

# On Deciding MUS Membership with QBF

Mikoláš Janota<sup>1</sup> and Joao Marques-Silva<sup>1,2</sup>

<sup>1</sup> INESC-ID, Lisbon, Portugal

<sup>2</sup> University College Dublin, Ireland

**Abstract.** This paper tackles the problem of deciding whether a given clause belongs to some minimally unsatisfiable subset (MUS) of a formula, where the formula is in conjunctive normal form (CNF) and unsatisfiable. Deciding MUS-membership helps the understanding of *why* a formula is unsatisfiable. If a clause does not belong to any MUS, then removing it will certainly not contribute to restoring the formula’s consistency. Unsatisfiable formulas and consistency restoration in particular have a number of practical applications in areas such as software verification or product configuration. The MUS-membership problem is known to be in the second level of polynomial hierarchy, more precisely it is  $\Sigma_2^P$ -complete. Hence, quantified Boolean formulas (QBFs) represent a possible avenue for tackling the problem. This paper develops a number of novel QBF formulations of the MUS-membership problem and evaluates their practicality using modern off-the-shelf solvers.

## 1 Introduction

Unsatisfiable formulas, representing refutation proofs or inconsistencies, appear in various areas of automated reasoning. This article focuses on helping us to understand *why* a certain formula is unsatisfiable. If a formula is represented in conjunctive normal form (CNF), it is sufficient to consider only certain subsets of clauses to see why it is unsatisfiable. In particular, a set of clauses is called a minimally unsatisfiable subset (MUS) if it is unsatisfiable and any of its proper subsets is satisfiable. The question addressed in this article is to determine whether a given clause belongs to some MUS of a formula. This is referred to as the MUS-MEMBERSHIP problem.

Deciding whether a clause belongs to some MUS is important when one wants to *restore consistency* of a formula: removing a clause that is *not* part of any MUS, will certainly not restore consistency. Restoring consistency is an active area of research in the area of *product configuration* [20,22]. For example, when configuring a product, some sets of its features result in an inconsistent configuration. Approaches for resolving conflicting features often involves user intervention, e.g. to decide which features to deselect. Clearly, it is useful for the user to know if a feature is relevant for the inconsistency.

Earlier work on the MUS-MEMBERSHIP problem consisted on complexity characterizations [13,12] and an algorithm based on heuristically-guided MUS enumeration [7]. In contrast, this article proposes four alternative solutions for solving MUS-MEMBERSHIP problem with Quantified Boolean Formulas (QBF).

Two of these solutions follow directly from the problem’s definition, and either involve a  $\text{QBF}_{3,\exists}$  formula or a  $\text{QBF}_{2,\exists}$  formula that grows quadratically with the size of the original formula. The paper also exploits the relationship between MUSes and Maximally Satisfiable Subsets (MSSes), and derives a  $\text{QBF}_{2,\exists}$  model for the MUS-MEMBERSHIP problem that grows linearly with the size of the original formula. Furthermore, this relationship is also used for relating the MUS-MEMBERSHIP with the problem of *inference in propositional circumscription*, which can be represented as a  $\text{QBF}_{2,\exists}$  with a specific structure. In turn, this enables the use of specialized algorithms for propositional circumscription [8]. Experimental results obtained on representative classes of problem instances demonstrate that the recent abstraction refinement algorithm [8] consistently outperforms all other approaches.

The paper is organized as follows. Section 2 introduces the notation and definitions used throughout the paper. Section 3 develops different models for solving the MUS-MEMBERSHIP problem. Section 4 analyzes results obtained on representative classes of problem instances. Finally, Section 5 concludes the paper.

## 2 Preliminaries

Throughout this paper,  $\phi$  and  $\psi$  denote Boolean formulas, defined on a set of variables  $X = \{x_1, \dots, x_n\}$ . Where necessary, additional sets of variables are considered, e.g.  $R, X'$ . A Boolean formula  $\phi$  in Conjunctive Normal Form (CNF) is a conjunction of disjunctions of literals and a literal is a variable or its negation. A disjunction of literals is called a *clause* and it is preferably represented by  $\omega$ . Unless specified otherwise,  $\phi$  is assumed to be of the form  $\{\omega_1, \dots, \omega_n\}$ . Where appropriate, a CNF formula is interpreted as a set of sets of literals.

A *truth assignment*  $\mu_X$  is a mapping from a set of variables  $X$  to  $\{0, 1\}$ ,  $\mu_X : X \rightarrow \{0, 1\}$ . A truth assignment is represented by the set  $M_X$  of true variables in  $\mu_X$ ,  $M_X = \{x_i \in X \mid \mu_X(x_i) = 1\}$ . In what follows, truth assignments will be represented by the set of true variables, since the definition of  $\mu_X$  is implicit, given  $X$  and  $M_X$ . Moreover,  $M_X \models \phi$  is used to denote that truth assignment  $\mu_X$  is a *model* of  $\phi$ , i.e. that  $\mu_X$  satisfies all clauses in  $\phi$ . Truth assignments will also be defined for other sets of variables, as needed, e.g.  $M_S, M_{S_a}, M_{S_b}$ . When a formula is defined over distinct sets of variables, e.g.  $X$  and  $S$ ,  $M_S, M_X \models \phi$  denotes that the truth assignment to the variables in  $S$  and the variables in  $X$  satisfies  $\phi$ . Finally, a truth assignment represented by  $M_X$  implicitly denotes that  $M_X \subseteq X$ . Similarly,  $M_R$  implicitly denotes that  $M_R \subseteq R$ . To simplify the notation, the set containment relation will be omitted in all formulas.

A *QBF* is a Boolean formula where each variable is either universally or existentially quantified. We write  $\text{QBF}_{k,\exists}$  to denote the class of formulas of the form  $\mathcal{Q}_1 X_1 \dots \mathcal{Q}_k X_k \cdot \phi$  where  $\mathcal{Q}_i = \exists$  if  $i$  is odd and  $\mathcal{Q}_i = \forall$  if  $i$  is even. In the context of QBF we write  $\phi(X)$  to denote a formula that is built on the variables from  $X$ . An important result from the complexity theory is that the validity of a formula in  $\text{QBF}_{k,\exists}$  is  $\Sigma_k^P$ -complete [18].

## 2.1 Minimal Unsatisfiability and Maximal Satisfiability

This section introduces the concepts of minimally unsatisfiable and maximally satisfiable sets of clauses as well as related decision problems.

**Definition 1 (MUS).** *A set of clauses  $\psi \subseteq \phi$  is a Minimally Unsatisfiable Subset (MUS) iff  $\psi$  is unsatisfiable and any set  $\psi' \subsetneq \psi$  is satisfiable.*

**Definition 2 (MSS).** *A set of clauses  $\psi \subseteq \phi$  is a Maximally Satisfiable Subset (MSS) iff  $\psi$  is satisfiable and any set  $\psi' \subseteq \phi$  such that  $\psi \subsetneq \psi'$  is unsatisfiable.*

**Definition 3 (MCS).** *A set of clauses  $\psi \subseteq \phi$  is a Minimally Correction Subset (MCS) if  $\phi \setminus \psi$  is satisfiable and for any subset  $\psi' \subsetneq \psi$ ,  $\phi \setminus \psi'$  is unsatisfiable.*

Deciding whether a CNF formula is an MUS is  $D^P$ -complete [21]. Algorithms for computing MUSes have been the subject of comprehensive research over the years [6,2,16]. Moreover, this article considers the following decision problems.

**Name:** MUS-MEMBERSHIP

**Given:** A CNF formula  $\phi$  and a clause  $\omega \in \phi$ .

**Question:** Is there an MUS  $\psi$  of  $\phi$  such that  $\omega \in \psi$ ?

**Name:** MUS-OVERLAP

**Given:** CNF formulas  $\phi$  and  $\gamma \subseteq \phi$ .

**Question:** Is there an MUS  $\psi$  of  $\phi$  such that  $\gamma \cap \psi \neq \emptyset$ ?

**Name:** MSS-MEMBERSHIP

**Given:** A CNF formula  $\phi$  and a clause  $\omega \in \phi$ .

**Question:** Is there an MSS  $\psi$  of  $\phi$  such that  $\omega \notin \psi$ ?

We make several observations regarding the definitions. MUS-OVERLAP can be expressed as a disjunction of  $k$  instances of MUS-MEMBERSHIP, where  $k$  is the number of clauses in the formula  $\gamma$ . Hence, we mainly focus on MUS-MEMBERSHIP. We require  $\omega \in \phi$ , which is done convenience, and the decision problems for  $\omega \notin \phi$  are trivial. In MUS-MEMBERSHIP we are looking for an MUS containing  $\omega$  whereas in MSS-MEMBERSHIP, we are looking for an MSS that does *not* contain  $\omega$ . Later on we show that the problems are convertible to one another.

To obtain the complexity classification of MUS-MEMBERSHIP, we realize that an MUS is a special case of irredundancy: an MUS is a subset-minimal representation that is equivalent to the original formula. The question whether a clause belongs to some minimal irredundant representation is known to be  $\Sigma_2^P$ -complete [13]. Hence, MUS-MEMBERSHIP is in  $\Sigma_2^P$ . In fact, it has been shown that MUS-MEMBERSHIP itself is  $\Sigma_2^P$ -hard (and therefore complete) [12].

## 2.2 Propositional Circumscription

Circumscription was introduced by McCarthy as a form of nonmonotonic reasoning [17]. While the original definition of circumscription is for first-order logic, for the purpose of this article we consider its propositional version.

**Definition 4 (Circumscription).** *Let  $P, Z$  be sets of variables and  $\phi$  a formula on the variables  $P \cup Z$ . Circumscription of  $\phi$  is defined as follows:*

$$CIRC(\phi; P; Z) = \phi(P, Z) \wedge (\forall P', Z')((\phi(P', Z') \wedge (P' \rightarrow P)) \rightarrow (P \rightarrow P')) \quad (1)$$

where  $P' \rightarrow P$  stands for  $\bigwedge_{x \in P}(x' \rightarrow x)$ .

We should note that circumscription often considers another set of variables  $Q$ , which comprises variables that remain fixed. However, this set is not needed for the purpose of this article. Circumscription is closely related to model minimization introduced by the following two definitions.

**Definition 5 (Model Orderings).** *Let  $M$  and  $N$  be models of  $\phi$  and let  $P$  be a set of variables. We write  $M \preceq_P N$  iff  $M \cap P \subseteq N \cap P$  and we write  $M \prec_P N$  iff  $M \preceq_P N$  and  $M \neq N$ .*

**Definition 6 (Minimal Models).** *A model  $M$  of  $\phi$  is  $P$ -minimal iff there is no model  $N$  of  $\phi$  such that  $N \prec_P M$ . We write  $MM(\phi, P)$  to denote the set of all  $P$ -minimal models of  $\phi$ . For formulas  $\phi$  and  $\psi$  we write  $\phi \models_P^{circ} \psi$  iff  $\psi$  holds in all  $P$ -minimal models of  $\phi$ .*

In short, the ordering  $\preceq$  is a bit-wise ordering on the variables from  $P$  and minimal models are the minimal elements of this ordering. The relation between circumscription and minimal models is well-known [19,15,1,3]. The following well-known result is used throughout the paper [1].

**Proposition 1.** *Let  $\phi$  and  $\psi$  be formulas using only variables from  $P \cup Z$ . It holds that  $CIRC(\phi; P; Z) \models \psi$  iff  $\phi \models_P^{circ} \psi$ .*

Proposition 1 tells us that inference from the set of minimal models is equivalent to inference from circumscription. Another observation we make is that the entailment in propositional circumscription is immediately expressible as a 2-level QBF.

**Observation 1** *For formulas  $\phi$  and  $\psi$ ,  $CIRC(\phi; P; Z) \not\models \psi$  iff*

$$\exists P, Z. \neg \psi(P, Z) \wedge \phi(P, Z) \wedge (\forall P', Z')((\phi(P', Z') \wedge (P' \rightarrow P)) \rightarrow (P \rightarrow P')) \quad (2)$$

Note that the QBF above expresses that  $P, Z$  should satisfy (1) and violate  $\psi$ , and thus serve as a counterexample to the entailment. Naturally, the entailment can be expressed positively by negating the QBF.

Hence, propositional circumscription can be seen as reasoning over minimal models or as a special case of a QBF; in the remainder of the paper we treat these properties of propositional circumscription interchangeably.

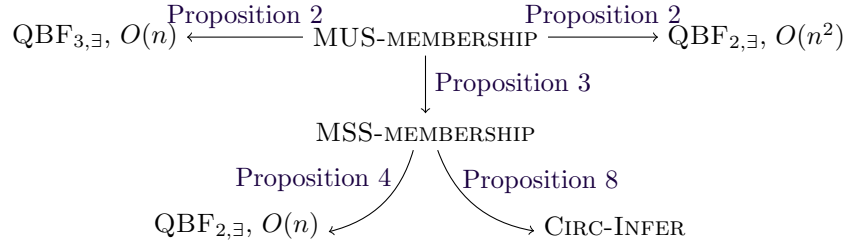


Fig. 1. Translating between problems

**Name:** CIRC-INFER

**Given:** CNF formulas  $\phi$  and  $\psi$  and sets of variables  $P$  and  $Q$ .

**Question:** Does  $\psi$  hold in all  $\langle P, Q \rangle$ -minimal models of  $\phi$ , i.e.  $\phi \models_{\langle P, Q \rangle}^{circ} \psi$ ?

### 2.3 Related Work

The MUS-MEMBERSHIP problem has been studied mostly from a theoretical perspective [13,12]. Motivated by practical applications, recent work addressed the development of algorithms for this problem [7]. These algorithms are based on explicit and implicit enumeration of MUSes. A simple algorithm for solving the MUS-MEMBERSHIP problem is to run an MUS enumerator (e.g. [14]) and check whether any MUS contains the target clause  $\omega \in \phi$ . This algorithm was improved in [7] where heuristics are proposed for reducing the sets of clauses to consider. The tool `cmMUS` [10] represents recent work on the MUS membership problem. `cmMUS` builds upon the work described in the present paper, namely the connection between MUS membership and propositional circumscription, which is detailed in Section 3.3.

## 3 Deciding MUS-MEMBERSHIP

Figure 1 depicts the relations between the problems investigated in the remainder of this section. The motivation for these translations is to derive QBF formulas for the MUS-MEMBERSHIP problem. We show that a direct translation of MUS-MEMBERSHIP leads to QBF<sub>3,∃</sub> despite the problem being in  $\Sigma_2^P$ . Alternatively, we propose a QBF<sub>2,∃</sub> model, that is quadratic in the size of the original problem. Alternative QBF formulations are developed by exploiting the relationship between MUSes and MSSes. As a result, we derive a QBF<sub>2,∃</sub> that is linear in the size of the original problem, and also relate MSS-MEMBERSHIP with CIRC-INFER. The QBF models can be solved with standard QBF algorithms, whereas for the CIRC-INFER, a dedicated algorithm can be used [8].

### 3.1 MUS-MEMBERSHIP with QBF

This section investigates the translation of MUS-MEMBERSHIP to QBF. Since MUS-MEMBERSHIP has been shown to be  $\Sigma_2^P$ -complete [13], the problem must be expressible as a QBF $_{2,\exists}$  formula. We begin by a straightforward translation from the problem statement following the following schema:

**exists**  $\psi \subseteq \phi$  s.t.  $\omega \in \psi$  **and**  $\psi$  is unsatisfiable **and forall**  $\psi' \subsetneq \psi$  is satisfiable

To be able to quantify over subsets of  $\phi$ , we introduce its relaxed form.

**Definition 7 (relaxation  $\phi^*$ ).** Let  $\phi$  be a set of clauses then its relaxation  $\phi^*$  is defined as follows:

$$\phi^* = \{\omega \vee r_\omega \mid \omega \in \phi\} \quad (3)$$

where  $r_\omega$  are variables not appearing in  $\phi$ . We refer to  $r_\omega$  as the relaxation variable of the clause  $\omega$  and if  $r_\omega$  has the value 1, we say that the clause  $\omega$  is relaxed.

In the following text we use  $R$  to denote the set of relaxation variables (and  $X$  for the set of original variables as before). The intuition behind relaxation variables is that once a clause is relaxed, it is equivalent to not having the clause in the formula. For succinctness, we introduce a dual term of *selected clauses*, which are clauses that are *not* relaxed.

**Definition 8.** Let  $\phi^*$  be a relaxation of  $\phi$  and  $M_R$  be a subset of the pertaining relaxation variables. The set of selected clauses  $\mathcal{S}(\phi^*, M_R)$  is defined as follows:

$$\mathcal{S}(\phi^*, M_R) = \{\omega \mid r_\omega \notin M_R\} \quad (4)$$

*Example 1.* Let  $\phi = \{x, \neg x, y\}$ , then  $\phi^* = \{x \vee r_1, \neg x \vee r_2, y \vee r_3\}$ . Let  $M_R = \{r_1, r_2\}$  then  $\mathcal{S}(\phi^*, M_R) = \{y\}$ . Observe that for any  $M_X$  s.t.  $y \in M_X$  the interpretation  $M_R \cup M_X$  is a model of  $\phi^*$ : when the clauses  $x$  and  $\neg x$  are relaxed, they do not need to be satisfied. However, if a clause is not relaxed (the corresponding relaxation variable is 0), the clause must be satisfied. Hence for a given  $M_R$ , satisfying  $\phi^*$  is equivalent to satisfying  $\mathcal{S}(\phi^*, M_R)$ .

The following observation establishes a relation between the set of selected clauses and the relaxed formula.

**Observation 2** An assignment  $M_R \cup M_X$  is a model of  $\phi^*$  iff  $M_X$  is a model of  $\mathcal{S}(\phi^*, M_R)$ .

In the following QBF the relaxed formula appears in two versions: a non-primed version ( $\phi^*(R, X)$ ), and, a primed version—where all the variables are replaced with their primed copy ( $\phi^*(R', X')$ ).

Observe that relaxing a clause results into removing it from the set of selected clauses, and therefore for any relaxations  $M_R$  and  $M'_R$ , the requirement  $\mathcal{S}(\phi^*, M_R) \subseteq \mathcal{S}(\phi^*, M'_R)$  is equivalent to  $M'_R \subseteq M_R$ . In the following QBFs, the requirement  $M'_R \subseteq M_R$  is captured by the formula  $R < R'$  defined as follows:

$$R < R' \equiv \bigwedge_{z \in R} z \rightarrow z' \wedge \bigvee_{z \in R} \neg z \wedge z' \quad (5)$$

Now let us express the MUS-MEMBERSHIP as a QBF formula:

$$\exists R. \neg r_\omega \wedge (\forall X. \neg \phi^*(R, X)) \wedge (\forall R'. (R < R') \rightarrow \exists X'. \phi^*(R', X')) \quad (6)$$

The formula expresses that we are searching for a relaxation  $R$  for which the clause  $\omega$  is not relaxed ( $\neg r_\omega$ ). The set of selected clauses induced by the relaxation  $R$  is unsatisfiable ( $\forall X. \neg \phi^*(R, X)$ ). If  $R$  is relaxed anymore, then the induced set of selected clauses is satisfiable ( $\forall R'. (R < R') \rightarrow \exists X'. \phi^*(R', X')$ ).

Formula (6) can be reformulated if we realize that a set of clauses is an MUS iff removing any clause yields a satisfiable set of clauses:

$$\exists R. \neg r_\omega \wedge (\forall X. \neg \phi^*(R, X)) \wedge \bigwedge_{r_{\omega_i} \in R} (\neg r_{\omega_i} \rightarrow \exists X^{\omega_i}. \phi^*[r_{\omega_i}/1](R, X^{\omega_i})) \quad (7)$$

Where  $\phi^*[r_{\omega_i}/1]$  is the substitution of 1 for  $r_{\omega_i}$  in  $\phi^*$  and  $X^{\omega_i}$  is a fresh copy of the variables  $X$  for each  $r_{\omega_i} \in R$ .

Since the variables  $X$  appear only in the first half of the formula, it can be rewritten into 2QBF as follows:

$$\exists R \exists X^{\omega_1} \dots \exists X^{\omega_n} \forall X. \neg r_\omega \wedge \neg \phi^*(R, X) \wedge \bigwedge_{r_{\omega_i} \in R} (\neg r_{\omega_i} \rightarrow \phi^*[r_{\omega_i}/1](R, X^{\omega_i})) \quad (8)$$

Altogether, a solution  $M_R$  to either of the formulas (6), (7), or (8) represents an MUS containing the clause  $\omega$ , which enables us to state the following proposition.

**Proposition 2.** *The clause  $\omega$  belongs to some MUS of the formula  $\phi$  iff (6), (7), or (8) is valid.*

### 3.2 QBF for MUS-MEMBERSHIP using MSS-MEMBERSHIP

We observe that the equations developed above are problematic from a practical perspective. Equation (6) uses 3 levels of quantifiers despite the problem being in  $\Sigma_2^P$  [12]. Equation (8) has only 2 levels of quantifiers but uses a quadratic number of variables.

The following describes how to construct a QBF with 2 quantifiers using a linear number of variables by first translating the problem to MSS-MEMBERSHIP. In order to get to show the relation between MUS-MEMBERSHIP and MSS-MEMBERSHIP, we invoke the following lemma [11]:

**Lemma 1 (Lemma 4.3 in [11]).** *Let  $\text{MU}(\phi)$  denote the set of all MUSes of  $\phi$  and let  $\text{MS}(\phi)$  denote the set of all MSSes of  $\phi$ . Then the following equality holds:*

$$\bigcup \text{MU}(\phi) = \phi \setminus \bigcap \text{MS}(\phi) \quad (9)$$

An immediate consequence of Lemma 1 is that a clause  $\omega$  is included in some MUS of  $\phi$  if and only if  $\omega$  is *not* included in some MSS of  $\phi$ . This consequence is stated in the following proposition.

**Proposition 3.** *A clause  $\omega$  belongs to some MUS of  $\phi$  iff there exists an MSS  $\psi$  of  $\phi$  such that  $\omega \notin \psi$ .*

*Example 2.* Let  $\phi = \{\neg x, x, z\}$ . The formula  $\phi$  has only one MUS  $\{\neg x, x\}$  while it has two MSSes  $\{x, z\}$  and  $\{\neg x, z\}$ . Observe that the clause  $z$  is in both MSSes and not in the MUS; for both of the clauses  $x, \neg x$  there is an MSS without the clause and both are in the MUS.

The relation between MUSes and MSSes established by Proposition 3 motivates the following quantified Boolean formula for MUS-MEMBERSHIP (again we use the notation  $R' < R$  introduced earlier).

$$\exists R \exists X \forall R' \forall X'. (r_\omega \wedge \phi^*(R, X) \wedge (R' < R \rightarrow \neg \phi^*(R', X'))) \quad (10)$$

The formula expresses that we are looking for a relaxation in which  $\omega$  is relaxed ( $r_\omega$ ). The relaxation is satisfiable ( $\phi^*(R, X)$ ) and any relaxation relaxing less clauses yields an unsatisfiable set of clauses ( $R' < R \rightarrow \neg \phi^*(R', X')$ ). Altogether, a solution  $M_R$  to the equation (10) corresponds to an MSS that does *not* contain the clause  $\omega$ .

**Proposition 4.** *The answer to MUS-MEMBERSHIP is “yes” iff (10) is valid.*

Observe that the quantified formula has two levels of quantifiers and linear number of variables.

Equation (10) provides a solution for testing whether a clause  $\omega$  is included in an MUS of  $\phi$ . However, it does not provide a *witness*, i.e. an MUS containing  $\omega$ . Nevertheless, a witness can be computed by exploiting the properties of MSSes and MUSes.

**Lemma 2.** *Let  $\psi$  be an MSS of  $\phi$  such that  $\omega \notin \psi$ , then any MUS of  $\psi \cup \{\omega\}$  contains  $\omega$ .*

*Proof (sketch).* Since  $\psi$  is an MSS, adding *any* clause from  $\phi$  to  $\psi$  will make the result unsatisfiable. Therefore, adding  $\omega$  to  $\psi$ ,  $\psi' = \psi \cup \{\omega\}$ , results in an unsatisfiable formula. Let  $\psi'' \subseteq \psi'$  be an MUS of  $\psi'$ , then  $\omega \in \psi''$  as otherwise  $\psi'' \subseteq \psi$  would lead to a contradiction because  $\psi$  is satisfiable.

Lemma 2 enables the use of standard MUS extraction algorithms to extract an MUS witness given an MSS not containing  $\omega$ .

**Proposition 5.** *Let  $\phi$  be a CNF formula and  $\omega \in \phi$ . For a clause  $\omega$ , if the answer to the MUS-MEMBERSHIP problem is “yes”, then a witness for the MUS-MEMBERSHIP problem is any MUS of an MSS not containing  $\omega$ .*

*Proof.* Immediate consequence of Proposition 4 and Lemma 2.



### 3.3 MUS-MEMBERSHIP with CIRC-INFER

Proposition 3 lets us translate MUS-MEMBERSHIP to MSS-MEMBERSHIP. In this section we show how to translate MSS-MEMBERSHIP to CIRC-INFER.

Recall that CIRC-INFER is the problem of deciding whether a formula  $\psi$  holds in all  $P$ -minimal models of a formula  $\phi$ , for some set of variables  $P$  (Proposition 1). We begin by showing a relation between minimal models and MSSes.

As in the previous section, we operate on the relaxed formula  $\phi^*$  where setting the relaxation variable  $r_\omega$  to 1 effectively eliminates the clause (relaxes the clause). Dually, setting the variable  $r_\omega$  to 0 results into adding the clause  $\omega$  into the set of selected clauses (Definition 8). Consequently, MSSes correspond to minimal models of the relaxed formula, which is captured by the following proposition.

**Proposition 6.** *For an interpretation  $M_R$ , the set of selected clauses  $\mathcal{S}(\phi^*, M_R)$  is an MSS of  $\phi$  iff there exists  $M_X$  such that  $M_R \cup M_X$  is an  $R$ -minimal model of  $\phi^*$ .*

*Proof (sketch).* If the set  $\mathcal{S}(\phi^*, M_R)$  is an MSS, then it must be satisfiable and therefore it must have some model  $M_X$ . Due to Observation 2,  $M_R \cup M_X$  is a model of  $\phi^*$ . The model  $M_R \cup M_X$  must be  $R$ -minimal because otherwise we would obtain a relaxation corresponding to a strict superset of  $\mathcal{S}(\phi^*, M_R)$  rendering it not maximal. If  $M_R \cup M_X$  is an  $R$ -minimal model of  $\phi^*$  then  $\mathcal{S}(\phi^*, M_R)$  is satisfied by  $M_X$  due to Observation 2. The set  $\mathcal{S}(\phi^*, M_R)$  must be an MSS otherwise  $M_R \cup M_X$  would not be  $R$ -minimal.

*Example 3.* Let  $\phi = \{x, \neg x, y \vee z, \neg y \vee \neg z\}$ , then  $\phi^* = \{x \vee r_1, \neg x \vee r_2, y \vee z \vee r_3, \neg y \vee \neg z \vee r_4\}$  for the relaxation variables  $R = \{r_1, r_2, r_3\}$ . In order to achieve consistency, one of the clauses  $x$  and  $\neg x$  must be relaxed. Hence, the formula  $\phi^*$  has the following four  $R$ -minimal models: two models have the clause  $\neg x$  relaxed  $\{r_2, x, y\}, \{r_2, x, z\}$  and two models have the clause  $x$  relaxed  $\{r_1, y\}, \{r_1, z\}$ . These models correspond to the MSSes  $\{x, y \vee z, \neg y \vee \neg z\}$  and  $\{\neg x, y \vee z, \neg y \vee \neg z\}$ . Observe that the clauses  $y \vee z$  and  $\neg y \vee \neg z$  are in both MSSes, which means that the corresponding variables  $r_2$  and  $r_3$  have the value 0 in all  $R$ -minimal models (they never need to be relaxed).

Proposition 6 establishes a relation between the MSSes of a formula and minimal models of the corresponding relaxed formula. Consequently, in order to solve MUS-MEMBERSHIP for a clause  $\omega$ , we need to look for a minimal model with the clause relaxed.

**Proposition 7.** *A clause  $\omega$  belongs to some MUS of  $\phi$  iff there exists a model  $M \in MM(\phi, R)$  such that  $M \models r_\omega$ , equivalently:*

$$\phi^* \not\models_R^{circ} \neg r_\omega \quad (11)$$

*Proof (sketch).* A clause  $\omega$  belongs to some MUS of  $\phi$  iff there exists an MSS  $\psi \subseteq \phi$  s.t.  $\omega \notin \psi$  (Proposition 3). There exists an MSS  $\psi$  of  $\phi$  s.t.  $\omega \notin \psi$  iff there exists an  $R$ -minimal model  $M$  of  $\phi^*$  s.t.  $M \models r_\omega$  (Proposition 6). To relate to

the circumscription inference notation we observe that there exists an  $R$ -minimal model  $M$  s.t.  $M \models r_\omega$  iff  $\phi^* \not\models_R^{circ} \neg r_\omega$  does *not* hold.

Proposition 7 is easily generalized for MUS-OVERLAP by observing that we only need to find a minimal model where at least one of the clauses in question is relaxed.

**Proposition 8.** *A set of clauses  $\omega_1, \dots, \omega_n$  overlaps with some MUS of  $\phi$  iff there exists  $M \in MM(\phi, R)$  such that  $M \models r_{\omega_1} \vee \dots \vee r_{\omega_n}$ , equivalently:*

$$\phi^* \not\models_R^{circ} (\neg r_{\omega_1} \wedge \dots \wedge \neg r_{\omega_n}) \quad (12)$$

Observe that in Propositions 6 and 8 CIRC-INFER appears in a negative sense. This again agrees with the known complexity classification, as MUS-MEMBERSHIP is  $\Sigma_2^P$ -complete [12] and circumscription is  $\Pi_2^P$ -complete [3]. Moreover, although the relationship with QBF is simple (see Section 2.2), we opt to solve CIRC-INFER with a dedicated algorithm [8].

### 3.4 Algorithms for MUS-MEMBERSHIP

The previous two sections develop a number of properties of the MUS-MEMBERSHIP problem. This section summarizes the concrete algorithms that these properties enable us to consider. The algorithms are classified into three classes: enumeration, QBF, and circumscription inference. These classes are discussed in turn.

The simplest approach for deciding MUS-MEMBERSHIP is to enumerate MUSes (e.g. [14]). Practical algorithms for MUS-MEMBERSHIP follow this approach [7], but are coupled with heuristics for reducing the number of MUSes to enumerate. An alternative solution, also based on enumeration, consists of enumerating MSSes, using Proposition 3. Given that MUS enumeration algorithms start by enumerating MSSes [14], an algorithm based on enumerating MSSes is guaranteed to outperform naïve solutions based on MUS enumeration. It should be noted that existing algorithms (e.g. [14]) are based on MCS enumeration. However, since an MCS is the complement of an MSS, a clause is in an MUS iff it is included in some MCS.

A second class of algorithms consists of using the mapping to QBF and solving the resulting problem instances with a QBF solver. This paper develops 3 alternative approaches for encoding MUS-MEMBERSHIP into QBF (see Figure 1). The first approach uses 3 levels of quantifiers and produces a formula linear in the size; second approach uses 2 levels of quantifiers but produces a formula of quadratic size; the third approach uses the relation between MUSes and MSSes and provides a 2-level formulation of linear size.

Finally, a third class of algorithms exploits the relationship between MSSes and CIRC-INFER. As noted in preliminaries, CIRC-INFER is a special case of a 2-level QBF formula. Hence, a general QBF solver could be used. However, this formulation also enables the use of specialized algorithms for CIRC-INFER [8].

## 4 Experimental Results

Following the discussion in Section 3.4, the following concrete tools were used for the evaluation.

**MUSer** is a tool for extracting MUSes [16]. The tool was used to obtain a witness MUS in approaches based on MSSes (see Proposition 5).

**AReQS** is a recently developed solver, implemented by the authors, for 2QBF formulas based on counterexample guided refinement (CEGAR). The solver was used for all QBF formulas with 2 levels of quantifiers since previous research showed that it consistently outperforms the available solvers on these types of formulas [9].

**cmMUS** is a tool that solves MSS-MEMBERSHIP using propositional circumscription (implemented by the authors) [10]. Just as **AReQS**, it uses counterexample guided abstraction refinement approach but tailored for propositional circumscription [8].

**look4MUS** is a tool dedicated to MUS-MEMBERSHIP based on MUS enumeration, guided by heuristics based on a measure of inconsistency [7].

**QuBE 7.1** is a QBF solver<sup>3</sup> which solved the most instances in the 2CNF track of QBF Evaluation 2010<sup>4</sup> and overall ranks high in all categories. **QuBE** has a powerful built-in preprocessor, which significantly improves its performance [5] (all of the preprocessing techniques were switched on for the purpose of the evaluation). The solver was used to evaluate 3-level formulas (see (6)). The downside of **QuBE 7.1** is that it does not provide a model. Hence, even though a solution to (6) is immediately an MUS containing the desired clause, it cannot be retrieved from the answer of **QuBE 7.1**.

**sSolve** is a QBF solver which returns a solution for valid formulas, unlike **QuBE 7.1** [4]. The solver was used to evaluate 3-level formulas (see (6)).

*MSS enum.* The tool **CAMUS** [14] was used to enumerate MSSes of the given formula. If the enumeration is looking for an MSS that overlaps with a set of clauses  $\gamma$ , then it immediately stops once it finds an MSS  $\psi$  that does not contain at least one of the clauses from  $\gamma$ , i.e.  $\gamma \setminus \psi \neq \emptyset$ .

### 4.1 Benchmarks

A variety of unsatisfiable formulas was selected from SAT competitions benchmarks<sup>5</sup> and from well-known applications of SAT (namely ATPG and product configuration). The selected formulas are relatively easy for modern SAT solvers because MUS-MEMBERSHIP is significantly harder than satisfiability. Even so, instances with tens of thousands of clauses were used (e.g. dining philosophers).

For each of these formulas, the MUS-OVERLAP was computed using the various approaches. The 1st, 3rd, 5th, and 7th clauses in the formula’s DIMACS representation were chosen as the set  $\gamma$  for which the overlap was to be determined—this evaluation methodology was also used in [7].

<sup>3</sup> Available at [www.star.dist.unige.it/~qube/](http://www.star.dist.unige.it/~qube/).

<sup>4</sup> <http://www.qbflib.org/>

<sup>5</sup> <http://www.satcompetition.org/>

	cmMUS	look4MUS	MSS enum.	2lev. lin.
Nemesis (bf) (223)	<b>223</b>	<b>223</b>	31	29
Daimler-Chrysler (84)	46	13	<b>49</b>	36
dining phil. (22)	<b>17</b>	<b>17</b>	4	8
dimacs (87)	<b>87</b>	82	51	51
ezfact (41)	<b>20</b>	11	11	10
total (457)	<b>393</b>	346	146	134

	2lev. qv.	3lev. lin. (QuBE)	3lev. lin. (sSolve)
Nemesis (bf) (223)	9	13	0
Daimler-Chrysler (84)	0	4	0
dining phil. (22)	2	1	0
dimacs (87)	18	25	4
ezfact (41)	0	0	0
total (457)	29	43	4

**Table 1.** Number of solved instances by the different approaches

## 4.2 Results

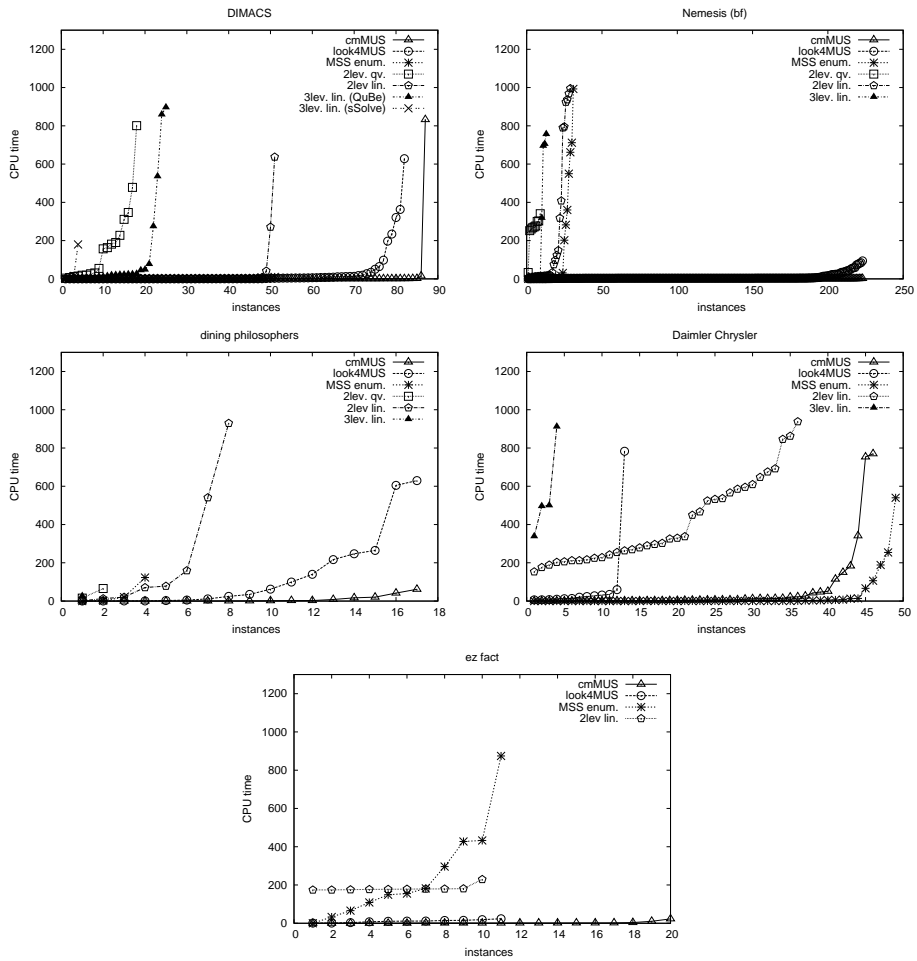
All experimental results were obtained on an Intel Xeon 5160 3GHz with 4GB of memory. The experiments were obtained with a memory limit of 2GB and time limit of 1,000 seconds. The results of the measurements are presented by Table 1 and Figure 2. Table 1 presents the number of solved instances by each of the approaches for each set of benchmarks. Figure 2 presents the computation times with *cactus plots*—the horizontal axis represents the number of instances that were solved within the time represented by the vertical axis.

The QBFs derived in Section 3.1 and Section 3.2 are denoted as: 2-level linear—2-level linear formula using MSS; 2-level quadratic—2-level quadratic formula directly yielding a witnessing MUS; 3-level linear—3-level linear formula directly yielding a witnessing MUS. The results for the approaches that first find an MSS include the runtime of `MUSer`, which was used to obtain the witnessing MUS (see Section 3.2).

Out of the presented approaches, the circumscription-based approach (`cmMUS`) turned out to be the most robust one: it has solved the most instances (393) and except for one class of benchmarks it exhibits the shortest overall running times. The set of benchmarks where `cmMUS` came second are the Daimler-Chrysler, for which the simple MSS enumeration solved 3 more instances.

The dedicated algorithm `look4MUS` came second in terms of the number of solved instances (346). However, it turned out not to be robust, e.g. a small number of instances were solved for Daimler-Chrysler and `ezfact`.

The use of general QBF solvers yielded significantly poorer results. As expected, out of these, the 2-level linear formulation (solved by `AReQS + MUSer`) had the best performance with 134 solved instances. Even though both `AReQS` and `cmMUS` use CEGAR to solve the problem, `cmMUS` uses a refinement specific



**Fig. 2.** Cactus plots for the measurements (number of instances  $x$  solved in less than  $y$  seconds)

to circumscription (cf. [8,9]) and that turned out to be important for the performance. The 3-level linear approach using QuBE 7.1 solved significantly more than both `sSolve` with 3-level approach and `AReQS` with quadratic 2-level formulation. However, we recall that QuBE 7.1 does not provide a model. In most cases when the quadratic formulation approach did not succeed it was because of exceeding the memory limit.

We should note that the runtime of `MUSer` affected very little the overall runtimes of the approaches based on MSSes. Mostly, the runtime of `MUSer` was below 1 second. Only two instances where the desired MUS was not found in time appeared (in dining philosophers and ezfact).

## 5 Conclusions

This article addresses the problem of deciding whether a given clause belongs to some minimal unsatisfiable subset (MUS) of some CNF formula. This is a well-known  $\Sigma_2^P$ -complete problem [13,12], for which recent work proposed heuristic-guided algorithms based on enumeration of MUSes [7]. In contrast, this paper develops new solutions for the MUS-MEMBERSHIP problem based on QBF. Some of the QBFs follow from the problem’s definition, whereas the others exploit the relationship between MUSes and MSSes [11]. The proposed solutions include one  $\text{QBF}_{3,\exists}$  and two  $\text{QBF}_{2,\exists}$  formulations. One additional solution consists of mapping MUS-MEMBERSHIP to `CIRC-INFER`, the propositional circumscription inference problem, itself expressible as a  $\text{QBF}_{2,\exists}$ . Given well-known mappings of propositional circumscription to other formalisms, this yields additional algorithms to solve the MUS-MEMBERSHIP problem. Experimental results obtained on a wide range of well-known benchmarks, demonstrate that the most effective approach consists of using a recent counterexample guided abstraction refinement algorithm for the propositional circumscription inference problem [8].

The promising experimental results suggest considering the use of dedicated algorithms for propositional circumscription inference in other settings, namely other  $\Sigma_2^P$ -complete and  $\Pi_2^P$ -complete decision problems.

**Acknowledgement.** This work is partially supported by SFI PI grant BEACON (09/IN.1/12618), EC FP7 project MANCOOSI (214898), FCT grants ATTEST (CMU-PT/ELE/0009/2009), and INESC-ID multiannual funding from the PIDDAC program funds.

## References

1. Cadoli, M., Lenzerini, M.: The complexity of closed world reasoning and circumscription. In: AAAI Conference on Artificial Intelligence. pp. 550–555 (1990)
2. Desrosiers, C., Galinier, P., Hertz, A., Paroz, S.: Using heuristics to find minimal unsatisfiable subformulas in satisfiability problems. *J. Comb. Optim.* 18(2), 124–150 (2009)
3. Eiter, T., Gottlob, G.: Propositional circumscription and extended closed-world reasoning are  $\Pi_2^P$ -complete. *Theor. Comput. Sci.* 114(2), 231–245 (1993)

4. Feldmann, R., Monien, B., Schamberger, S.: A distributed algorithm to evaluate quantified Boolean formulae. In: AAAI/IAAI. pp. 285–290 (2000)
5. Giunchiglia, E., Marin, P., Narizzano, M.: An effective preprocessor for QBF pre-reasoning. In: 2nd International Workshop on Quantification in Constraint Programming (QiCP) (2008)
6. Grégoire, E., Mazure, B., Piette, C.: On approaches to explaining infeasibility of sets of Boolean clauses. In: International Conference on Tools with Artificial Intelligence. pp. 74–83 (November 2008)
7. Grégoire, E., Mazure, B., Piette, C.: Does this set of clauses overlap with at least one MUS? In: Proceedings of the 22nd International Conference on Automated Deduction. pp. 100–115. CADE-22, Springer-Verlag, Berlin, Heidelberg (2009)
8. Janota, M., Grigore, R., Marques-Silva, J.: Counterexample guided abstraction refinement algorithm for propositional circumscription. In: Proceeding of the 12th European Conference on Logics in Artificial Intelligence (JELIA) (2010)
9. Janota, M., Marques-Silva, J.: Abstraction-based algorithm for 2QBF. In: Sakallah and Simon [23]
10. Janota, M., Marques-Silva, J.: cmMUS: a circumscription-based tool for MUS membership testing. In: Int. Conf. on Logic Prog. and Nonmonotonic Reasoning (LPNMR). Springer (2011)
11. Kullmann, O.: An application of matroid theory to the SAT problem. In: IEEE Conference on Computational Complexity. pp. 116–124 (2000)
12. Kullmann, O.: Constraint satisfaction problems in clausal form: Autarkies and minimal unsatisfiability. ECCC 14(055) (2007)
13. Liberatore, P.: Redundancy in logic I: CNF propositional formulae. *Artif. Intell.* 163(2), 203–232 (2005)
14. Liffiton, M.H., Sakallah, K.A.: Algorithms for computing minimal unsatisfiable subsets of constraints. *J. Autom. Reasoning* 40(1), 1–33 (2008)
15. Lifschitz, V.: Some results on circumscription. In: NMR. pp. 151–164 (1984)
16. Marques-Silva, J., Lynce, I.: On improving MUS extraction algorithms. In: Sakallah and Simon [23]
17. McCarthy, J.: Circumscription - a form of non-monotonic reasoning. *Artif. Intell.* 13(1-2), 27–39 (1980)
18. Meyer, A.R., Stockmeyer, L.J.: The equivalence problem for regular expressions with squaring requires exponential space. In: *Switching and Automata Theory* (1972)
19. Minker, J.: On indefinite databases and the closed world assumption. In: *Conference on Automated Deduction*. pp. 292–308 (1982)
20. O’Callaghan, B., O’Sullivan, B., Freuder, E.C.: Generating corrective explanations for interactive constraint satisfaction. In: van Beek, P. (ed.) *CP. Lecture Notes in Computer Science*, vol. 3709, pp. 445–459. Springer (2005)
21. Papadimitriou, C.H., Wolfe, D.: The complexity of facets resolved. *J. Comput. Syst. Sci.* 37(1), 2–13 (1988)
22. Papadopoulos, A., O’Sullivan, B.: Relaxations for compiled over-constrained problems. In: Stuckey, P.J. (ed.) *CP*. pp. 433–447. Springer (2008)
23. Sakallah, K.A., Simon, L. (eds.): *The 14th International Conference on Theory and Applications of Satisfiability Testing (SAT)*. Springer (2011)