose share of weakened universal quantifiers falls in the range from the first row; as during postprocessing our implementation removes quantifications from the prefix whose variables have no occurrences in the matrix, the numerator of this fraction includes only weakened universal quantifications whose variables still occur in some clause of the matrix of the core. For example, for q-,c-minimal qc-cores with separate CSR, there were 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 in the interval $[0.6, 0.8[$. A number of instances exhibited q-cores in which quite a large share of universal quantifiers was weakened to existential ones; in the light of Section IX note, though, that these cores need not be c-minimal. Finding a qc-core in which a significant share of universal quantifiers is weakened to existential ones seems to require enabling minimization with separate CSR. Then also here instances in which a fairly large share of universal quantifiers is weakened to existential ones can be found; these cores are c-minimal. Unsurprisingly, our data show that for q-cores, q-minimal q-cores, and q-,c-minimal qc-cores with separate CSR higher numbers of weakened universal quantifiers tend to be obtained from original instances with higher numbers of universal quantifiers.

In Figure 1 (a) we compare the sizes of q-,c-minimally unsatisfiable qc-cores obtained with separate CSR with the corresponding c-minimally unsatisfiable c-cores. We find that the qc-cores obtained with CSR can be significantly larger than the corresponding c-cores. This is not surprising: weakening a universal to an existential quantifier corresponds to weakening a conjunction to a disjunction, and proving unsatisfiability of a disjunction requires both disjuncts, while proving unsatisfiability of a conjunction requires only one conjunct. Remember (see Section XI) that already the fact that a certain universal quantifier has been weakened to an existential one may convey valuable information. Our data also show that large increases in 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 core extraction via q-core extraction, qc-core extraction and q-,c-minimal qc-core extraction to q-,c-minimal qc-core extraction with separate CSR. The relation of the run times between no core extraction and q-core extraction is quite variable (b). While moving from q-cores to qc-cores incurs only a moderate overhead (c), adding minimization (d) and, on top of that, separate CSR (e) are quite costly. Notice that (b) involves solving the original versus solving the A2AECC-transformed instance; although the increase in alternation depth of the transformed instance depends on 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 also ran the experiments using set-inner instead of list semantics. As expected, when using set-inner semantics, often fewer universal quantifiers were weakened to existential ones. However, despite lower alternation depth of the transformed formula, we did not find an unambiguous performance advantage for set-inner semantics.

c) Solving A2AECC-Transformed Versus Original Instances: In our second set of experiments we used DepQBF-a2aecc as a preprocessor and ran the following QBF solvers on the original and transformed instances: DepQBF v. 6.03 [19], AIGSolve [49], CAQE v. qbfeval 2017 [50], GhostQ v. 2017-07-26 [51], QESTO v. 1.0 [52], and RAReQS v. 1.1 [51]. This allows for a partial evaluation of our proposed methodology beyond DepQBF-a2aecc. Figure 1 (f)–(h)

TABLE II: Number of instances whose number of weakened ∀ in the core divided by the number of ∀ in the original formula is in a range.

	sol-ved	no ∀ in input	0	[0.002, 0.004[	[0.004, 0.006[	[0.006, 0.008[	[0.008, 0.02[	[0.02, 0.04[	[0.04, 0.06[	[0.06, 0.08[	[0.08, 0.2[	[0.2, 0.4[	[0.4, 0.6[	[0.6, 0.8[	[0.8, 1[	1
q	1649	21	465	1	4	4	13	46	11	8	95	159	305	96	291	130
q min	1139	21	195				5	5	6		37	82	250	96	266	176
qc	1551	21	1528	1			1									
qc min	1441	21	1356	5	3	37	10	5	2	1	1					
qc minsepcsr	927	21	580	1	3	9	42	101	37	9	20	44	27	22	11	

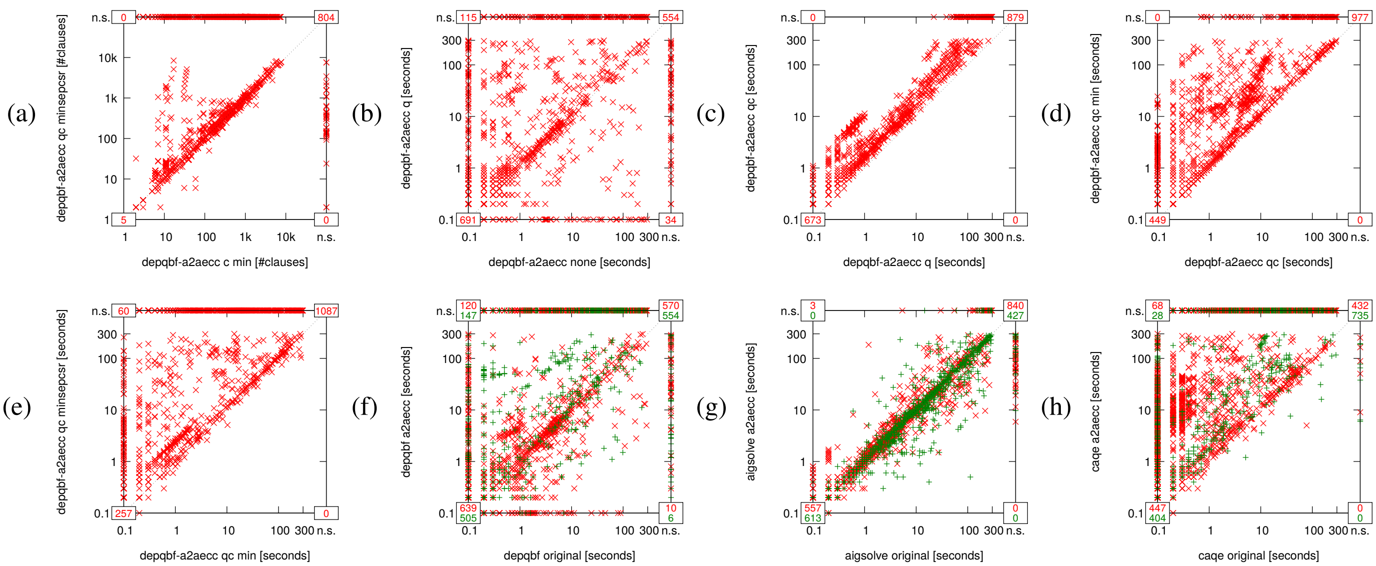

Fig. 1: (a) Comparing sizes of unsatisfiable cores [number of clauses]: x-axis: c-minimal c-cores, y-axis: q-,c-minimal qc-cores with separate CSR. (b)–(e) Comparing run times for extracting unsatisfiable cores [seconds]: (b) x-axis: no cores, y-axis: q-cores; (c) x-axis: q-cores, y-axis: qc-cores; (d) x-axis: qc-cores, y-axis: q-,c-minimal qc-cores; (e) x-axis: q-,c-minimal qc-cores, y-axis: q-,c-minimal qc-cores with separate CSR. (f)–(h): Comparing run times for solving transformed (y-axis) versus original (x-axis) instances [seconds]: (f) `DepQBF`; (g) `AIGSolve`; (h) `CAQE`. Red diagonal crosses are unsatisfiable and green horizontal-/vertical crosses are satisfiable instances. "n.s." stands for not solved. Scatter plots such as the above potentially suffer from overplotting, i.e., several benchmark instances resulting in the same x- and y-coordinates cannot be distinguished in the plot. In our case the effect is most pronounced in the corners of the plot. We therefore replace the crosses *in the corners* by the numbers of instances exhibiting the corresponding x- and y-coordinates. When two values are given, then the red, upper value is for unsatisfiable and the green, lower value for satisfiable instances. For example, in (b) there are 115 instances that were solved in 0.1 seconds by the method on the x-axis and remained unsolved by the method on the y-axis.

compare the run times for solving the transformed versus the original instances for `DepQBF`, `AIGSolve`, and `CAQE`; the picture for `GhostQ` is similar to that of `AIGSolve`, and those for `QESTO` and `RAReQS` are largely similar to that of `CAQE`. We observe that (i) the transformed instances can be solved in many cases, (ii) the overhead for solving the transformed instances depends on the solver, and (iii) some of the 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 transformed versus the original instance on twice the number of universal quantifiers minus the alternation depth in the original instance.

We also ran the experiments using 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 transformation, while for the remaining solvers no clear picture arose.

d) `quantom`*:* Despite its differences `quantom` is the most closely related tool. In our last set of experiments we performed a preliminary comparison of both tools. We used `quantom` to obtain a minimum cardinality set of universal quantifiers that, when weakened to existential ones, make an unsatisfiable QBF satisfiable, and compared the performance with extracting q-minimally unsatisfiable q-cores with `DepQBF-a2aecc` (this compares minimum cardinality diagnoses with minimal unsatisfiable cores, which are quite different!). `DepQBF-a2aecc` (resp. `quantom`) was faster on 835 (resp. 81) instances, with large differences both ways.

XIII. Conclusions

We proposed a notion of unsatisfiable q- and qc-cores for QBF in PCNF that weakens universal to existential quantifiers in addition to removing clauses, leading to unsatisfiable cores and, thus, explanations and diagnoses of unsatisfiability that cannot be obtained from traditional c-cores. We used the A2AECC-transformation to reduce obtaining unsatisfiable q-and qc-cores to obtaining unsatisfiable c-cores. We illustrated with case studies that useful additional information can be obtained from unsatisfiable qc-cores, and we demonstrated in our 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 and its variant affect different solvers, obtaining unsatisfiable q- and qc-cores without using a transformation, e.g., directly from a run of the solver, and extending this work to logics with quantification beyond QBF.

Acknowledgments

I thank the authors of [25], especially Sven Reimer, for discussion of their work and for providing me with `quantom` [26]. I thank Alessandro Cimatti for mentioning that proofs can be used to compare formulas. I am grateful to the reviewers for their suggestions on how to improve the 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.*, vol. 10, 1999.

[3] G. Pan and M. Y. Vardi, "Optimizing a BDD-Based Modal Solver," in *CADE*, ser. LNCS, vol. 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*, ser. LNCS, vol. 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*, vol. 3, no. 2, 1991.

[8] R. Bruni and A. Sassano, "Restoring Satisfiability or Maintaining Unsatisfiability by finding small Unsatisfiable Subformulae," *Electronic Notes in Discrete Mathematics*, vol. 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.*, vol. 77, no. 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*, ser. LNCS, vol. 4121. Springer, 2006.

[13] R. Reiter, "A Theory of Diagnosis from First Principles," *Artif. Intell.*, vol. 32, no. 1, 1987.

[14] S. Schlobach, "Diagnosing Terminologies," in *AAAI*. AAAI Press / The MIT Press, 2005.

[15] H. Kleine Büning and X. Zhao, "Minimal False Quantified Boolean Formulas," in *SAT*, ser. LNCS, vol. 4121. Springer, 2006.

[16] A. Ignatiev, M. Janota, and J. Marques-Silva, "Quantified Maximum Satisfiability: A Core-Guided Approach," in *SAT*, ser. LNCS, vol. 7962. Springer, 2013.

[17] F. Lonsing and U. Egly, "Incrementally Computing Minimal Unsatisfiable Cores of QBFs via a Clause Group Solver API," in *SAT*, ser. LNCS, vol. 9340. Springer, 2015.

[18] J. Slaney, "Set-theoretic duality: A fundamental feature of combinatorial optimisation," in *ECAI*, ser. Frontiers in Artificial Intelligence and Applications, vol. 263. IOS Press, 2014.

[19] F. Lonsing and U. Egly, "DepQBF 6.0: A Search-Based QBF Solver Beyond Traditional QCDCL," in *CADE*, ser. LNCS, vol. 10395. Springer, 2017.

[20] E. Giunchiglia, M. Narizzano, L. Pulina, and A. Tacchella, "Quantified Boolean Formulas satisfiability library (QBFLIB)," http://www.qbflib.org/.

[21] J. Rintanen, "Asymptotically Optimal Encodings of Conformant Planning in QBF," in *AAAI*. AAAI Press, 2007.

[22] F. Lonsing and A. Biere, "Failed Literal Detection for QBF," in *SAT*, ser. LNCS, vol. 6695. Springer, 2011.

[23] F. Lonsing, U. Egly, and M. Seidl, "Q-Resolution with Generalized Axioms," in *SAT*, ser. LNCS, vol. 9710. Springer, 2016.

[24] R. Brummayer, F. Lonsing, and A. Biere, "Automated Testing and Debugging of SAT and QBF Solvers," in *SAT*, ser. LNCS, vol. 6175. Springer, 2010.

[25] S. Reimer, M. Sauer, P. Marin, and B. Becker, "QBF with Soft Variables," *ECEASST*, vol. 70, 2014.

[26] S. Reimer, F. Pigorsch, C. Scholl, and B. Becker, "Enhanced Integration of QBF Solving Techniques," in *MBMV*. Verlag Dr. Kovac, 2012.

[27] I. Shlyakhter, R. Seater, D. Jackson, M. Sridharan, and M. Taghdiri, "Debugging Overconstrained Declarative Models Using Unsatisfiable Cores," in *ASE*. IEEE Computer Society, 2003.

[28] A. Kalyanpur, B. Parsia, and B. C. Grau, "Beyond Asserted Axioms: Fine-Grain Justifications for OWL-DL Entailments," in *DL*, ser. CEUR Workshop Proceedings, vol. 189. CEUR-WS.org, 2006.

[29] A. Kalyanpur, B. Parsia, E. Sirin, and B. C. Grau, "Repairing Unsatisfiable Concepts in OWL Ontologies," in *ESWC*, ser. LNCS, vol. 4011. Springer, 2006.

[30] É. Grégoire, B. Mazure, and C. Piette, "MUST: Provide a Finer-Grained Explanation of Unsatisfiability," in *CP*, ser. LNCS, vol. 4741. Springer, 2007.

[31] S. Grimm and J. Wissmann, "Elimination of Redundancy in Ontologies," in *ESWC*, ser. LNCS, vol. 6643. Springer, 2011.

[32] I. Pill and T. Quaritsch, "Behavioral Diagnosis of LTL Specifications at Operator Level," in *IJCAI*. IJCAI/AAAI, 2013.

[33] V. Schuppan, "Extracting unsatisfiable cores for LTL via temporal resolution," *Acta Inf.*, vol. 53, no. 3, 2016.

[34] ——, "Enhancing unsatisfiable cores for LTL with information on temporal relevance," *Theor. Comput. Sci.*, vol. 655, Part B, 2016.

[35] H. Kleine Büning and U. Bubeck, "Theory of Quantified Boolean Formulas," in *Handbook of Satisfiability*. IOS Press, 2009.

[36] L. J. Stockmeyer and A. R. Meyer, "Word Problems Requiring Exponential Time: Preliminary Report," in *STOC*. ACM, 1973.

[37] L. J. Stockmeyer, "The Polynomial-Time Hierarchy," *Theor. Comput. Sci.*, vol. 3, no. 1, 1976.

[38] C. Wrathall, "Complete Sets and the Polynomial-Time Hierarchy," *Theor. Comput. Sci.*, vol. 3, no. 1, 1976.

[39] A. V. Gelder, "Contributions to the Theory of Practical Quantified Boolean Formula Solving," in *CP*, ser. LNCS, vol. 7514. Springer, 2012.

[40] S. Coste-Marquis, H. Fargier, J. Lang, D. L. Berre, and P. Marquis, "Representing Policies for Quantified Boolean Formulae," in *KR*. AAAI Press, 2006.

[41] N. Francez, "The Granularity of Meaning in Proof-Theoretic Semantics," in *LACL*, ser. LNCS, vol. 8535. Springer, 2014.

[42] H. Kleine Büning, M. Karpinski, and A. Flögel, "Resolution for Quantified Boolean Formulas," *Inf. Comput.*, vol. 117, no. 1, 1995.

[43] A. Biere, F. Lonsing, and M. Seidl, "Blocked Clause Elimination for QBF," in *CADE*, ser. LNCS, vol. 6803. Springer, 2011.

[44] A. Nadel, "Boosting minimal unsatisfiable core extraction," in *FMCAD*. IEEE, 2010.

[45] M. H. Liffiton and K. A. Sakallah, "Algorithms for Computing Minimal Unsatisfiable Subsets of Constraints," *J. Autom. Reasoning*, vol. 40, no. 1, 2008.

[46] J. Marques-Silva, "Computing Minimally Unsatisfiable Subformulas: State of the Art and Future Directions," *Multiple-Valued Logic and Soft Computing*, vol. 19, no. 1-3, 2012.

[47] A. Belov, I. Lynce, and J. Marques-Silva, "Towards efficient MUS extraction," *AI Commun.*, vol. 25, no. 2, 2012.

[48] W. Faber, N. Leone, M. Maratea, and F. Ricca, "Looking Back in DLV: Experiments and Comparison to QBF Solvers," in *ASP*, 2007.

[49] F. Pigorsch and C. Scholl, "An AIG-Based QBF-solver using SAT for preprocessing," in *DAC*. ACM, 2010.

[50] L. Tentrup, "On Expansion and Resolution in CEGAR Based QBF Solving," in *CAV*, ser. LNCS, vol. 10427. Springer, 2017.

[51] M. Janota, W. Klieber, J. Marques-Silva, and E. M. Clarke, "Solving QBF with Counterexample Guided Refinement," in *SAT*, ser. LNCS, vol. 7317. Springer, 2012.

[52] M. Janota and J. Marques-Silva, "Solving QBF by Clause Selection," in *IJCAI*. AAAI Press, 2015.