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 [GR03,AGS05], variants of planning [Rin99,Tur02], satisfiability of formulas in the modal logic K [PV03], and a number of problems in knowledge representation [EETW00] and formal methods [AB00, SB01]; for a more extensive list see [GMN09]. Unsatisfiable cores have been established as a fundamental concept in applied 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 [CD91,BS01,SC03,SSJ+03,YM05,Sch12], as building blocks to obtain advanced explanations of unsatisfiability [KLM06], to diagnose unsatisfiability [Rei87], and

*Extended version; git: 0ba3bbe; compiled: 2020-06-17 18:33:02+02:00.

to repair unsatisfiability [Sch05]. Previous work on unsatisfiable cores for QBF in PCNF (prenex conjunctive normal form) removes clauses to weaken formulas [YM05,KZ06,IJM13,LE15].

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 [KZ06] 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 $\forall$-to-$\exists$ reducible. We then describe the implementation of our approach in `DepQBF` [LE17] in Section 12. We illustrate with case studies including two-player games [GR03], conformant planning [Rin07], and satisfiability of modal logic K [PV03] 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` [GNPT] (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 [Sch18]. 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. Previous work on unsatisfiable cores for QBF in PCNF uses the traditional notion of removal of clauses from the matrix [YM05,KZ06,IJM13,LE15].

Reimer et al. [RSMB14] 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 [CFLS93] 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` [RPSB12]. The main differences between our work and that of Reimer et al. [RSMB14] are as follows. When seen as a specification language for sets of prefixes of QBF the notion of soft variables in Definition 1–3 of Reimer et al. [RSMB14] is more powerful than our notion of cores in Definition 3.1. Reimer et al. [RSMB14] 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 [IJM13]). Reimer et al. [RSMB14] make no connection to unsatisfiable cores. Reimer et al. [RSMB14] use a MaxSAT-based algorithm [ZSM03]; we use a standard algorithm [Mar12] to obtain (optionally) minimal clausal unsatisfiable cores. Reimer et al. [RSMB14] search for a maxim*um* solution, while we (optionally) search for a minim*al* solution. Reimer et al. [RSMB14] do not modify the matrix, whereas we (optionally) also remove clauses from the matrix. As a minor practical point, our approach does not require to enhance a QBF in PCNF with additional information, thus making a large set of benchmarks directly available.

Weakening universal to existential quantifiers has been called "quantifier abstraction" in a work on failed literal detection for QBF [LB11] and "existential abstraction" in the context of generalizing Q-resolution [LES16]. `QBFDD` [BLB10, qbfdd] allows quantifier manipulations when minimizing failure-inducing input.

QCSP. Ferguson and O'Sullivan [FO07] define a number of weakening operations for QCSP (quantified constraint satisfaction problems) [Che04]. For universal quan-

tifications 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 what is essentially an insertion-based algorithm [Mar12] to find minimal unsatisfiable cores by Junker [Jun01,Jun04] 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 [BHZ06]). 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 [MOQ15] 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. [BCM09] 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. [SSJ+03] 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. [SSJ+03] 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 [KLM06,Sch16b], temporal logic [Sch12,Sch16a], and constraint programming [GMP07,FO07]; equivalent formulas [GW11]; unrealizable cores [Sch12]; vacuity [AFF+03,GC04]; justifications [KPG06,LPSV06,HPS08]; diagnoses [PQ13]; and repair [KPSG06,DQF14]. For a uniform treatment of some such notions and their relationships see the work [MJ14] on minimal sets over monotone predicates.

2. Preliminaries

We consider QBF in PCNF [KB09,GMN09]; using standard techniques any QBF can be turned into an equivalent QBF in PCNF [KB09].

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. A literal l evaluates to 1 under v iff $l = \top$, $l = \neg\bot$, $l = p$ and $v(p) = 1$, or $l = \neg p$ and $v(p) = 0$. A clause c evaluates to 1 under v iff one or more of its literals evaluate to 1 under v. The empty clause evaluates to 0. A CNF formula C evaluates to 1 under v iff all of its clauses evaluate to 1 under v. The empty CNF formula evaluates to 1. A CNF formula C is *satisfiable* if there exists an assignment v such that C evaluates to 1 under v; otherwise, it is unsatisfiable.

$\forall$ 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 [SM73]; 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 [Sto76,Wra76], 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 [LE15]) 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 $\Pi.C$.

If $\Pi'.C'$ is an unsatisfiable c-, q-, or qc-core of $\Pi.C$, then $\Pi.C$ is unsatisfiable.

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 [CD91,BS01,SC03,SSJ+03,YM05,Sch12]. Some authors prefer minimally or minimum cardinality unsatisfiable cores [CD91,LM04,SC03,TCJ08], and some

authors use unsatisfiable cores as building blocks of more advanced explanations [KLM06]. 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 [Gel12,CFL+06]. 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.

(1) The labels of non-leaf nodes are variables in Π; the labels of leaf nodes are irrelevant.
(2) The labels of edges are Booleans; they represent assignments to the variables that are labeling their source nodes.
(3) If a node is labeled with a universally quantified variable, then it has one outgoing edge; it is labeled with either 0 or 1.
(4) 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.
(5) 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.
(6) 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 [Fra14]. For example, we can assign to each unsatisfiable QBF in PCNF the set of its Q-resolution proofs of unsatisfiability [KKF95]. 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 [KKF95].

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

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

$\Pi.C$ is clearly q-minimally unsatisfiable; however, removing any clause from C results in a proper c-core of $\Pi.C$, i.e., $\Pi.C$ is not c-minimally unsatisfiable. This concludes the proof. □

Lemma 5.1. *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

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

is unsatisfiable. Expanding $\exists p_l$ gives unsatisfiability of both

$$(Q_{l+1} p_{l+1} \ldots Q_n p_n . c_1 \wedge \ldots \wedge c_{i-1} \wedge c_i \wedge c_{i+1} \wedge \ldots \wedge c_m)[\bot / p_l],$$

and

$$(Q_{l+1} p_{l+1} \ldots Q_n p_n . c_1 \wedge \ldots \wedge c_{i-1} \wedge c_i \wedge c_{i+1} \wedge \ldots \wedge c_m)[\top / p_l].$$

Without limitation of generality let p_l occur non-negated in c_i. Hence,

$$(Q_{l+1}p_{l+1}\dots Q_np_n.c_1\wedge\dots\wedge c_{i-1}\wedge c_{i+1}\wedge\dots\wedge c_m)[\top/p_l]$$

is also unsatisfiable. Finally, by the semantics of $\forall p_l$,

$$\forall p_lQ_{l+1}p_{l+1}\dots Q_np_n.c_1\wedge\dots\wedge c_{i-1}\wedge c_{i+1}\wedge\dots\wedge c_m$$

is unsatisfiable as desired.

For the inductive case let $l-1>0$. First let $Q_1=\exists$. By assumption

$$\Pi'.C'=\exists p_1Q_2p_2\dots Q_{l-1}p_{l-1}\exists p_lQ_{l+1}p_{l+1}\dots Q_np_n.c_1\wedge\dots\wedge c_{i-1}\wedge c_i\wedge c_{i+1}\wedge\dots\wedge c_m$$

is unsatisfiable. Expanding $\exists p_1$ gives unsatisfiability of both

$$(Q_2p_2\dots Q_{l-1}p_{l-1}\exists p_lQ_{l+1}p_{l+1}\dots Q_np_n.c_1\wedge\dots\wedge c_{i-1}\wedge c_i\wedge c_{i+1}\wedge\dots\wedge c_m)[\bot/p_1],$$

and

$$(Q_2p_2\dots Q_{l-1}p_{l-1}\exists p_lQ_{l+1}p_{l+1}\dots Q_np_n.c_1\wedge\dots\wedge c_{i-1}\wedge c_i\wedge c_{i+1}\wedge\dots\wedge c_m)[\top/p_1].$$

With the inductive assumption both

$$(Q_2p_2\dots Q_{l-1}p_{l-1}\forall p_lQ_{l+1}p_{l+1}\dots Q_np_n.c_1\wedge\dots\wedge c_{i-1}\wedge c_{i+1}\wedge\dots\wedge c_m)[\bot/p_1],$$

and

$$(Q_2p_2\dots Q_{l-1}p_{l-1}\forall p_lQ_{l+1}p_{l+1}\dots Q_np_n.c_1\wedge\dots\wedge c_{i-1}\wedge c_{i+1}\wedge\dots\wedge c_m)[\top/p_1]$$

are unsatisfiable as well. Finally, by the semantics of $\exists p_1$,

$$\exists p_1Q_2p_2\dots Q_{l-1}p_{l-1}\forall p_lQ_{l+1}p_{l+1}\dots Q_np_n.c_1\wedge\dots\wedge c_{i-1}\wedge c_{i+1}\wedge\dots\wedge c_m$$

is unsatisfiable as desired.

Now let $Q_1=\forall$. By assumption

$$\Pi'.C'=\forall p_1Q_2p_2\dots Q_{l-1}p_{l-1}\exists p_lQ_{l+1}p_{l+1}\dots Q_np_n.c_1\wedge\dots\wedge c_{i-1}\wedge c_i\wedge c_{i+1}\wedge\dots\wedge c_m$$

is unsatisfiable. Expanding $\forall p_1$ gives unsatisfiability of

$$(Q_2p_2\dots Q_{l-1}p_{l-1}\exists p_lQ_{l+1}p_{l+1}\dots Q_np_n.c_1\wedge\dots\wedge c_{i-1}\wedge c_i\wedge c_{i+1}\wedge\dots\wedge c_m)[\bot/p_1]$$

or

$$(Q_2p_2\dots Q_{l-1}p_{l-1}\exists p_lQ_{l+1}p_{l+1}\dots Q_np_n.c_1\wedge\dots\wedge c_{i-1}\wedge c_i\wedge c_{i+1}\wedge\dots\wedge c_m)[\top/p_1].$$

Without limitation of generality let the first part $\bot/p_1$ be unsatisfiable. Hence, with the inductive assumption,

$$(Q_2p_2\dots Q_{l-1}p_{l-1}\forall p_lQ_{l+1}p_{l+1}\dots Q_np_n.c_1\wedge\dots\wedge c_{i-1}\wedge c_{i+1}\wedge\dots\wedge c_m)[\bot/p_1]$$

is unsatisfiable. Finally, by the semantics of $\forall p_1$,

$$\forall p_1Q_2p_2\dots Q_{l-1}p_{l-1}\forall p_lQ_{l+1}p_{l+1}\dots Q_np_n.c_1\wedge\dots\wedge c_{i-1}\wedge c_{i+1}\wedge\dots\wedge c_m$$

is unsatisfiable as desired. This concludes the proof. □

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 [KZ06]. 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 PSPACE-hardness of QCMF is similar. This concludes the proof. □

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- 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

[a] In parts of the literature in a set-based setting the complements of satisfiable subsets (cores) are referred to as diagnoses [Rei87] and sometimes as repair (solutions) [KPSG06]; 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., [Rei87]; for a generic formulation see Slaney [Sla14]). 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.

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

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

Clearly, $\Pi'.C'$ is a q-maximally satisfiable core of $\Pi.C$, and p_3 occurs in both, $(p_2 \rightarrow p_3)$ or $(p_3 \rightarrow p_2)$; conjoining $(p_2 \rightarrow p_3)$ or $(p_3 \rightarrow p_2)$ with Π' does not make the result unsatisfiable. This concludes the proof. □

Lemma 7.1. *Let* $\Pi.C = Q_1 p_1 \ldots Q_{l-1} p_{l-1} \forall p_l Q_{l+1} p_{l+1} \ldots Q_n p_n.C$ *be a QBF in PCNF. Let* $\Pi'.C' = Q'_1 p_1 \ldots Q'_{l-1} p_{l-1} \exists p_l Q'_{l+1} p_{l+1} \ldots Q'_n p_n.C'$ *be a c-maximally satisfiable qc-core of* $\Pi.C$ *such that* p_l *occurs in some clause in* $C \setminus C'$. *Then* $\Pi''.C' = Q'_1 p_1 \ldots Q'_{l-1} p_{l-1} \forall p_l Q'_{l+1} p_{l+1} \ldots Q'_n p_n.C'$ *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. This is formalized in Definition 8.1.

Definition 8.1. (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 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

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

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 "$\neq \{(p_i \to p'_i), (p'_i \to p_i)\}$" 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*

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

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

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

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

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

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) and (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

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

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

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

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. This concludes the proof. □

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 [Sto76,Wra76]. 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 $(p_i \to p'_i)$ and $(p'_i \to p_i)$ 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 their first quantifier:

- $QBF_{m,\forall} = \{\Pi.C \mid ad(\Pi.C) < m \text{ or } (ad(\Pi.C) = m \text{ and } \Pi = \forall p_1 Q_2 p_2 \ldots)\}$
- $QBF_{m,\exists} = \{\Pi.C \mid ad(\Pi.C) < m \text{ or } (ad(\Pi.C) = m \text{ and } \Pi = \exists p_1 Q_2 p_2 \ldots)\}$

[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.

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. For experimental results we refer to Section 14.

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 $(p' \to p)$ (resp. $(p \to p')$) is a quantified blocked clause [BLS11] 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 [NRS14,Nad10,LS08], as does `DepQBF` [LE15], then placing each pair of clauses $(p_i \to p'_i), (p'_i \to p_i)$ introduced by Definition 8.1 in a separate clause group ensures that either none or both of $(p_i \to p'_i), (p'_i \to p_i)$ 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

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

The unsatisfiable c-core $\Pi'.C'_1 = \forall p' \exists p.(p \to p') \wedge (p' \to p) \wedge (p)$ of $a2aecc(\Pi.C)$ corresponds to the unsatisfiable c-core $\forall p.(p)$ of $\Pi.C$; the unsatisfiable c-core $\Pi'.C'_2 = \forall p' \exists p.(p \to p') \wedge (p' \to p) \wedge (\neg p)$ 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 $(p \to p'), (p' \to p)$ 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 [Sla14] 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 very conservatively by assigning maximal meaning to the order of the quantifications in a quantifier block, whereas set-inner semantics acts very 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

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

with a non-c-minimally unsatisfiable qc-core

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

We can see by inspection that the unsatisfiability of $\Pi'.C'$ is caused by $\exists p_1 \forall p_2.(p_1 \rightarrow p_2) \wedge (p_2 \rightarrow p_1)$ 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, $\exists p_3 \exists p_4.(p_3 \rightarrow p_4)$ 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: $\Pi'.C'' = \exists p_1 \forall p_2 \exists p_3 \exists p_4.(p_1 \rightarrow p_2) \wedge (p_2 \rightarrow p_1)$. 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''. □

11. ∀-to-∃ Reducibility

We now lift the discussion of the previous 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 $\forall$-to-$\exists$ 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* $\forall$*-to-*$\exists$ *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* $\forall$*-to-*$\exists$ *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 [GMN09] — $\forall p$ is either trivially or not $\forall$-to-$\exists$ 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 $\forall$-to-$\exists$ reducible, p_2 is not $\forall$-to-$\exists$ reducible, and p_3 is trivially $\forall$-to-$\exists$ reducible. □

To better understand the potential for weakening universal quantifiers to existential quantifiers we are interested in computing which variables in an unsatisfiable QBF $\Pi.C$ are non-trivially $\forall$-to-$\exists$ 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 $\forall$-to-$\exists$ 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 $\forall$-to-$\exists$ 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 quantifiers 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 unsatisfiable.
(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` [LE17] 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 [LE15]. 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 [Mar12] with CSR (clause set refinement) [BLM12] to minimize C'; we use the `DepQBF` API [LE15] 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` [GNPT], 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.

13.1. *Winning strategies in two-player games*

The `Gent-Rowley` suite models variants of the well-known Connect-4 game [GR03]. 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 [GR03] 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 [GR03]. This raises the question of whether this part of the model of the game is indeed as intended.

13.2. *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 [Rin07,BG00]. 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.

13.3. *Satisfiability of modal logic* K

The `Pan` suite of examples encodes formulas in the modal logic K as equisatisfiable QBF [PV03,BHS00]. 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 [PV03]. 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.

13.4. *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 [FLMR07,LPF+06,CEG97]. 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 and 32 GB RAM, utilizing 3 out of 4 physical cores for our experiments. The operating system was Ubuntu 16.04. Run time and memory limits were 300 s and 8 GB. The experiments took about 2.5 months on our machine.

Our set of benchmarks consists of 5342 instances from `QBFLIB` [GNPT]. 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 (for the selection algorithm see Appendix A). 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. Table 1 shows the resulting number of instances per benchmark suite; "solved" means solved by any solver in any of our experiments. Table 2 shows the minimum, first quartile, medium, third quartile, maximum, and mean values of the number of universal quantifiers, the number of existential quantifiers, the alternation depth, the number of clauses, and the maximum variable index for our benchmark set. We did not apply a preprocessor such as `bloqqer` [BLS11] 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`.

For our implementation, our experimental data, and more tables and plots, partitioned by benchmark family or structural properties such as number of universal quantifications or alternation depth, see `http://schuppan.de/viktor/ijait20/`.

In the tables and plots below "n.s." abbreviates not solved. In plots red diagonal crosses represent unsatisfiable and green horizontal/vertical crosses represent satisfiable instances. Scatter plots such as Figure 2 (a) potentially suffer from 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 Figure 2 (a) there are 804 instances that remained unsolved by both methods.

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 Section 13 we already described some of the unsatisfiable qc-cores that we obtained in more detail.

In the upper two sections of Table 3 we show how many universal quantifiers could be weakened to existential quantifiers as a share of the number of univer-

Table 1. Number of instances per benchmark suite.

	all	solved unsatisfiable	solved satisfiable
Akshay-Chakraborty-John-Shah-Rabe	20	2	16
Amendola-Ricca-Truszczynski	112	9	7
Ansotegui	38	12	11
Ayari	71	48	23
Basler	193	75	118
Biere	194	25	159
Cashmore-Fox-Giunchiglia	150	110	40
Castellini	169	112	57
Chen-Interian	194	16	0
Diptarama-Jordan-Shinohara	14	11	2
Egly-Seidl-Tompits-Woltran-Zolda	194	97	78
Faber-Leone-Maratea-Ricca	194	128	7
Gent-Rowley	193	142	10
Herbstritt	194	157	27
Interian	193	24	69
Jordan-Kaiser	194	83	87
Katz	20	8	8
Klieber	30	15	6
Kontchakov	136	70	66
Kronegger-Pfandler-Pichler	194	133	44
Lahiri-Seshia	3	1	2
Lee-Jiang	5	2	3
Letombe	194	85	107
Letz	14	9	5
Ling	8	3	5
Mangassarian-Veneris	170	60	71
MayerEichberger-Saffidine	113	3	35
Messinger	63	0	9
Miller-Marin	194	189	5
Miller-Scholl-Becker	194	160	18
Mneimneh-Sakallah	180	44	123
Narizzano	193	78	115
Palacios	24	9	14
Pan	194	89	98
Peitl	10	10	0
Preusser	12	0	9
qbfeval12	17	8	9
Rabe	14	3	0
Rintanen	131	55	71
Sauer-Reimer	193	42	142
Scholl-Becker	64	30	25
Seidl	194	194	0
Tacchella	193	122	70
Tentrup	74	17	29
Wintersteiger	194	38	96
sum	5342	2528	1896

sal quantifiers in the original formula. In column 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

Table 2. Statistics about structural properties of the benchmark set.

	min.	1st quartile	median	3rd quartile	max.	mean
all ($n = 5342$)						
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
solved unsatisfiable ($n = 2528$)						
number of ∀	0	20	81.5	189	55,022	313.8
number of ∃	1	622	2,993	8,332.5	2,202,774	23,311.2
alternation depth	1	3	3	12	781	25.9
number of clauses	5	2,274	11,207.5	35,299	5,534,890	104,313.6
max. variable index	5	764.25	3,287.5	9,880	2,202,778	42,507.8
solved satisfiable ($n = 1896$)						
number of ∀	0	12	63	232	10,404	314.8
number of ∃	1	408.75	1,338.5	4,707	1,112,278	7,802.7
alternation depth	1	2	3	4	133	6.3
number of clauses	1	1,525.25	5,007	18,777	2,812,458	35,569.7
max. variable index	1	458	1,592.5	5,605.5	1,112,282	15,112.6

q- and c-minimality for unsatisfiable qc-cores, and "minsepcsr" stands for q- and c-minimality with separate CSR. "list" refers to list semantics and "set-inner" refers to set-inner semantics in the A2AECC-transformation (see Section 9). In column 2 we list how many instances of each kind were solved (this is the sum of the remaining columns). For reference, the corresponding numbers for c-cores and c-minimal c-cores are 1830 and 1682, respectively. In column 3 we provide the number of solved instances that had no universal quantifiers to begin with. In the remaining columns 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 row 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

Table 3. 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 unsatisfiable c-cores and c-minimally unsatisfiable c-cores are 1830 and 1682, respectively.

	solved	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 list	1649	21	465	1	4	4	13	46	11	8	95	159	305	96	291	130
q set-inner	1516	21	432	2	6	5	20	25	22	25	128	140	183	85	290	132
q min list	1139	21	195				5	5	6		37	82	250	96	266	176
q min set-inner	995	21	207				2	1	6		32	78	146	79	245	178
qc list	1551	21	1528	1			1									
qc set-inner	1414	21	1392						1							
qc min list	1441	21	1356	5	3	37	10	5	2	1	1					
qc min set-inner	1305	21	1266	1	1	3	5	3	3	2						
qc minsepcsr list	927	21	580	1	3	9	42	101	37	9	20	44	27	22	11	
qc minsepcsr set-inner	854	21	659		2	1	27	33	12	5	14	37	20	20	3	
enuma2e1 list	986	21	831		1		5	9	8	1	21	15	4	10	7	53
enuma2e1 set-inner	930	21	825		1		1	5			5	9	5	4	1	53
enuma2e2 list	657	21	385	1	2	9	29	38	28	22	36	19	7	10	5	45
enuma2e2 set-inner	645	21	465	2		6	11	25	15	10	15	21	4	6		44

we found instances in which a fairly large share of universal quantifiers had been weakened to existential quantifiers (these cores are c-minimally unsatisfiable). Unsurprisingly, Figure 1 shows 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.

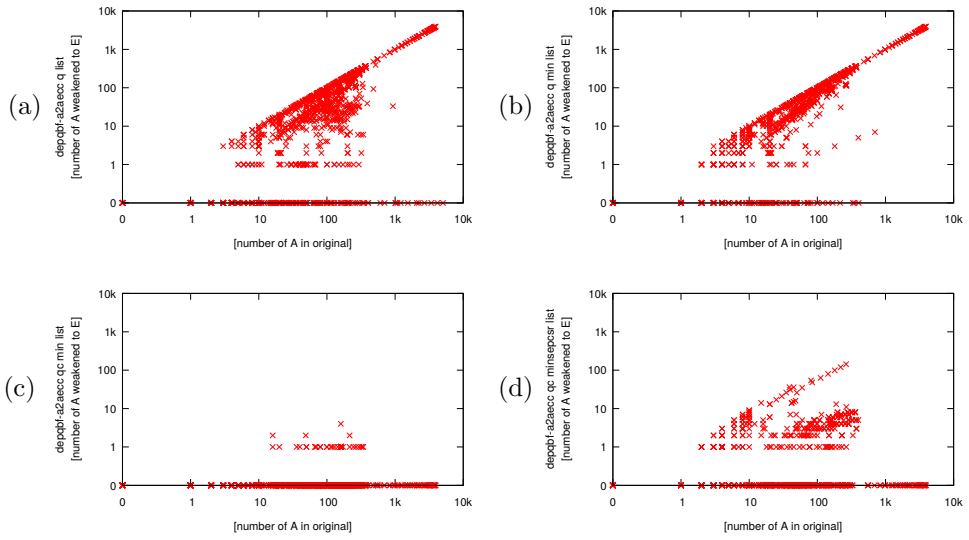

Fig. 1. Number of universal quantifiers weakened to existential quantifiers depending on the number of universal quantifiers in the original formula: (a): unsatisfiable q-cores; (b): q-minimally unsatisfiable q-cores; (c): q- and c-minimally unsatisfiable qc-cores; (d): q- and c-minimally unsatisfiable qc-cores with separate CSR.

In the lower section of Table 3 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 2 (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 in terms of number of clauses. 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 cer-

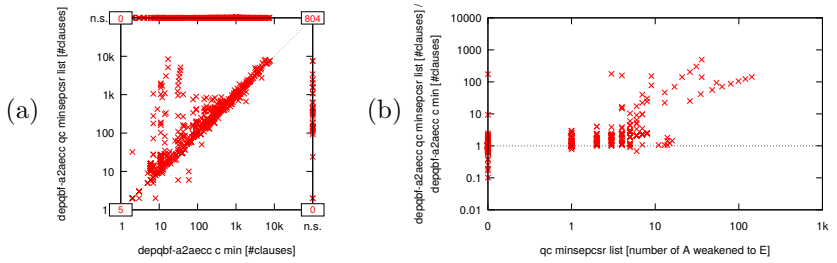

Fig. 2. (a) Comparing the sizes of unsatisfiable cores [number of clauses]: x-axis: c-minimally unsatisfiable c-cores, y-axis: q- and c-minimally unsatisfiable qc-cores with separate CSR. (b) Ratio of the core sizes [numbers of clauses] between q- and c-minimally unsatisfiable qc-cores with separate CSR and c-minimally unsatisfiable c-cores (y-axis) depending on the number of universal quantifications weakened to existential quantifications in the q- and c-minimally unsatisfiable qc-core with separate CSR (x-axis); only pairs for which both unsatisfiable cores were obtained are included.

tain universal quantifier has been weakened to an existential quantifier may convey valuable information, irrespective of the remainder of the unsatisfiable core under consideration. Figure 2 (b) shows that large increases in unsatisfiable core size tend to coincide with large numbers of weakened universal quantifiers, which is expected.

In Figure 3 (a)–(f) we show the run time overhead that is incurred by each step when going from no unsatisfiable core extraction via unsatisfiable c-core extraction to c-minimally unsatisfiable c-core extraction and 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. Unsatisfiable c-core extraction incurs limited costs (a); minimization comes with a high overhead (b). The relation of the run times between no unsatisfiable core extraction and unsatisfiable q-core extraction is quite variable (c). Moving from unsatisfiable q-core extraction to unsatisfiable qc-core extraction incurs only a moderate overhead (d). In contrast, additionally requiring q- and c-minimality (e) and, on top of that, using separate CSR (f) are quite costly. In Figure 3 (g)–(l) we show the corresponding plots for memory; memory usage turned out not to be a problem for `DepQBF-a2aecc`. 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) (see Figure 4).

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 (see Figure 5).

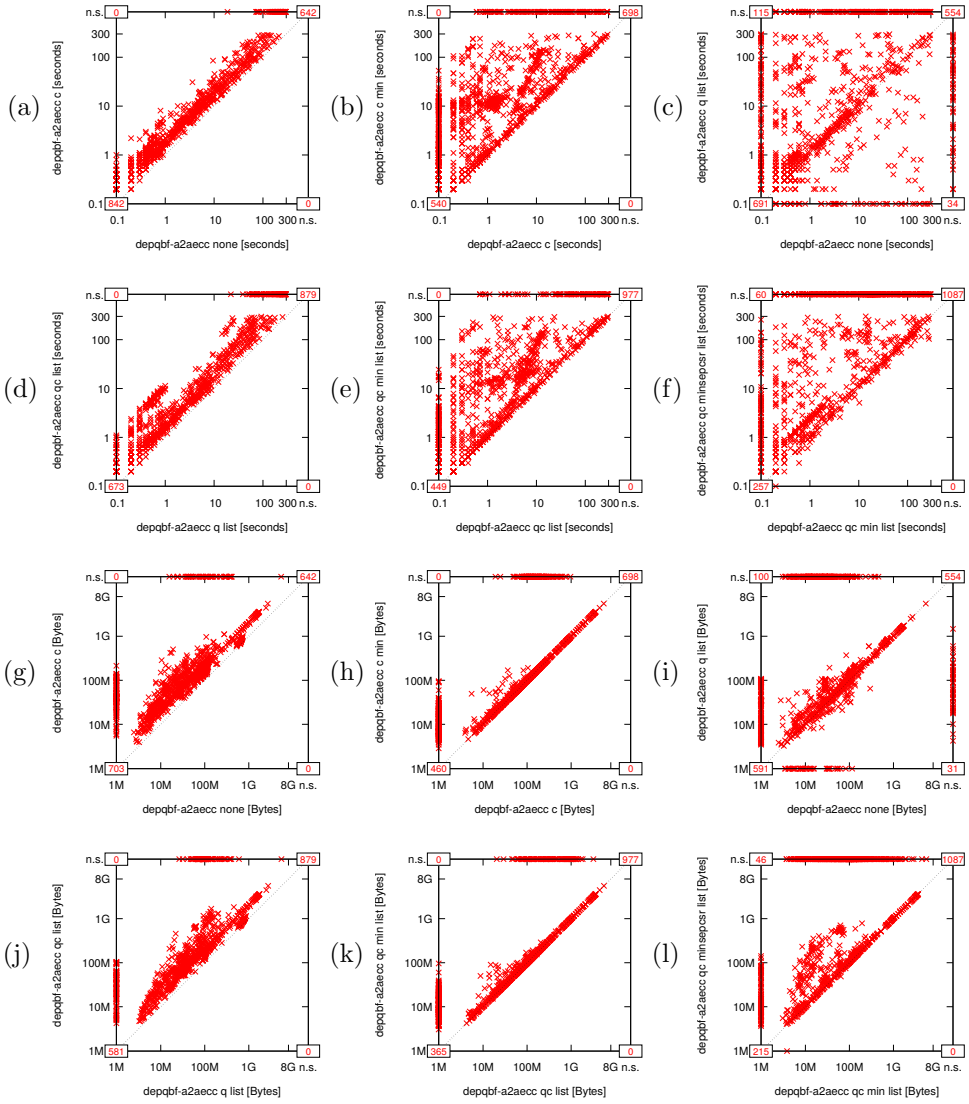

Fig. 3. (a)–(f) Comparing the run times for extracting unsatisfiable cores in list semantics [seconds]: (a) x-axis: no unsatisfiable cores, y-axis: unsatisfiable c-cores; (b) x-axis: unsatisfiable c-cores, y-axis: c-minimally unsatisfiable c-cores; (c) x-axis: no unsatisfiable cores, y-axis: unsatisfiable q-cores; (d) x-axis: unsatisfiable q-cores, y-axis: unsatisfiable qc-cores; (e) x-axis: unsatisfiable qc-cores, y-axis: q- and c-minimally unsatisfiable qc-cores; (f) x-axis: q- and c-minimally unsatisfiable qc-cores, y-axis: q- and c-minimally unsatisfiable qc-cores with separate CSR. (g)–(l): as (a)–(f) but for memory [Bytes].

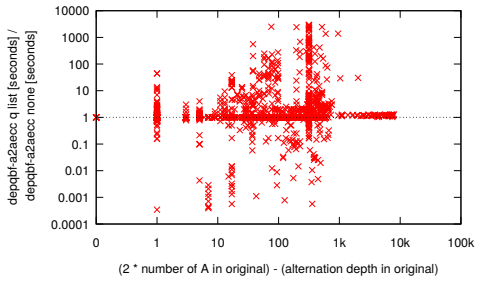

Fig. 4. Ratio of the run times [seconds] between unsatisfiable q-core extraction and no unsatisfiable core extraction (y-axis) depending on the increase in alternation depth between the original and the A2AECC-transformed instances (x-axis); only pairs for which both instances were solved are included. Using `gbutils` (see [gbutils]) we obtained Kendall's rank correlation coefficient $\tau = 0.17$ at $p = 3.1 \cdot 10^{-25}$.

14.3. *Solving A2AECC-transformed versus original instances*

Our method for extracting unsatisfiable q- and qc-cores as described in Section 8 consists of a preprocessing step that applies the A2AECC-transformation, extraction of unsatisfiable c-cores, and a postprocessing step that maps unsatisfiable c-cores back to unsatisfiable q- or qc-cores. This makes it possible to investigate the impact of the preprocessing step not only on `DepQBF-a2aecc` but also on other QBF solvers, thus allowing for a partial evaluation of our proposed methodology beyond `DepQBF-a2aecc`. Therefore, in our second set of experiments, we used `DepQBF-a2aecc` as a preprocessor and ran the following QBF solvers on both the original and the A2AECC-transformed instances: `DepQBF` v. 6.03 [LE17,depqbf], `AIGSolve` [PS10,aigsolve], `CAQE` v. qbfeval 2017 [Ten17,caqe], `GhostQ` v. 2017-07-26 [JKMC12, ghostq], `QESTO` v. 1.0 [JM15,qesto], and `RAReQS` v. 1.1 [JKMC12,rareqs]. Table 4 shows the numbers of solved instances, and Figure 6 (a)–(f) compare the run times for solving the A2AECC-transformed versus the original instances with `DepQBF` (a), `AIGSolve` (b), `CAQE` (c), `GhostQ` (d), `QESTO` (e), and `RAReQS` (f). We observe that

Table 4. Solving A2AECC-transformed versus original instances: number of (un-)solved instances. The best value per column is in **bold** font, and "vbs" is the virtual best solver.

	original			A2AECC list			A2AECC set-inner		
	unsat	sat	n.s.	unsat	sat	n.s.	unsat	sat	n.s.
`DepQBF`	1911	1293	2138	1556	980	2806	1589	974	2779
`AIGSolve`	1655	**1445**	2242	1663	**1448**	**2231**	1670	**1447**	**2225**
`CAQE`	**2091**	1154	**2097**	1666	921	2755	1413	670	3259
`GhostQ`	1831	1275	2236	**1829**	1280	2233	**1818**	1282	2242
`QESTO`	1793	995	2554	1387	826	3129	1149	528	3665
`RAReQS`	1900	942	2500	1106	519	3717	1201	519	3622
vbs	2496	1866	980	2374	1787	1181	2333	1789	1220

(i) the A2AECC-transformed instances can be solved in many cases, (ii) the over-

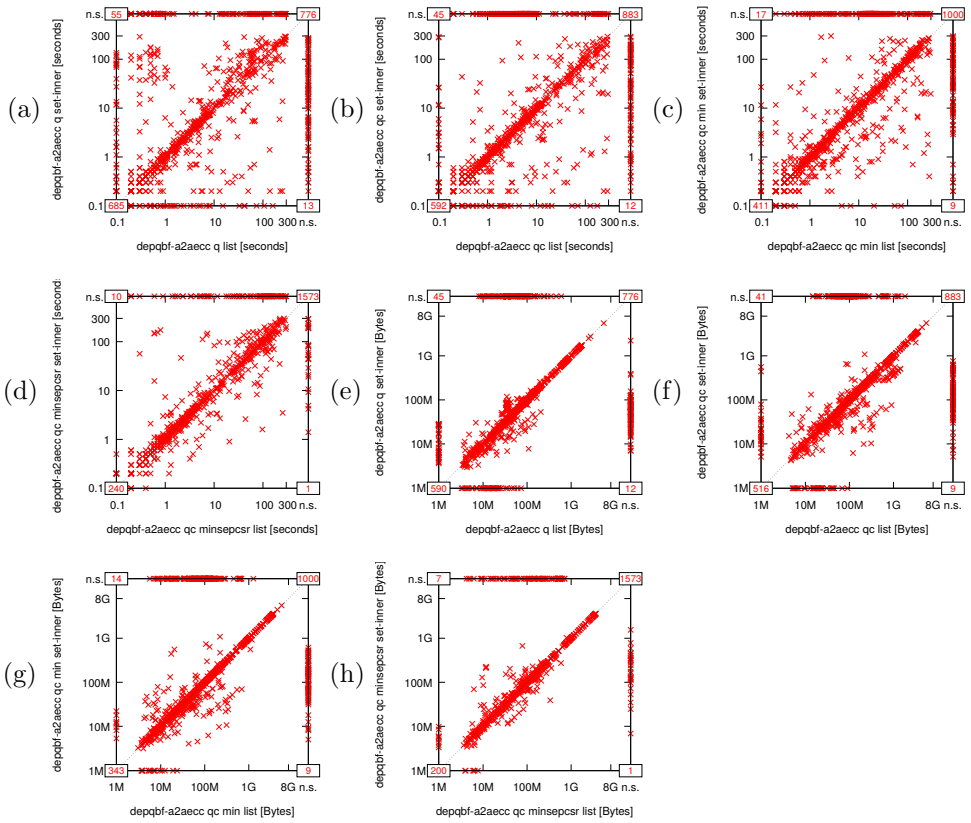

Fig. 5. (a)–(d) Comparing the run times for extracting unsatisfiable cores in list semantics (x-axis) vs. set-inner semantics (y-axis) [seconds]: (a) unsatisfiable q-cores; (b) unsatisfiable qc-cores; (c) q- and c-minimally unsatisfiable qc-cores; (d) q- and c-minimally unsatisfiable qc-cores with separate CSR. (e)–(h): as (a)–(d) but for memory [Bytes].

head 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. Only `AIGSolve` and `QESTO` ran into memory out on a larger number of instances; the number of memory outs reached up to 25 % of the number of time outs. For plots see Figure 6 (g)–(l). 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 (see Figure 7).

We repeated the experiments with set-inner instead of list semantics (see Figure 8). 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.

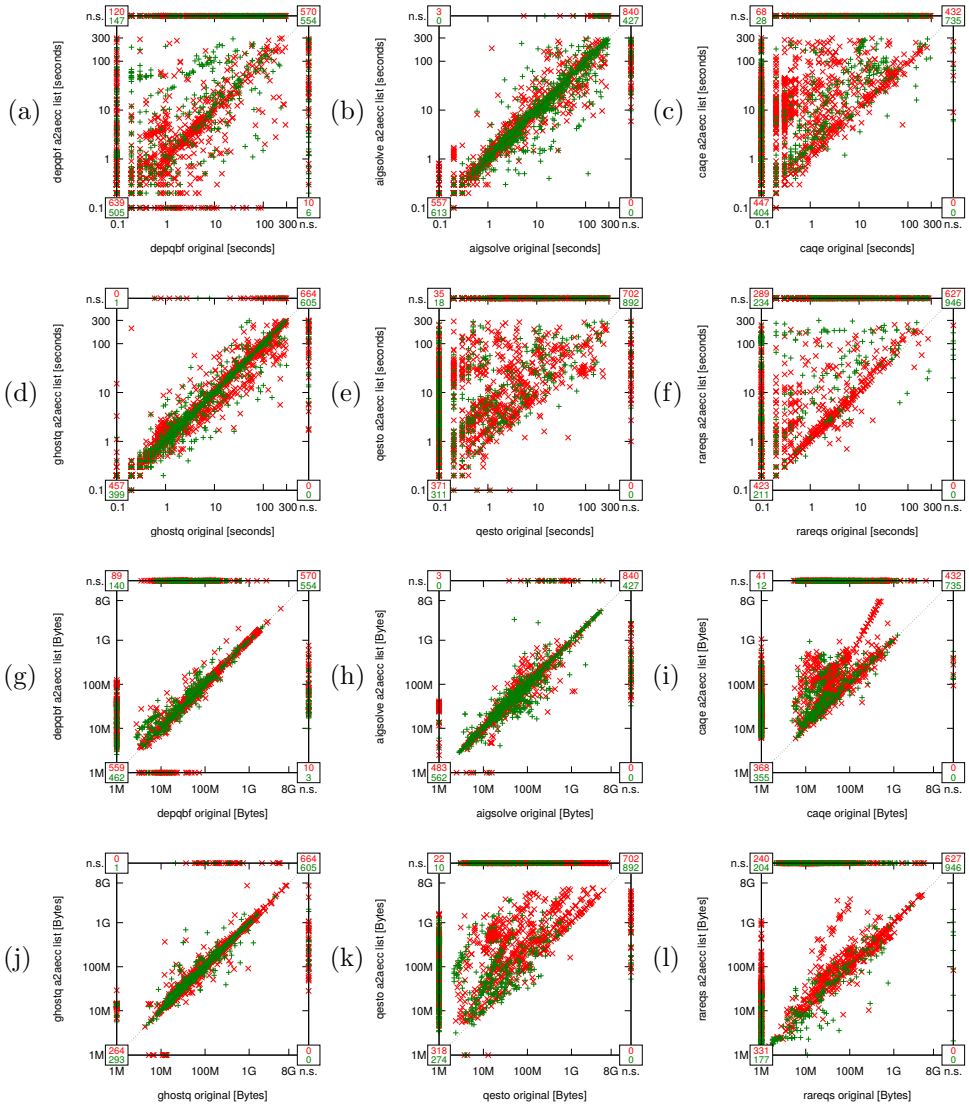

Fig. 6. (a)–(f): Comparing the run times for solving original (x-axis) versus A2AECC-transformed (y-axis) instances [seconds]: (a) `DepQBF`; (b) `AIGSolve`; (c) `CAQE`; (d) `GhostQ`; (e) `QESTO`; (f) `RAReQS`. (g)–(l): as (a)–(f) but for memory [Bytes].

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

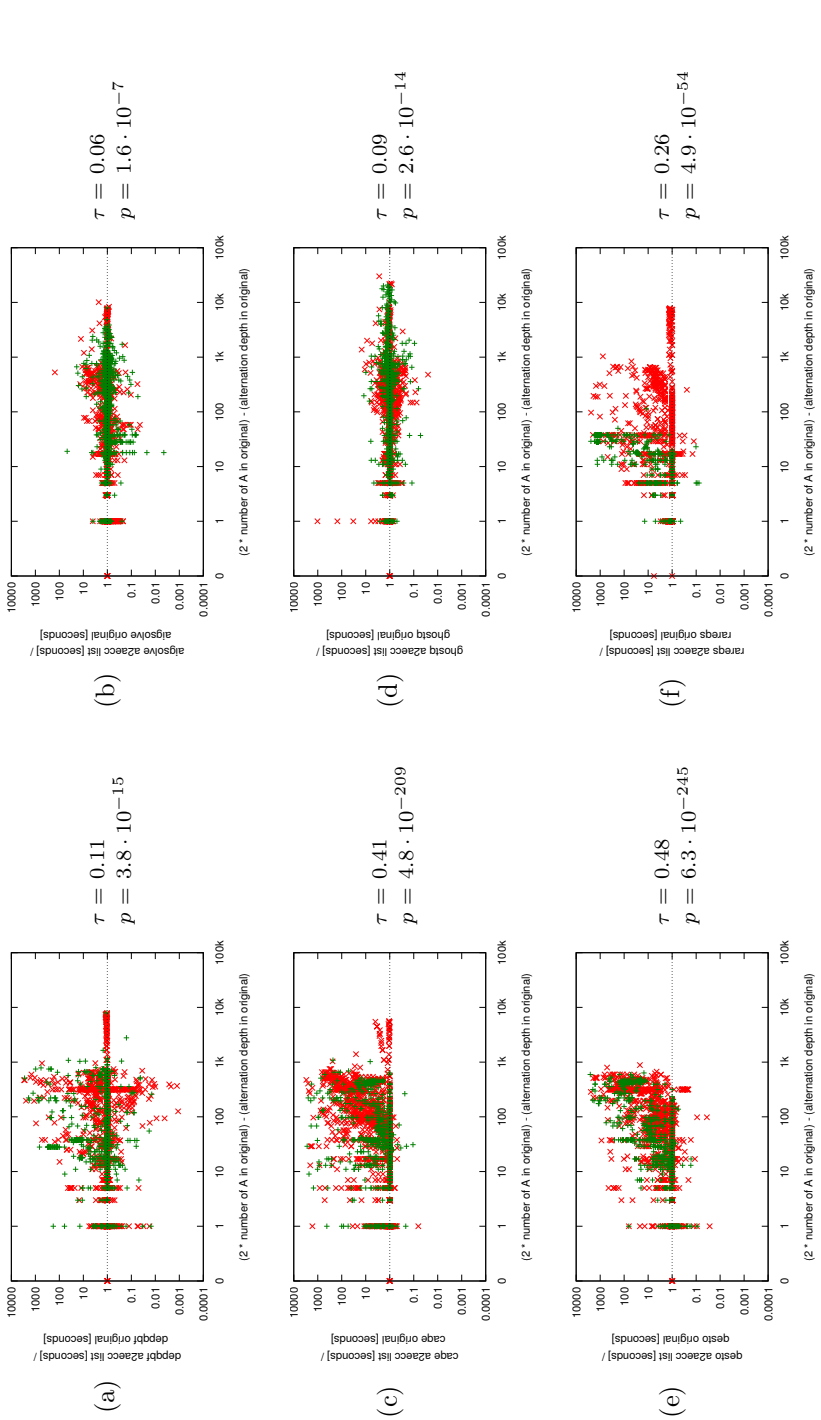

Fig. 7. Ratio of the run times [seconds] between solving A2AECC-transformed and original instances (y-axis) depending on the increase in alternation depth between the original and the A2AECC-transformed instances (x-axis): (a) `DepQBF`; (b) `AIGSolve`; (c) `CAQE`; (d) `GhostQ`; (e) `QESTO`; (f) `RAReQS`. Only pairs for which both instances were solved are included. Using `gbutils` (see [gbutils]) we obtained Kendall's rank correlation coefficient τ at p as shown to the right of the plots.

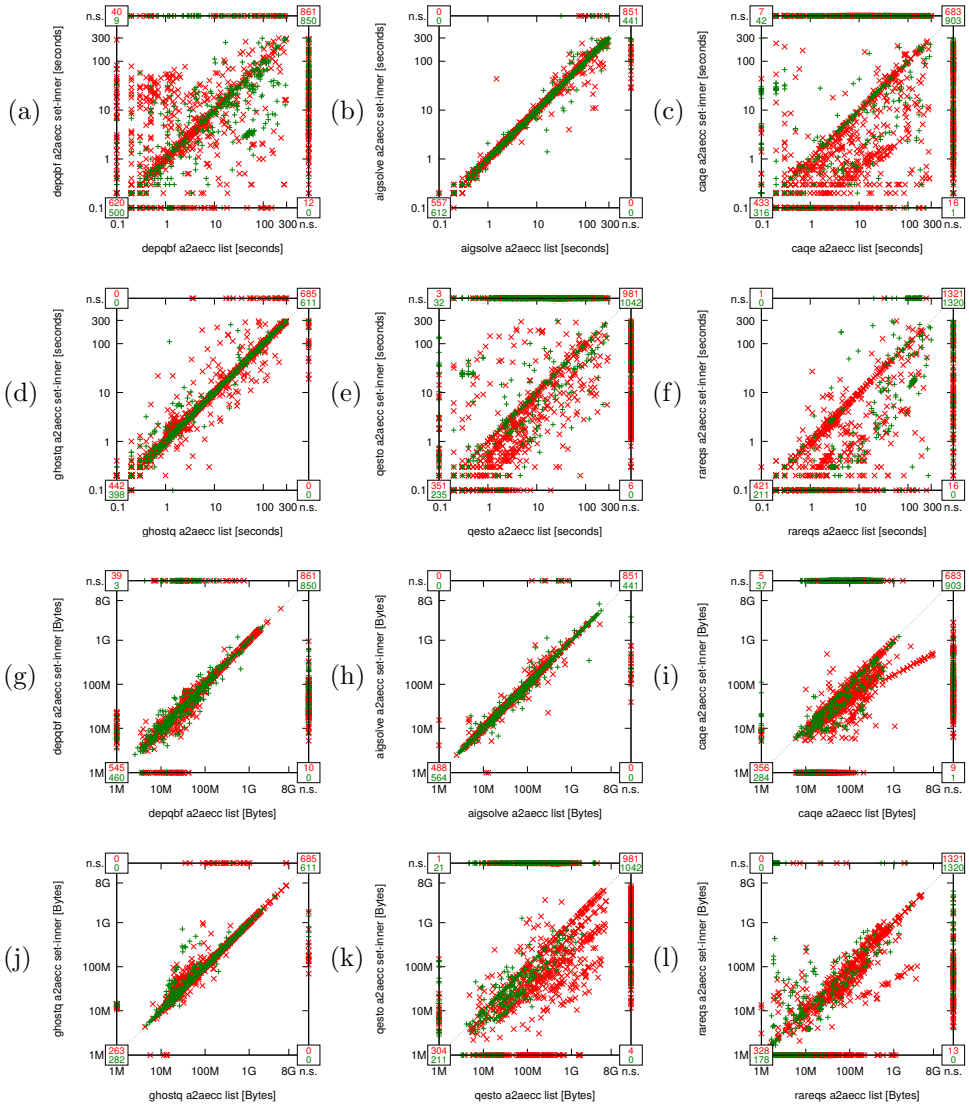

Fig. 8. (a)–(f) Comparing the run times for solving A2AECC-transformed instances in list semantics (x-axis) vs. set-inner semantics (y-axis) [seconds]: (a) `DepQBF`; (b) `AIGSolve`; (c) `CAQE`; (d) `GhostQ`; (e) `QESTO`; (f) `RAReQS`. (g)–(l): as (a)–(f) but for memory [Bytes].

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 9 we show the results. `DepQBF-a2aecc` was faster on 835 instances, while `quantom` was faster on 81 instances, with some large differences both ways.

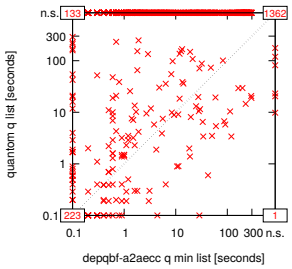

Fig. 9. Comparing the run times for finding q-minimally unsatisfiable q-cores with `DepQBF--a2aecc` (x-axis) with finding minimum cardinality sets of universal quantifiers whose weakening to existential quantifiers results in satisfiability (y-axis) [seconds].

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 [RSMB14] and for providing me with `quantom` [RPSB12]; Alessandro Cimatti for mentioning that proofs can be used to compare formulas; Barry O'Sullivan for pointing out related work [FO07,MOQ15] in QCSP; and the anonymous reviewers for their suggestions on how to improve this paper.

References

[AB00] A. Ayari and D. A. Basin. "Bounded Model Construction for Monadic Second-Order Logics". In: *Computer Aided Verification, 12th International Conference, CAV 2000, Chicago, IL, USA, July 15-19, 2000, Proceedings*. Ed. by E. A. Emerson and A. P. Sistla. Vol. 1855. Lecture Notes in Computer Science. Springer, 2000, pp. 99–112. ISBN: 3-540-67770-4. DOI: 10.1007/10722167_11 (cit. on p. 1).

[AFF+03] R. Armoni, L. Fix, A. Flaisher, O. Grumberg, N. Piterman, A. Tiemeyer, and M. Y. Vardi. "Enhanced Vacuity Detection in Linear Temporal Logic".

In: *Computer Aided Verification, 15th International Conference, CAV 2003, Boulder, CO, USA, July 8-12, 2003, Proceedings*. Ed. by W. A. Hunt Jr. and F. Somenzi. Vol. 2725. Lecture Notes in Computer Science. Springer, 2003, pp. 368–380. ISBN: 3-540-40524-0. DOI: 10.1007/978-3-540-45069-6_35 (cit. on p. 4).

[AGS05] C. Ansótegui, C. P. Gomes, and B. Selman. "The Achilles' Heel of QBF". In: *Proceedings, The Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference, July 9-13, 2005, Pittsburgh, Pennsylvania, USA*. Ed. by M. M. Veloso and S. Kambhampati. AAAI Press / The MIT Press, 2005, pp. 275–281. ISBN: 1-57735-236-X. URL: http://www.aaai.org/Library/AAAI/2005/aaai05-044.php (cit. on p. 1).

[aigsolve] http://abs.informatik.uni-freiburg.de/src/projects_view.php?projectID=19 (cit. on p. 29).

[BCM09] L. Bordeaux, M. Cadoli, and T. Mancini. "Generalizing consistency and other constraint properties to quantified constraints". In: *ACM Trans. Comput. Log.* 10.3 (2009), 17:1–17:25. DOI: 10.1145/1507244.1507247 (cit. on p. 4).

[BG00] B. Bonet and H. Geffner. "Planning with Incomplete Information as Heuristic Search in Belief Space". In: *Proceedings of the Fifth International Conference on Artificial Intelligence Planning Systems, Breckenridge, CO, USA, April 14-17, 2000*. Ed. by S. A. Chien, S. Kambhampati, and C. A. Knoblock. AAAI, 2000, pp. 52–61. ISBN: 1-57735-111-8. URL: http://www.aaai.org/Library/AIPS/2000/aips00-006.php (cit. on p. 21).

[BHS00] P. Balsiger, A. Heuerding, and S. Schwendimann. "A Benchmark Method for the Propositional Modal Logics K, KT, S4". In: *J. Autom. Reasoning* 24.3 (2000), pp. 297–317. DOI: 10.1023/A:1006249507577 (cit. on p. 21).

[BHZ06] L. Bordeaux, Y. Hamadi, and L. Zhang. "Propositional Satisfiability and Constraint Programming: A comparative survey". In: *ACM Comput. Surv.* 38.4 (2006), p. 12. DOI: 10.1145/1177352.1177354 (cit. on p. 4).

[BLB10] R. Brummayer, F. Lonsing, and A. Biere. "Automated Testing and Debugging of SAT and QBF Solvers". In: *Theory and Applications of Satisfiability Testing - SAT 2010, 13th International Conference, SAT 2010, Edinburgh, UK, July 11-14, 2010. Proceedings*. Ed. by O. Strichman and S. Szeider. Vol. 6175. Lecture Notes in Computer Science. Springer, 2010, pp. 44–57. ISBN: 978-3-642-14185-0. DOI: 10.1007/978-3-642-14186-7_6 (cit. on p. 3).

[BLM12] A. Belov, I. Lynce, and J. Marques-Silva. "Towards efficient MUS extraction". In: *AI Commun.* 25.2 (2012), pp. 97–116. DOI: 10.3233/AIC-2012-0523 (cit. on p. 20).

[BLS11] A. Biere, F. Lonsing, and M. Seidl. "Blocked Clause Elimination for QBF". In: *Automated Deduction - CADE-23 - 23rd International Conference on Automated Deduction, Wroclaw, Poland, July 31 - August 5, 2011. Proceedings*. Ed. by N. Bjørner and V. Sofronie-Stokkermans. Vol. 6803. Lecture Notes in Computer Science. Springer, 2011, pp. 101–115. ISBN: 978-3-642-22437-9. DOI: 10.1007/978-3-642-22438-6_10 (cit. on pp. 15, 22).

[BS01] R. Bruni and A. Sassano. "Restoring Satisfiability or Maintaining Unsatisfiability by finding small Unsatisfiable Subformulae". In: *Electronic Notes in Discrete Mathematics* 9 (2001), pp. 162–173. DOI: 10.1016/S1571-0653(04)00320-8 (cit. on pp. 1, 6).

[caqe] https://www.react.uni-saarland.de/tools/caqe/ (cit. on p. 29).

[CD91] J. W. Chinneck and E. W. Dravnieks. "Locating Minimal Infeasible Constraint Sets in Linear Programs". In: *INFORMS Journal on Computing* 3.2 (1991), pp. 157–168. DOI: 10.1287/ijoc.3.2.157 (cit. on pp. 1, 6).

[CEG97] M. Cadoli, T. Eiter, and G. Gottlob. "Default Logic as a Query Language". In: *IEEE Trans. Knowl. Data Eng.* 9.3 (1997), pp. 448–463. DOI: 10.1109/69.599933 (cit. on p. 21).

[CFL+06] S. Coste-Marquis, H. Fargier, J. Lang, D. Le Berre, and P. Marquis. "Representing Policies for Quantified Boolean Formulae". In: *Proceedings, Tenth International Conference on Principles of Knowledge Representation and Reasoning, Lake District of the United Kingdom, June 2-5, 2006*. Ed. by P. Doherty, J. Mylopoulos, and C. A. Welty. AAAI Press, 2006, pp. 286–297. ISBN: 978-1-57735-271-6. URL: http://www.aaai.org/Library/KR/2006/kr06-031.php (cit. on p. 7).

[CFLS93] A. Condon, J. Feigenbaum, C. Lund, and P. W. Shor. "Probabilistically checkable debate systems and approximation algorithms for PSPACE-hard functions". In: *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing, May 16-18, 1993, San Diego, CA, USA*. Ed. by S. R. Kosaraju, D. S. Johnson, and A. Aggarwal. ACM, 1993, pp. 305–314. ISBN: 0-89791-591-7. DOI: 10.1145/167088.167190 (cit. on p. 3).

[Che04] H. M. Chen. "The Computational Complexity of Quantified Constraint Satisfaction". PhD thesis. Cornell University, 2004 (cit. on p. 3).

[depqbf] http://lonsing.github.io/depqbf/ (cit. on p. 29).

[DQF14] J. Du, G. Qi, and X. Fu. "A Practical Fine-grained Approach to Resolving Incoherent OWL 2 DL Terminologies". In: *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM 2014, Shanghai, China, November 3-7, 2014*. Ed. by J. Li, X. S. Wang, M. N. Garofalakis, I. Soboroff, T. Suel, and M. Wang. ACM, 2014, pp. 919–928. ISBN: 978-1-4503-2598-1. DOI: 10.1145/2661829.2662046 (cit. on p. 4).

[EETW00] U. Egly, T. Eiter, H. Tompits, and S. Woltran. "Solving Advanced Reasoning Tasks Using Quantified Boolean Formulas". In: *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on on Innovative Applications of Artificial Intelligence, July 30 - August 3, 2000, Austin, Texas, USA*. Ed. by H. A. Kautz and B. W. Porter. AAAI Press / The MIT Press, 2000, pp. 417–422. ISBN: 0-262-51112-6. URL: http://www.aaai.org/Library/AAAI/2000/aaai00-064.php (cit. on p. 1).

[FLMR07] W. Faber, N. Leone, M. Maratea, and F. Ricca. "Looking Back in DLV: Experiments and Comparison to QBF Solvers". In: *Answer Set Programming, 4th International Workshop, ASP 2007, Porto, Portugal, September 8 and 13, 2007, Proceedings*. 2007 (cit. on p. 21).

[FO07] A. Ferguson and B. O'Sullivan. "Quantified Constraint Satisfaction Problems: From Relaxations to Explanations". In: *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*. Ed. by M. M. Veloso. 2007, pp. 74–79. URL: http://ijcai.org/Proceedings/07/Papers/010.pdf (cit. on pp. 3, 4, 34).

[Fra14] N. Francez. "The Granularity of Meaning in Proof-Theoretic Semantics". In: *Logical Aspects of Computational Linguistics - 8th International Conference, LACL 2014, Toulouse, France, June 18-20, 2014. Proceedings*. Ed. by N. Asher and S. Soloviev. Vol. 8535. Lecture Notes in Computer Science.

Springer, 2014, pp. 96–106. ISBN: 978-3-662-43741-4. DOI: 10.1007/978-3-662-43742-1_8 (cit. on p. 7).

[gbutils] http://cafim.sssup.it/~giulio/software/gbutils/ (cit. on pp. 29, 32).

[GC04] A. Gurfinkel and M. Chechik. "How Vacuous Is Vacuous?" In: *Tools and Algorithms for the Construction and Analysis of Systems, 10th International Conference, TACAS 2004, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2004, Barcelona, Spain, March 29 - April 2, 2004, Proceedings*. Ed. by K. Jensen and A. Podelski. Vol. 2988. Lecture Notes in Computer Science. Springer, 2004, pp. 451–466. ISBN: 3-540-21299-X. DOI: 10.1007/978-3-540-24730-2_34 (cit. on p. 4).

[Gel12] A. V. Gelder. "Contributions to the Theory of Practical Quantified Boolean Formula Solving". In: *Principles and Practice of Constraint Programming - 18th International Conference, CP 2012, Québec City, QC, Canada, October 8-12, 2012. Proceedings*. Ed. by M. Milano. Vol. 7514. Lecture Notes in Computer Science. Springer, 2012, pp. 647–663. ISBN: 978-3-642-33557-0. DOI: 10.1007/978-3-642-33558-7_47 (cit. on p. 7).

[ghostq] https://www.wklieber.com/ghostq/ (cit. on p. 29).

[GMN09] E. Giunchiglia, P. Marin, and M. Narizzano. "Reasoning with Quantified Boolean Formulas". In: *Handbook of Satisfiability*. Ed. by A. Biere, M. Heule, H. van Maaren, and T. Walsh. Vol. 185. Frontiers in Artificial Intelligence and Applications. IOS Press, 2009, pp. 761–780. ISBN: 978-1-58603-929-5. DOI: 10.3233/978-1-58603-929-5-761 (cit. on pp. 1, 4, 19).

[GMP07] É. Grégoire, B. Mazure, and C. Piette. "MUST: Provide a Finer-Grained Explanation of Unsatisfiability". In: *Principles and Practice of Constraint Programming - CP 2007, 13th International Conference, CP 2007, Providence, RI, USA, September 23-27, 2007, Proceedings*. Ed. by C. Bessiere. Vol. 4741. Lecture Notes in Computer Science. Springer, 2007, pp. 317–331. ISBN: 978-3-540-74969-1. DOI: 10.1007/978-3-540-74970-7_24 (cit. on p. 4).

[GNPT] E. Giunchiglia, M. Narizzano, L. Pulina, and A. Tacchella. *Quantified Boolean Formulas satisfiability library (QBFLIB)*. http://www.qbflib.org/ (cit. on pp. 2, 20, 22).

[GR03] I. P. Gent and A. G. D. Rowley. "Encoding Connect-4 Using Quantified Boolean Formulae". In: *Modelling and Reformulating Constraint Satisfaction Problems: Towards Systematisation and Automation, Second International Workshop, Kinsale Ireland, September 2003, Proceedings*. Ed. by A. M. Frisch. 2003, pp. 78–93. URL: https://www-users.cs.york.ac.uk/frisch/Reformulation/03/proceedings.pdf#page=84 (cit. on pp. 1, 2, 20, 21).

[GW11] S. Grimm and J. Wissmann. "Elimination of Redundancy in Ontologies". In: *The Semantic Web: Research and Applications - 8th Extended Semantic Web Conference, ESWC 2011, Heraklion, Crete, Greece, May 29-June 2, 2011, Proceedings, Part I*. Ed. by G. Antoniou, M. Grobelnik, E. P. B. Simperl, B. Parsia, D. Plexousakis, P. D. Leenheer, and J. Z. Pan. Vol. 6643. Lecture Notes in Computer Science. Springer, 2011, pp. 260–274. ISBN: 978-3-642-21033-4. DOI: 10.1007/978-3-642-21034-1_18 (cit. on p. 4).

[HPS08] M. Horridge, B. Parsia, and U. Sattler. "Laconic and Precise Justifications in OWL". In: *The Semantic Web - ISWC 2008, 7th International Semantic Web Conference, ISWC 2008, Karlsruhe, Germany, October 26-30, 2008. Proceedings*. Ed. by A. P. Sheth, S. Staab, M. Dean, M. Paolucci, D. Maynard, T. W. Finin, and K. Thirunarayan. Vol. 5318. Lecture Notes in Computer

Science. Springer, 2008, pp. 323–338. ISBN: 978-3-540-88563-4. DOI: 10.1007/978-3-540-88564-1_21 (cit. on p. 4).

[IJM13] A. Ignatiev, M. Janota, and J. Marques-Silva. "Quantified Maximum Satisfiability: A Core-Guided Approach". In: *Theory and Applications of Satisfiability Testing - SAT 2013 - 16th International Conference, Helsinki, Finland, July 8-12, 2013. Proceedings*. Ed. by M. Järvisalo and A. V. Gelder. Vol. 7962. Lecture Notes in Computer Science. Springer, 2013, pp. 250–266. ISBN: 978-3-642-39070-8. DOI: 10.1007/978-3-642-39071-5_19 (cit. on pp. 2, 3).

[JKMC12] M. Janota, W. Klieber, J. Marques-Silva, and E. M. Clarke. "Solving QBF with Counterexample Guided Refinement". In: *Theory and Applications of Satisfiability Testing - SAT 2012 - 15th International Conference, Trento, Italy, June 17-20, 2012. Proceedings*. Ed. by A. Cimatti and R. Sebastiani. Vol. 7317. Lecture Notes in Computer Science. Springer, 2012, pp. 114–128. ISBN: 978-3-642-31611-1. DOI: 10.1007/978-3-642-31612-8_10 (cit. on p. 29).

[JM15] M. Janota and J. Marques-Silva. "Solving QBF by Clause Selection". In: *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*. Ed. by Q. Yang and M. Wooldridge. AAAI Press, 2015, pp. 325–331. ISBN: 978-1-57735-738-4. URL: http://ijcai.org/Abstract/15/052 (cit. on p. 29).

[Jun01] U. Junker. "QuickXplain: Conflict Detection for Arbitrary Constraint Propagation Algorithms". In: *Modelling and Solving Problems with Constraints, Workshop, held at the International Joint Conference on Artificial Intelligence (IJCAI-2001), Seattle WA, August 5-6, 2001, Proceedings*. 2001. URL: http://www.lirmm.fr/~bessiere/ws_ijcai01/junker.ps.gz (cit. on p. 4).

[Jun04] U. Junker. "QUICKXPLAIN: Preferred Explanations and Relaxations for Over-Constrained Problems". In: *Proceedings of the Nineteenth National Conference on Artificial Intelligence, Sixteenth Conference on Innovative Applications of Artificial Intelligence, July 25-29, 2004, San Jose, California, USA*. Ed. by D. L. McGuinness and G. Ferguson. AAAI Press / The MIT Press, 2004, pp. 167–172. ISBN: 0-262-51183-5. URL: http://www.aaai.org/Library/AAAI/2004/aaai04-027.php (cit. on p. 4).

[KB09] H. Kleine Büning and U. Bubeck. "Theory of Quantified Boolean Formulas". In: *Handbook of Satisfiability*. Ed. by A. Biere, M. Heule, H. van Maaren, and T. Walsh. Vol. 185. Frontiers in Artificial Intelligence and Applications. IOS Press, 2009, pp. 735–760. ISBN: 978-1-58603-929-5. DOI: 10.3233/978-1-58603-929-5-735 (cit. on p. 4).

[KKF95] H. Kleine Büning, M. Karpinski, and A. Flögel. "Resolution for Quantified Boolean Formulas". In: *Inf. Comput.* 117.1 (1995), pp. 12–18. DOI: 10.1006/inco.1995.1025 (cit. on pp. 7, 8).

[KLM06] 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: *Theory and Applications of Satisfiability Testing - SAT 2006, 9th International Conference, Seattle, WA, USA, August 12-15, 2006, Proceedings*. Ed. by A. Biere and C. P. Gomes. Vol. 4121. Lecture Notes in Computer Science. Springer, 2006, pp. 22–35. ISBN: 3-540-37206-7. DOI: 10.1007/11814948_4 (cit. on pp. 1, 4, 7).

[KPG06] A. Kalyanpur, B. Parsia, and B. C. Grau. "Beyond Asserted Axioms: Fine-Grain Justifications for OWL-DL Entailments". In: *Proceedings of the 2006 International Workshop on Description Logics (DL2006), Windermere, Lake*

District, UK, May 30 - June 1, 2006. Ed. by B. Parsia, U. Sattler, and D. Toman. Vol. 189. CEUR Workshop Proceedings. CEUR-WS.org, 2006. URL: http://ceur-ws.org/Vol-189/submission_30.pdf (cit. on p. 4).

[KPSG06] A. Kalyanpur, B. Parsia, E. Sirin, and B. C. Grau. "Repairing Unsatisfiable Concepts in OWL Ontologies". In: *The Semantic Web: Research and Applications, 3rd European Semantic Web Conference, ESWC 2006, Budva, Montenegro, June 11-14, 2006, Proceedings*. Ed. by Y. Sure and J. Domingue. Vol. 4011. Lecture Notes in Computer Science. Springer, 2006, pp. 170–184. ISBN: 3-540-34544-2. DOI: 10.1007/11762256_15 (cit. on pp. 4, 11).

[KZ06] H. Kleine Büning and X. Zhao. "Minimal False Quantified Boolean Formulas". In: *Theory and Applications of Satisfiability Testing - SAT 2006, 9th International Conference, Seattle, WA, USA, August 12-15, 2006, Proceedings*. Ed. by A. Biere and C. P. Gomes. Vol. 4121. Lecture Notes in Computer Science. Springer, 2006, pp. 339–352. ISBN: 3-540-37206-7. DOI: 10.1007/11814948_32 (cit. on pp. 2, 3, 10).

[LB11] F. Lonsing and A. Biere. "Failed Literal Detection for QBF". In: *Theory and Applications of Satisfiability Testing - SAT 2011 - 14th International Conference, SAT 2011, Ann Arbor, MI, USA, June 19-22, 2011. Proceedings*. Ed. by K. A. Sakallah and L. Simon. Vol. 6695. Lecture Notes in Computer Science. Springer, 2011, pp. 259–272. ISBN: 978-3-642-21580-3. DOI: 10.1007/978-3-642-21581-0_21 (cit. on p. 3).

[LE15] F. Lonsing and U. Egly. "Incrementally Computing Minimal Unsatisfiable Cores of QBFs via a Clause Group Solver API". In: *Theory and Applications of Satisfiability Testing - SAT 2015 - 18th International Conference, Austin, TX, USA, September 24-27, 2015, Proceedings*. Ed. by M. Heule and S. Weaver. Vol. 9340. Lecture Notes in Computer Science. Springer, 2015, pp. 191–198. ISBN: 978-3-319-24317-7. DOI: 10.1007/978-3-319-24318-4_14 (cit. on pp. 2, 3, 6, 15, 20).

[LE17] F. Lonsing and U. Egly. "DepQBF 6.0: A Search-Based QBF Solver Beyond Traditional QCDCL". In: *Automated Deduction - CADE 26 - 26th International Conference on Automated Deduction, Gothenburg, Sweden, August 6-11, 2017, Proceedings*. Ed. by L. de Moura. Vol. 10395. Lecture Notes in Computer Science. Springer, 2017, pp. 371–384. ISBN: 978-3-319-63045-8. DOI: 10.1007/978-3-319-63046-5_23 (cit. on pp. 2, 19, 29).

[LES16] F. Lonsing, U. Egly, and M. Seidl. "Q-Resolution with Generalized Axioms". In: *Theory and Applications of Satisfiability Testing - SAT 2016 - 19th International Conference, Bordeaux, France, July 5-8, 2016, Proceedings*. Ed. by N. Creignou and D. Le Berre. Vol. 9710. Lecture Notes in Computer Science. Springer, 2016, pp. 435–452. ISBN: 978-3-319-40969-6. DOI: 10.1007/978-3-319-40970-2_27 (cit. on p. 3).

[LM04] I. Lynce and J. P. Marques-Silva. "On Computing Minimum Unsatisfiable Cores". In: *SAT 2004 - The Seventh International Conference on Theory and Applications of Satisfiability Testing, 10-13 May 2004, Vancouver, BC, Canada, Online Proceedings*. 2004. URL: http://www.satisfiability.org/SAT04/programme/110.pdf (cit. on p. 6).

[LPF+06] N. Leone, G. Pfeifer, W. Faber, T. Eiter, G. Gottlob, S. Perri, and F. Scarcello. "The DLV system for knowledge representation and reasoning". In: *ACM Trans. Comput. Log.* 7.3 (2006), pp. 499–562. DOI: 10.1145/1149114.1149117 (cit. on p. 21).

[LPSV06] S. C. Lam, J. Z. Pan, D. H. Sleeman, and W. W. Vasconcelos. "A Fine-Grained Approach to Resolving Unsatisfiable Ontologies". In: *2006 IEEE / WIC / ACM International Conference on Web Intelligence (WI 2006), 18-22 December 2006, Hong Kong, China*. IEEE Computer Society, 2006, pp. 428–434. ISBN: 0-7695-2747-7. DOI: 10.1109/WI.2006.11 (cit. on p. 4).

[LS08] M. H. Liffiton and K. A. Sakallah. "Algorithms for Computing Minimal Unsatisfiable Subsets of Constraints". In: *J. Autom. Reasoning* 40.1 (2008), pp. 1–33. DOI: 10.1007/s10817-007-9084-z (cit. on p. 15).

[Mar12] J. Marques-Silva. "Computing Minimally Unsatisfiable Subformulas: State of the Art and Future Directions". In: *Multiple-Valued Logic and Soft Computing* 19.1-3 (2012), pp. 163–183. URL: http://www.oldcitypublishing.com/MVLSC/MVLSCabstracts/MVLSC18.5-6abstracts/MVLSCv18n5-6p617-636Li.html (cit. on pp. 3, 4, 20).

[MJ14] J. Marques-Silva and M. Janota. "Computing Minimal Sets on Propositional Formulae I: Problems & Reductions". In: *CoRR* abs/1402.3011 (2014). URL: http://arxiv.org/abs/1402.3011 (cit. on p. 4).

[MOQ15] D. Mehta, B. O'Sullivan, and L. Quesada. "Extending the Notion of Preferred Explanations for Quantified Constraint Satisfaction Problems". In: *Theoretical Aspects of Computing - ICTAC 2015 - 12th International Colloquium Cali, Colombia, October 29-31, 2015, Proceedings*. Ed. by M. Leucker, C. Rueda, and F. D. Valencia. Vol. 9399. Lecture Notes in Computer Science. Springer, 2015, pp. 309–327. ISBN: 978-3-319-25149-3. DOI: 10.1007/978-3-319-25150-9_19 (cit. on pp. 4, 34).

[Nad10] A. Nadel. "Boosting minimal unsatisfiable core extraction". In: *Proceedings of 10th International Conference on Formal Methods in Computer-Aided Design, FMCAD 2010, Lugano, Switzerland, October 20-23*. Ed. by R. Bloem and N. Sharygina. IEEE, 2010, pp. 221–229. ISBN: 978-1-4577-0734-6. URL: http://ieeexplore.ieee.org/document/5770953/ (cit. on p. 15).

[NRS14] A. Nadel, V. Ryvchin, and O. Strichman. "Accelerated Deletion-based Extraction of Minimal Unsatisfiable Cores". In: *JSAT* 9 (2014), pp. 27–51. URL: https://satassociation.org/jsat/index.php/jsat/article/view/116 (cit. on p. 15).

[PQ13] I. Pill and T. Quaritsch. "Behavioral Diagnosis of LTL Specifications at Operator Level". In: *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*. Ed. by F. Rossi. IJCAI/AAAI, 2013, pp. 1053–1059. ISBN: 978-1-57735-633-2. URL: http://www.aaai.org/ocs/index.php/IJCAI/IJCAI13/paper/view/6595 (cit. on p. 4).

[PS10] F. Pigorsch and C. Scholl. "An AIG-Based QBF-solver using SAT for preprocessing". In: *Proceedings of the 47th Design Automation Conference, DAC 2010, Anaheim, California, USA, July 13-18, 2010*. Ed. by S. S. Sapatnekar. ACM, 2010, pp. 170–175. ISBN: 978-1-4503-0002-5. DOI: 10.1145/1837274.1837318 (cit. on p. 29).

[PV03] G. Pan and M. Y. Vardi. "Optimizing a BDD-Based Modal Solver". In: *Automated Deduction - CADE-19, 19th International Conference on Automated Deduction Miami Beach, FL, USA, July 28 - August 2, 2003, Proceedings*. Ed. by F. Baader. Vol. 2741. Lecture Notes in Computer Science. Springer, 2003, pp. 75–89. ISBN: 3-540-40559-3. DOI: 10.1007/978-3-540-45085-6_7 (cit. on pp. 1, 2, 21).

[qbfdd] https://github.com/aniemetz/qbfdd (cit. on p. 3).

[qesto] http://sat.inesc-id.pt/~mikolas/sw/qesto/ (cit. on p. 29).

[rareqs] http://sat.inesc-id.pt/~mikolas/sw/areqs/ (cit. on p. 29).

[Rei87] R. Reiter. "A Theory of Diagnosis from First Principles". In: *Artif. Intell.* 32.1 (1987), pp. 57–95. DOI: 10.1016/0004-3702(87)90062-2 (cit. on pp. 1, 11).

[Rin07] J. Rintanen. "Asymptotically Optimal Encodings of Conformant Planning in QBF". In: *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, July 22-26, 2007, Vancouver, British Columbia, Canada*. AAAI Press, 2007, pp. 1045–1050. ISBN: 978-1-57735-323-2. URL: http://www.aaai.org/Library/AAAI/2007/aaai07-166.php (cit. on pp. 2, 21).

[Rin99] J. Rintanen. "Constructing Conditional Plans by a Theorem-Prover". In: *J. Artif. Intell. Res.* 10 (1999), pp. 323–352. DOI: 10.1613/jair.591 (cit. on p. 1).

[RPSB12] S. Reimer, F. Pigorsch, C. Scholl, and B. Becker. "Enhanced Integration of QBF Solving Techniques". In: *Methoden und Beschreibungssprachen zur Modellierung und Verifikation von Schaltungen und Systemen (MBMV), Kaiserslautern, Germany, March 5-7, 2012*. Ed. by J. Brandt and K. Schneider. Verlag Dr. Kovac, 2012, pp. 133–143 (cit. on pp. 3, 34).

[RSMB14] S. Reimer, M. Sauer, P. Marin, and B. Becker. "QBF with Soft Variables". In: *ECEASST* 70 (2014). URL: http://journal.ub.tu-berlin.de/eceasst/article/view/973 (cit. on pp. 3, 34).

[SB01] C. Scholl and B. Becker. "Checking Equivalence for Partial Implementations". In: *Proceedings of the 38th Design Automation Conference, DAC 2001, Las Vegas, NV, USA, June 18-22, 2001*. ACM, 2001, pp. 238–243. ISBN: 1-58113-297-2. DOI: 10.1145/378239.378471 (cit. on p. 1).

[SC03] S. Schlobach and R. Cornet. "Non-Standard Reasoning Services for the Debugging of Description Logic Terminologies". In: *IJCAI-03, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, Mexico, August 9-15, 2003*. Ed. by G. Gottlob and T. Walsh. Morgan Kaufmann, 2003, pp. 355–362. URL: http://ijcai.org/Proceedings/03/Papers/053.pdf (cit. on pp. 1, 6).

[Sch05] S. Schlobach. "Debugging and Semantic Clarification by Pinpointing". In: *The Semantic Web: Research and Applications, Second European Semantic Web Conference, ESWC 2005, Heraklion, Crete, Greece, May 29 - June 1, 2005, Proceedings*. Ed. by A. Gómez-Pérez and J. Euzenat. Vol. 3532. Lecture Notes in Computer Science. Springer, 2005, pp. 226–240. ISBN: 3-540-26124-9. DOI: 10.1007/11431053_16 (cit. on p. 2).

[Sch12] V. Schuppan. "Towards a notion of unsatisfiable and unrealizable cores for LTL". In: *Sci. Comput. Program.* 77.7-8 (2012), pp. 908–939. DOI: 10.1016/j.scico.2010.11.004 (cit. on pp. 1, 4, 6).

[Sch16a] V. Schuppan. "Enhancing unsatisfiable cores for LTL with information on temporal relevance". In: *Theor. Comput. Sci.* 655, Part B (2016), pp. 155–192. DOI: 10.1016/j.tcs.2016.01.014 (cit. on p. 4).

[Sch16b] V. Schuppan. "Extracting unsatisfiable cores for LTL via temporal resolution". In: *Acta Inf.* 53.3 (2016), pp. 247–299. DOI: 10.1007/s00236-015-0242-1 (cit. on p. 4).

[Sch18] V. Schuppan. "Enhanced Unsatisfiable Cores for QBF: Weakening Universal to Existential Quantifiers". In: *IEEE 30th International Conference on Tools with Artificial Intelligence, ICTAI 2018, 5-7 November 2018, Volos, Greece.*

Ed. by L. H. Tsoukalas, É. Grégoire, and M. Alamaniotis. IEEE, 2018, pp. 81–89. DOI: 10.1109/ICTAI.2018.00023 (cit. on p. 2).

[Sla14] J. Slaney. "Set-theoretic duality: A fundamental feature of combinatorial optimisation". In: *ECAI 2014 - 21st European Conference on Artificial Intelligence, 18-22 August 2014, Prague, Czech Republic - Including Prestigious Applications of Intelligent Systems (PAIS 2014)*. Ed. by T. Schaub, G. Friedrich, and B. O'Sullivan. Vol. 263. Frontiers in Artificial Intelligence and Applications. IOS Press, 2014, pp. 843–848. ISBN: 978-1-61499-418-3. DOI: 10.3233/978-1-61499-419-0-843 (cit. on pp. 11, 16).

[SM73] L. J. Stockmeyer and A. R. Meyer. "Word Problems Requiring Exponential Time: Preliminary Report". In: *Proceedings of the 5th Annual ACM Symposium on Theory of Computing, April 30 - May 2, 1973, Austin, Texas, USA*. Ed. by A. V. Aho, A. Borodin, R. L. Constable, R. W. Floyd, M. A. Harrison, R. M. Karp, and H. R. Strong. ACM, 1973, pp. 1–9. DOI: 10.1145/800125.804029 (cit. on p. 5).

[SSJ+03] I. Shlyakhter, R. Seater, D. Jackson, M. Sridharan, and M. Taghdiri. "Debugging Overconstrained Declarative Models Using Unsatisfiable Cores". In: *18th IEEE International Conference on Automated Software Engineering (ASE 2003), 6-10 October 2003, Montreal, Canada*. IEEE Computer Society, 2003, pp. 94–105. ISBN: 0-7695-2035-9. DOI: 10.1109/ASE.2003.1240298 (cit. on pp. 1, 4, 6).

[Sto76] L. J. Stockmeyer. "The Polynomial-Time Hierarchy". In: *Theor. Comput. Sci.* 3.1 (1976), pp. 1–22. DOI: 10.1016/0304-3975(76)90061-X (cit. on pp. 5, 14).

[TCJ08] E. Torlak, F. S. Chang, and D. Jackson. "Finding Minimal Unsatisfiable Cores of Declarative Specifications". In: *FM 2008: Formal Methods, 15th International Symposium on Formal Methods, Turku, Finland, May 26-30, 2008, Proceedings*. Ed. by J. Cuéllar, T. S. E. Maibaum, and K. Sere. Vol. 5014. Lecture Notes in Computer Science. Springer, 2008, pp. 326–341. ISBN: 978-3-540-68235-6. DOI: 10.1007/978-3-540-68237-0_23 (cit. on p. 6).

[Ten17] L. Tentrup. "On Expansion and Resolution in CEGAR Based QBF Solving". In: *Computer Aided Verification - 29th International Conference, CAV 2017, Heidelberg, Germany, July 24-28, 2017, Proceedings, Part II*. Ed. by R. Majumdar and V. Kuncak. Vol. 10427. Lecture Notes in Computer Science. Springer, 2017, pp. 475–494. ISBN: 978-3-319-63389-3. DOI: 10.1007/978-3-319-63390-9_25 (cit. on p. 29).

[Tur02] H. Turner. "Polynomial-Length Planning Spans the Polynomial Hierarchy". In: *Logics in Artificial Intelligence, European Conference, JELIA 2002, Cosenza, Italy, September, 23-26, Proceedings*. Ed. by S. Flesca, S. Greco, N. Leone, and G. Ianni. Vol. 2424. Lecture Notes in Computer Science. Springer, 2002, pp. 111–124. ISBN: 3-540-44190-5. DOI: 10.1007/3-540-45757-7_10 (cit. on p. 1).

[Wra76] C. Wrathall. "Complete Sets and the Polynomial-Time Hierarchy". In: *Theor. Comput. Sci.* 3.1 (1976), pp. 23–33. DOI: 10.1016/0304-3975(76)90062-1 (cit. on pp. 5, 14).

[YM05] Y. Yu and S. Malik. "Validating the result of a Quantified Boolean Formula (QBF) solver: theory and practice". In: *Proceedings of the 2005 Conference on Asia South Pacific Design Automation, ASP-DAC 2005, Shanghai, China, January 18-21, 2005*. Ed. by T. Tang. ACM Press, 2005, pp. 1047–1051. ISBN: 0-7803-8737-6. DOI: 10.1145/1120725.1120821 (cit. on pp. 1–3, 6).

[ZSM03] H. Zhang, H. Shen, and F. Manyà. "Exact Algorithms for MAX-SAT". In: *Electr. Notes Theor. Comput. Sci.* 86.1 (2003), pp. 190–203. DOI: 10.1016/S1571-0661(04)80663-7 (cit. on p. 3).

Appendix A. Selection of Benchmarks

We briefly describe how we obtained our set of benchmarks instances. There were three main goals.

(1) The set of benchmarks instances should be randomized at the lowest (leaf-) level of the benchmark family tree.
(2) The set of benchmarks instances should give equal weight to the immediate subfamilies of each (inner) node of the benchmark family tree, as long as each subfamily has enough members.
(3) It should be possible to continue supplying slices of benchmark instances to the server running the experiments until the time for running the experiments is up — without knowing in advance when that is or how many slices will be run until then.

To achieve these goals we created a queue of benchmark instances as described below that, when traversed from head to any point between head and tail, ensures the first two goals to a reasonable extent. We then split the benchmark queue into slices of size 25 and fed the slices in ascending order to the server running the experiments. We ran slices 1 through 214. Finally, we removed 8 benchmark instances in which at least one variable occurred both universally and existentially quantified in the prefix. For full details we refer to our experimental data, which includes the shell scripts used, and which is available from http://schuppan.de/viktor/ijait20/.

Figure 10 shows the algorithm used to generate the queue of benchmark instances. It is called with the root node of the benchmark family tree as argument. In that tree each node represents a (sub)family of benchmark instances; a leaf node holds a set of benchmark instances; and an inner node holds no benchmark instances but has a non-emtpy set of child nodes. The algorithm uses the following subroutines.

`is_leaf_node`(*node* n) returns true iff n is a leaf node.
`get_instances`(*node* n) returns the set of benchmark instances of leaf node n.
`number_of_subfamilies`(*node* n) returns the number of subfamilies of node n.
`get_subfamily`(*node* n, *natural* i) returns the node corresponding to the i-th subfamily of node n.
`empty_queue`() returns the empty queue.
`dequeue`(*queue* q) takes a non-empty queue q, dequeues its first element, and returns that element.
`enqueue`(*queue* q, *element* e) takes a queue q and an element e and enqueues e to q.
`enqueue_queue`(*queue* q_1, *queue* q_2) takes two queues q_1 and q_2 and returns the concatenation of q_1 and q_2.
`shuffle`(*queue* q) takes a queue of benchmark instances q and returns a random permutation of q.

```
1  Function benchmark_queue(node n): queue
2      if is_leaf_node (n) then
3          return shuffle (get_instances (n));
4      else
5          k ←number_of_subfamilies (n);
6          for i ←1 to k do
7              queues_subfamilies[i] ←benchmark_queue (get_subfamily (n,i));
8          end
9          queue ←empty_queue ();
10         repeat
11             tmp ←empty_queue ();
12             for i ←1 to k do
13                 if queues_subfamilies[i] ≠ empty_queue () then
14                     enqueue (tmp, dequeue (queues_subfamilies[i]));
15                 end
16             end
17             enqueue_queue (queue, shuffle (tmp));
18         until tmp = empty_queue ();
19         return queue;
20     end
21 end
```

Fig. 10. Generating a queue of benchmark instances.