

Research Note

An Extension of Deficiency and Minimal Unsatisfiability of Quantified Boolean Formulas

Hans Kleine Büning

kbcs1@upb.de

*Department of Computer Science,
Universität Paderborn
33095 Paderborn, Germany*

Xishun Zhao*

hsszxs@mail.sysu.edu.cn

*Institute of Logic and Cognition,
Sun Yat-sen University
510275 Guangzhou, P.R. China*

Abstract

The notion of minimal unsatisfiability and minimal falsity will be extended to non-clausal and non-prenex quantified Boolean formulas. For quantified Boolean formulas in negation normal form we generalize the notion of deficiency to the so-called cohesion, which is $1 +$ the difference between the number of occurrences of the conjunction symbol \wedge and the number of existential and free variables. Further, we show that all the complexity results with respect to minimal unsatisfiability or minimal falsity known for formulas with fixed deficiency can be adapted to formulas with fixed cohesion. For example, (1) the minimal unsatisfiability of propositional formulas with fixed cohesion is still solvable in polynomial time and, (2) the minimal falsity of quantified Boolean formulas with cohesion 1 is solvable in polynomial time.

KEYWORDS: *deficiency, cohesion, QBF, minimal unsatisfiability, minimal falsity*

Submitted September 2007; revised October 2007; published October 2007

1. Introduction

For quantified Boolean formulas in conjunctive normal form (QCNF) the deficiency has been defined as the difference between the number of clauses and the number of existential and free variables [1, 4]. Based on the simple observation that for QCNF the number of clauses equals the number of occurrences of the binary conjunction symbol \wedge plus 1, we will introduce an extension of the deficiency called cohesion. For formulas in negation normal form the cohesion is defined as $1 + \text{and}(\Phi) - v_{\exists}(\Phi)$, where $\text{and}(\Phi)$ is the number of occurrences of the \wedge -symbol and $v_{\exists}(\Phi)$ is the number of existential and free variables of the formula Φ .¹ Obviously, for QCNF the cohesion is nothing but the deficiency. Moreover, we will show that standard techniques transforming a quantified Boolean formula into QCNF

* Research was partially supported by the NSFC projects under grant No. 60573011, 10410638 and a MOE project under grant No. 05JJD72040122.

1. Here we implicitly regard free variables as existential ones. This is suitable because we are interested in the satisfiability problem of QBF.

preserve the cohesion. All the results known for classes with fixed deficiency can be adapted straight forward to classes with fixed cohesion.

A propositional formula in conjunctive normal form (CNF) is *minimal unsatisfiable*, if the formula is unsatisfiable and removing an arbitrary clause results in a satisfiable formula. The class of these formulas is denoted as MU. It has been shown [6], that MU is D^P -complete. MU has its quantified counterpart in QCNF, the class of quantified Boolean formulas with matrix in CNF. In order to distinguish between quantified formulas and propositional formulas these formulas are called *minimal false*. MF is the class of unsatisfiable formulas in QCNF, which after the deletion of an arbitrary clause are satisfiable. In case of quantified Boolean formulas without free variables we also use the term unsatisfiable for false formulas. The PSPACE-completeness of MF is a simple consequence of the PSPACE-completeness of the satisfiability problem for QCNF [5].

In the paper in hand we assume that the quantified Boolean formulas are in negation normal form, that is, each occurrence of the negation symbol \neg proceeds a variable. We will extend the notion of minimal unsatisfiability, minimal falsity respectively, to these formulas.

For the deficiency it has been shown [1, 4], that for all minimal unsatisfiable, minimal false respectively, formulas the deficiency is greater than 0. For CNF, $MU(k)$ is the class of formulas in MU with fixed deficiency k . Analogously, the class $MF(k)$ is defined. $MU(k)$ and $MF(1)$ are solvable in polynomial time [2, 4], whereas for $k \geq 2$ the computational complexity of $MF(k)$ is open. All these results remain valid for the cohesion and the extension of MU and MF to non-clausal formulas.

2. Cohesion

In this section we will introduce the *cohesion* of quantified Boolean formulas. For sake of simplicity we demand that quantifiers always refer to distinct variables, and no variable occurs both free and bound. The length of a quantified Boolean formula is the number of occurrences of literals, where the occurrences of variables in a prefix are not included. More formally, the length can be defined as $l(L) = 1$ for a literal L , $l(\alpha \wedge \beta) = l(\alpha \vee \beta) = l(\alpha) + l(\beta)$, and $l(Qx\alpha) = l(\alpha)$ for $Q \in \{\exists, \forall\}$.

Let Φ be a formula with r universal variables, t existential and s free variables. Then $v(\Phi) = r + t + s$ is the number of variables and $v_{\exists}(\Phi) = t + s$ is the number of existential and free variables.

The number of occurrences of \wedge -symbols in a formula Φ is denoted as $and(\Phi)$, whereas $or(\Phi)$ is the number of \vee -symbols. Obviously, we have $l(\Phi) = 1 + and(\Phi) + or(\Phi)$. The deficiency of a formula $\Phi \in QCNF$ has been defined as $d(\Phi) = n - v_{\exists}(\Phi)$, where n is the number of clauses [4].

Definition 1 (*Cohesion*) *Let Φ be a quantified Boolean formula in negation normal form. The cohesion $D(\Phi)$ is defined as $D(\Phi) = 1 + and(\Phi) - v_{\exists}(\Phi)$.*

For example, the formula $\Phi = \forall x \exists y ((x \wedge y) \vee (\neg x \wedge (z \vee \neg y)))$ has the cohesion $D(\Phi) = 1 + and(\Phi) - v_{\exists}(\Phi) = 1 + 2 - 2 = 1$.

In order to transform a quantified Boolean formula Φ into a formula in QCNF, two standard procedures can be applied with Φ .

There is a procedure generating from a general quantified formula in negation normal form a logically equivalent formula in prenex form [3]. Please notice, that distinct occurrences of quantifiers have distinct variables. The procedure consists of iterated applications of the following rules: A subformula $(Qx\Theta) \star \Psi$ (resp. $\Theta \star (Qx\Psi)$) will be replaced by $Qx(\Theta \star \Psi)$ for $\star \in \{\wedge, \vee\}$ and $Q \in \{\forall, \exists\}$. For example the formula $(\forall x_1 \exists y_1 \varphi) \wedge (\exists y_2 \exists y_3 \psi)$ can be transformed equivalently into $\forall x_1 \exists y_1 \exists y_2 \exists y_3 (\varphi \wedge \psi)$. Please note, that different orders of the application of the above rules may transform a formula Φ into different prenex forms. An arbitrary but fixed formula in prenex form generated by the procedure is denoted as $\text{pr}(\Phi)$. Since any application of the rules changes neither the number of \wedge -symbols nor the number of free and existential variables, we obtain $D(\Phi) = D(\text{pr}(\Phi))$.

For propositional formulas in negation normal form the so-called Tseitin procedure generates satisfiability-equivalent formulas in CNF [7]. Two formulas φ and ψ are satisfiability equivalent, if φ is satisfiable if and only if ψ is satisfiable. Let φ be a propositional formula in negation normal form, the Tseitin procedure replaces step by step subformulas of the form $(\pi \wedge \beta) \vee \sigma$ by the formula $(z \vee \pi) \wedge (z \vee \beta) \wedge (\neg z \vee \sigma)$ for a new variable z . Let $\text{ts}(\varphi)$ denote an arbitrary but fixed formula in CNF obtained from φ by this Tseitin procedure. Clearly, $D(\text{ts}(\varphi)) = D(\varphi)$, because the Tseitin procedure adds as many \wedge -symbols as new variables.

For a quantified formula $\Phi = Q\varphi$ in prenex form, we first transform the matrix φ into a CNF formula $\text{ts}(\varphi)$ by applying the Tseitin procedure. Let z_1, \dots, z_n be the introduced new variables. We then add to the prefix the existential quantifiers $\exists z_1 \dots \exists z_n$. We obtain a formula $Q\exists z_1 \dots \exists z_n \text{ts}(\varphi)$, denoted as $\text{ts}(\Phi)$, which is in QCNF and equivalent to Φ . Again, we have $D(\text{ts}(\Phi)) = D(\Phi)$, because $\text{ts}(\Phi)$ adds as many \wedge -symbols as new existential variables.

Let $Q\varphi$ be a quantified Boolean formula with prefix Q and matrix φ in CNF. Then $\varphi_{|\exists}$ denotes the propositional formula we obtain from φ by eliminating all occurrences of universal literals. A universal clause is a clause consisting of universally quantified literals. That means, the clause contains no free or existentially quantified variable.

Lemma 1 *Let Φ be a quantified Boolean formula in negation normal form.*

1. $D(\text{pr}(\Phi)) = D(\text{ts}(\text{pr}(\Phi))) = D(\Phi)$
2. $D(\Phi) = l(\Phi) - \text{or}(\Phi) - v_{\exists}(\Phi)$
3. *If $\Phi \in \text{CNF}$, then $D(\Phi) = d(\Phi)$.*
4. *If $\Phi = Q\varphi \in \text{QCNF}$ with r universal clauses, then $D(\Phi) = r + D(\varphi_{|\exists})$.*
5. *If Φ is a propositional formula over the variables x_1, \dots, x_n , where x_i occurs r_i -times in Φ , then we have $D(\Phi) = \sum_{1 \leq i \leq n} (r_i - 1) - \text{or}(\Phi)$.*

Proof: *Ad 1, 2, 3:* The first three items follow from the previous remarks.

Ad 4: Let $\varphi = \varphi_1 \wedge \varphi_2$ be the matrix of Φ , where φ_1 is the conjunction of r universal clauses and φ_2 is the conjunction of clauses with at least one existential or free variable. Then we have $D(\Phi) = 1 + \text{and}(\Phi) - v_{\exists}(\Phi) = 1 + (r-1) + 1 + \text{and}(\varphi_2) - v_{\exists}(\Phi) = r + 1 + \text{and}(\varphi_{|\exists}) - v_{\exists}(\Phi) = r + D(\varphi_{|\exists})$.

Ad 5: For a formula Φ we have $l(\Phi) = \sum_{1 \leq i \leq n} r_i$ and therefore $l(\Phi) - v_{|\exists}(\Phi) = \sum_{1 \leq i \leq n} (r_i - 1)$. By means of (2) we obtain our desired result. ■

There are various additional prenexing techniques transforming a formula into a formula in prenex form. For example, the rule replacing $((\exists x\alpha) \vee (\exists y\beta))$ by $\exists x(\alpha \vee \beta[y/x])$ reduces the number of existential quantifiers. $\beta[y/x]$ is the formula we obtain by substituting y by x in β . For example, $\Phi = (\exists x(x \vee z_1)) \vee (\exists y(y \vee z_2))$ will be transformed into $\Phi' = \exists x((x \vee z_1) \vee (x \vee z_2))$ and for the cohesion holds $D(\Phi) = 1 - 4 = -3$ and $D(\Phi') = 1 - 3 = -2$. Because the number of existential variables has been reduced, the cohesion increases, where the formulas are equivalent.

An application of the distributive law $\alpha \wedge (\beta \vee \gamma) \approx (\alpha \wedge \beta) \vee (\alpha \wedge \gamma)$ may change the cohesion. For example, for $\varphi = z \wedge (x \vee y)$ we obtain the formula $\varphi' = (z \wedge x) \vee (z \wedge y)$. The cohesion of φ is -1 and the cohesion of φ' is 0 .

3. Minimal Unsatisfiability and Falsity

The minimal unsatisfiability and minimal falsity are defined for formulas in conjunctive normal form. The formulas are unsatisfiable and after eliminating an arbitrary clause the formulas are satisfiable. In non-clausal formulas instead of the deletion of a clause we will remove so called or-subformulas based on a representation of formulas as trees.

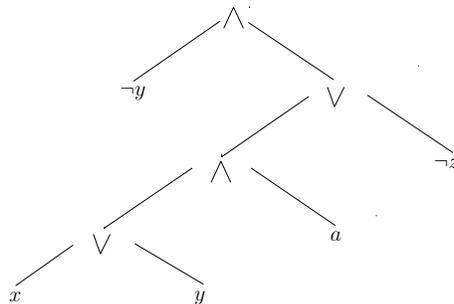
For a quantified Boolean formula in negation normal form we delete the quantifiers and represent the remaining propositional formula as a tree: If the formula is a literal L , then the associated tree T_L is a node labelled with L .

For a formula $\alpha \wedge \beta$ (resp. $\alpha \vee \beta$) let T_α and T_β be the associated trees. Then we have a root labelled with \wedge (resp. \vee) and the subtrees T_α and T_β . That means, the successor nodes of the \wedge -node (resp. \vee -node) are the roots of T_α and T_β . Please notice, that the leafs of the tree are labelled with the literals of the formula.

Let T be the associated tree of a formula. An *or-subtree* of T is a subtree T' , whose root is either a \vee -node or a literal and the root of the subtree T' is a successor of an \wedge -node.

For a quantified Boolean formulas Φ the associated tree T uniquely represents the propositional part of Φ . An or-subtree T' of T is again a representation of a propositional formula, say $\alpha_{T'}$, which is a subformula of the propositional part of Φ . The formula $\alpha_{T'}$ is called an or-subformula of Φ .

In case of formulas in conjunctive normal form the or-subformulas are exactly the clauses. For non-clausal formulas we give an example: For the formula $(\exists a \forall x \exists y (\neg y)) \wedge (((x \vee y) \wedge a) \vee \exists z (\neg z))$, the associated tree is



and the or-subformulas are $(\neg y)$, $((x \vee y) \vee a) \vee (\neg z)$, $(x \vee y)$, and a .

The extension of the definition of minimal unsatisfiability and minimal falsity is based on the deletion of or-subformulas. In case of a propositional formula φ an or-subformula of φ is a subformula of φ . The deletion of or-subformulas leads again to propositional formulas. For formulas with quantifiers the deletion of an or-subformula may lead to formulas in which for a quantifier, say $\exists x$, the variable x does not occur in the propositional part of the formula. In this case we delete the quantifier, too. For instance, the formula $\exists x(x \wedge \exists z \neg z)$ contains the or-subformulas x and $\neg z$. The deletion of z generates the formula $\exists x(x \wedge \exists z)$, which will be simplified to $\exists x(x)$, because z only occurs as quantifier variable.

For formulas in conjunctive normal form, that means a conjunction of clauses, a satisfiable formula remains satisfiable after the deletion of a clause. Similarly, after the deletion of an or-subtree in a satisfiable formula the formula is satisfiable. That follows from the definition of or-subtrees, because the predecessor node of an or-subtree is an \wedge -node and the formulas are in negation normal form. Suppose γ is an or-subformula of a satisfiable formula Φ . Then after the deletion of γ in Φ , a subformula $(\alpha \wedge \gamma)$ has been replaced by α , because the predecessor node of the or-subtree γ is an \wedge -node. Since all the negation symbols occur directly in front of the variables, any truth assignment making $(\alpha \wedge \gamma)$ true satisfies α , too. Therefore Φ remains satisfiable.

Definition 2 *A quantified Boolean formula Φ in negation normal form is in MF^* if and only if Φ is unsatisfiable and eliminating an arbitrary or-subformula yields a satisfiable formula.*

The set of formulas in MF^ with cohesion k is denoted as $MF^*(k)$.*

Formulas in MF^* with quantifiers are termed minimal false and formulas without quantifiers are called minimal unsatisfiable. For example, the following formula $\Phi = \exists x \exists y ((x \wedge \neg x) \vee (y \wedge \neg y))$ is in MF^* . The formula is false and the or-subformulas are x , $\neg x$, y , and $\neg y$. The deletion of an arbitrary or-subformula leads to a true formula. For the cohesion we see that $D(\Phi) = 1 + \text{and}(\Phi) - v_{\exists}(\Phi) = 1 + 2 - 2 = 1$. Please note that the matrix of Φ is not in CNF. Applying the Tseitin procedure with Φ results in a formula $\exists x \exists y \exists z_1 \exists z_2 ((z_1 \vee x) \wedge (z_1 \vee \neg x) \wedge (\neg z_1 \vee \neg z_2) \wedge (z_2 \vee y) \wedge (z_2 \vee \neg y))$, which is in QCNF. The formula is unsatisfiable and after eliminating an arbitrary clause the formula is satisfiable, i.e., it is in MF.

Lemma 2 *(Equivalence)*

1. $MF^* \cap QCNF = MF$.
2. $\Phi \in MF^* \Leftrightarrow pr(\Phi) \in MF^*$.
3. If Φ is in prenex form, then $\Phi \in MF^* \Leftrightarrow ts(\Phi) \in MF^*$.

Proof: *Ad 1:* Any minimal false formula in QCNF contains at least two clauses. Since the or-subformulas of a formula in QCNF with more than one clause are exactly the clauses, we immediately obtain our desired result.

Ad 2: It is not hard to see that if ψ is an or-subformula then it is also an or-subformula of $pr(\Phi)$. Since the prenexing procedure preserves the satisfiability, the equivalence holds.

Ad 3: Since the Tseitin procedure $\text{ts}(\Phi)$ preserves the satisfiability, we only consider false formulas. A formula Φ is minimal false, if after the deletion of an arbitrary or-subformula the resulting formula is true.

(\Rightarrow) Please note, that in each step of the Tseitin procedure ts , a subformula of the form $(\alpha \wedge \beta) \vee \sigma$ is replaced by $(z \vee \alpha) \wedge (z \vee \beta) \wedge (\neg z \vee \sigma)$ for a new variable z , and $\exists z$ is added to the prefix. Thus, it is sufficient to show that such a step always preserves the minimal falsity.

Suppose $\Phi := Q\varphi$ is minimal false, and $(\alpha \wedge \beta) \vee \sigma$ is a subformula of φ . Let $\Phi' := Q\exists z\varphi'$ be the new formula, where φ' is obtained by the above-mentioned replacement. Please notice, that $(z \vee \alpha)$, $(z \vee \beta)$, and $(\neg z \vee \sigma)$ are or-subformulas in the new formula. It remains to show, that for any or-subformula of the new formula, after the deletion of the or-subformula the formula is true. Let ψ be an arbitrary or-subformula of φ' . We shall proceed by a case distinction on or-subtrees:

Case 1: ψ contains $(z \vee \alpha) \wedge (z \vee \beta) \wedge (\neg z \vee \sigma)$. That is, ψ is obtained from a subformula θ of φ which contains $(\alpha \wedge \beta) \vee \sigma$. It is easy to see that θ must be an or-subtree of φ . Further, the result of deleting ψ from φ' is the same as the result of deleting θ from φ . Thus, deleting ψ from Φ' results in a satisfiable formula.

Case 2: $(z \vee \alpha) \wedge (z \vee \beta) \wedge (\neg z \vee \sigma)$ does not occur in ψ . Since z and $\neg z$ are not or-subformula in φ' , either ψ lies inside of α , β or σ , or outside of all them. Obviously, the deletion of ψ from φ' is the same as the result obtained by deleting ψ from φ and then applying the Tseitin procedure. Since Φ is minimal false and the Tseitin procedure preserves the satisfiability, deleting ψ from Φ' results in a satisfiable formula.

Case 3: ψ is one of the new or-subformulas: $(z \vee \alpha)$, $(z \vee \beta)$, and $(\neg z \vee \sigma)$. W.l.o.g. we only consider the case when ψ is $(z \vee \alpha)$. Suppose α contains no occurrence of \wedge -symbol, i.e., α is a disjunction of some literals, then α is an or-subformula of φ . The assertion clearly holds since Φ is minimal false and $\beta \vee \sigma$ is equivalent to $(z \vee \beta) \wedge (\neg z \vee \sigma)$. So, we assume that α contains the symbol \wedge . Then in the initial formula φ , α must contain an or-subformula. Now we delete an or-subformula from α , the resulting formula is denoted by α_1 . Since Φ is minimal false, $Q\varphi_1$ is satisfiable. Here φ_1 is the result of deleting from φ the above mentioned or-subformula occurring in α . Let φ'_1 be the result of deleting $z \vee \alpha$ from φ' . To show the satisfiability of $Q\exists z\varphi'_1$ it is sufficient to prove the following statement: Each satisfying truth assignment of φ_1 can be extended to a satisfying truth assignment of φ'_1 . Now let v be a truth assignment such that $v(\varphi_1) = 1$. It is enough to prove that v can be extended to v' such that $v((\alpha_1 \wedge \beta) \vee \sigma) = v'((z \vee \beta) \wedge (\neg z \vee \sigma))$. We omit the proof since it is easy but tedious.

(\Leftarrow) Let $Q\exists z\varphi'$ be minimal false. Consider an arbitrary or-subformula ψ in φ . Let φ_1 be obtained by deleting ψ from φ . We need to show that $Q\varphi_1$ is satisfiable. We proceed by a case distinction.

Case 1: ψ occurs in α , β , or σ . W.l.o.g. we only consider the case when ψ occurs in α . We delete ψ from α , the result is denoted as α_1 . And let φ'_1 denote the result obtained from φ' by deleting ψ . Now it is easy to see that $Q\exists z\varphi'_1$ can be obtained from $Q\varphi_1$ by a step of the Tseitin procedure. Thus, they are satisfiability equivalent. Please note, that ψ is also an or-subformula of φ' . Hence, $Q\exists z\varphi'_1$ is satisfiable, and therefore $Q\varphi_1$ is satisfiable.

Case 2: ψ is $(\alpha \wedge \beta) \vee \sigma$. Please notice that the root the formula tree T_ψ is a \vee -node. Then in T_φ the predecessor node of the root of T_ψ must be a \wedge -node. That is, $\gamma \wedge ((\alpha \wedge \beta) \vee \sigma)$ is a

subformula of φ for some γ . And accordingly, $\gamma \wedge (z \vee \alpha) \wedge (z \vee \beta) \wedge (\neg z \vee \sigma)$ is a subformula of φ' . Now it is obvious that after deleting one or two of the formulas $(z \vee \alpha), (z \vee \beta), (\neg z \vee \sigma)$ the remaining formulas are still or-subformulas. Please note, that φ_1 can also be obtained from φ' by the deletion of $(z \vee \alpha), (z \vee \beta), (\neg z \vee \sigma)$. By the minimal falsity of $Q\exists z\varphi'$, $Q\varphi_1$ is satisfiable.

Case 3: Not Case 1 or 2. That is, either ψ properly contains $(\alpha \wedge \beta) \vee \sigma$ or has no intersection with it. Then we can change the order of the replacement and the deletion without changing the result. Since the Tseitin procedure preserves the satisfiability, we get the satisfiability of $Q\varphi_1$. ■

For MF, the class of minimal false formulas in QCNF, we know that MF is PSPACE-complete, MF(1) is solvable in polynomial time, and minimal false formulas always have a deficiency greater than or equal to 1. Because the prenexing procedure pr and the Tseitin procedure ts preserve the cohesion and the minimal falsity (see Lemma 1(1) and Lemma 2), the propositions remain valid for MF^* and $MF^*(k)$.

Theorem 1 (*Quantified Boolean formulas in negation normal form*)

1. MF^* is PSPACE -complete.
2. For $\Phi \in MF^*$, we have $D(\Phi) \geq 1$.
3. For fixed k : $MF^*(k)$ is in D^P iff $MF(k)$ is in D^P .
4. $MF^*(1)$ is solvable in polynomial time.

For propositional formulas the classes MU^* and $MU^*(k)$ are the quantifier-free formulas in MF^* and $MF^*(k)$. Besides the D^P -completeness of MU [6] and the lower bound for the deficiency of minimal unsatisfiable formulas, it has been shown that for fixed deficiency k the minimal unsatisfiability can be decided in polynomial time. By the property of the Tseitin procedure, and the results known for MU and $MU(k)$, the following theorem holds.

Theorem 2 (*Propositional formulas in negation normal form*)

1. MU^* is D^P -complete.
2. For $\alpha \in MU^*$ we have $D(\alpha) \geq 1$.
3. $MU^*(k)$ is solvable in polynomial time.

For propositional formulas there is an equivalent definition of minimal unsatisfiability.

Definition 3 (MU_L)

$MU_L = \{\alpha \mid \alpha \text{ is unsatisfiable and replacing an arbitrary occurrence of a literal by its complement leads to a satisfiable formula}\}$

We will show that in the propositional case MU^* equals MU_L . For quantified formulas we give an counter-example later on.

Lemma 3 (*Equivalence*)

For propositional formulas we have $MU^* = MU_L$.

Proof: Suppose, we have an unsatisfiable formula $\alpha = \alpha_1 \wedge \cdots \wedge \alpha_n \in \text{CNF}$. Let L be a literal in some clause, say α_1 . If $\alpha \in MU_L$ then $\alpha' = \alpha_1[L/\neg L] \wedge \alpha_2 \wedge \cdots \wedge \alpha_n$ is satisfiable. That implies the satisfiability of $\alpha_2 \wedge \cdots \wedge \alpha_n$. Hence, we have shown $MU_L \cap \text{CNF} \subseteq MU$. Now, let α be in MU . Then $\alpha' = \alpha_2 \wedge \cdots \wedge \alpha_n$ is satisfiable. For any literal L in α_1 , every satisfying truth assignment I for α' must make L false. That means, we have $I(L) = 0$. Hence, we get $I(\alpha_1[L/\neg L])=1$ and the satisfiability of the formula $\alpha_1[L/\neg L] \wedge \alpha_2 \wedge \cdots \wedge \alpha_n$. That shows $MU \subseteq MU_L \cap \text{CNF}$.

Because of Lemma 2, it suffices to show that MU_L is closed under the Tseitin procedure $ts(\varphi)$. That means, we have to show for false formulas $\varphi : \varphi \in MU_L$ iff $ts(\varphi) \in MU_L$. The direction from right to left follows immediately from the fact that the Tseitin procedure preserves the satisfiability. For the other direction we assume that a subformula $\psi := (\alpha \wedge \beta) \vee \sigma$ in φ has been replaced by $\psi' := (z \vee \alpha) \wedge (z \vee \beta) \wedge (\neg z \vee \sigma)$, where z is a new variable. And let φ' be resulting formula. The only critical case is the substitution of occurrences of the new literals z and $\neg z$.

Suppose, $\neg z$ has been replaced by z in $(\neg z \vee \sigma)$. Then z occurs only positively in the new formula. Since φ becomes satisfiable after a substitution of some literal in α (say the result is α_1), let v be a satisfying truth assignment of the formula after the substitution in φ . If v makes $(\alpha_1 \wedge \beta) \vee \sigma$ true, then we make $(z \vee \alpha) \wedge (z \vee \beta) \wedge (z \vee \sigma)$ true by setting $z = 1$. Else, v must make σ false, then we can make $(z \vee \alpha) \wedge (z \vee \beta) \wedge (z \vee \sigma)$ false by setting $z = 0$. Now we can see that φ' becomes satisfiable when $\neg z$ is replaced by z .

Now we suppose that one of the occurrences of z has been replaced by $\neg z$, w.l.o.g. say in $(z \vee \alpha)$. Since a substitution of a literal in α in the formula φ leads to satisfiable formula, let v be a satisfying truth assignment for this modified formula. Since φ is unsatisfiable, $v(\sigma) = 0$, otherwise, v would satisfy φ . If v makes $(\alpha_1 \wedge \beta) \vee \sigma$ false, then we can make $(\neg z \vee \alpha) \wedge (z \vee \beta) \wedge (\neg z \vee \sigma)$ false by setting $z = 1$. Else, β must be true under v , we now set $z = 0$, then the formula $(\neg z \vee \alpha) \wedge (z \vee \beta) \wedge (\neg z \vee \sigma)$ is true. Therefore, φ' becomes satisfiable if one occurrence of z is replaced by $\neg z$. That completes the proof. ■

Similar to the definition of MU_L , we can define MF_L to be the class of unsatisfiable quantified Boolean formulas in which replacing any literal by its complement results in satisfiable formulas. Clearly, MF_L is a subclass of MF^* , however, the inverse is invalid, that is, $MF^* \neq MF_L$. The following formula Φ is a counter-example.

$$\Phi := \exists y_1 \exists y_2 \forall x \exists y \begin{pmatrix} x & \neg x & y_1 & y_2 & \neg y_1 \\ \neg y & y & x & \neg x & \neg y_2 \\ & & y & \neg y & y \end{pmatrix}$$

Here, each column represents a clause. It is easy to check that Φ is minimal false. Replacing in the last clause the literal y by $\neg y$ leads to the formula

$$\Phi' := \exists y_1 \exists y_2 \forall x \exists y \begin{pmatrix} x & \neg x & y_1 & y_2 & \neg y_1 \\ \neg y & y & x & \neg x & \neg y_2 \\ & & y & \neg y & \neg y \end{pmatrix}$$

which is still false. Thus, we have $\Phi \notin MF_L$.

Remark: The cohesion depends on the number of existential and free variables, but not on the number of universal variables. The generalization to all variables leads to a measure, which is not very useful for investigating minimal falsity problems. For example, for $D_{all}(\Phi) := 1 + \text{and}(\Phi) - v(\Phi)$ and fixed natural numbers k it can easily be shown that $\{\Phi \mid \Phi \in \text{MF}^* \text{ and } D_{all}(\Phi) = k\}$ is PSPACE-complete [4].

References

- [1] R. Aharoni and N. Linial, Minimal Non–Two–Colorable Hypergraphs and Minimal Unsatisfiable Formulas. *Journal of Combinatorial Theory*, **43** (1986), 196–204.
- [2] H. Fleischner, O. Kullmann, and S. Szeider, Polynomial-time recognition of minimal unsatisfiable formulas with fixed clause-variable deficiency. *Theoretical Computer Science*, **289** (2002) 503–516.
- [3] H. Kleine Büning and T. Lettmann, *Propositional Logic: Deduction and Algorithms*, Cambridge University Press, 1999.
- [4] H. Kleine Büning and X. Zhao, Minimal False Quantified Boolean Formulas. *Lecture Notes in Computer Science*, **4121** (2006), 339–352.
- [5] A.R. Meyer and L.J. Stockmeyer, Word Problems Requiring Exponential Time. *Proc. 5th Ann. Symp. on Theory of Computing*, 1973, 1–9.
- [6] C.H. Papadimitriou and D. Wolfe, The complexity of facets resolved. *Journal of Computer Systems Science*, **37** (1988), 2–13.
- [7] G.S. Tseitin, On the Complexity of Derivation in Propositional Calculus. In A.O. Silenko (ed.), *Studies in Constructive Mathematics and Mathematical Logic*, Part II, 1970, 115–125.