

# Inconsistency Measurement based on Variables in Minimal Unsatisfiable Subsets

Guohui Xiao<sup>1</sup> and Yue Ma<sup>2</sup>

**Abstract.** Measuring inconsistency degrees of knowledge bases (KBs) provides important context information for facilitating inconsistency handling. Several semantic and syntax based measures have been proposed separately.

In this paper, we propose a new way to define inconsistency measurements by combining semantic and syntax based approaches. It is based on counting the variables of minimal unsatisfiable subsets (MUSes) and minimal correction subsets (MCSes), which leads to two equivalent inconsistency degrees, named  $ID_{MUS}$  and  $ID_{MCS}$ . We give the theoretical and experimental comparisons between them and two purely semantic-based inconsistency degrees: 4-valued and the Quasi Classical semantics based inconsistency degrees. Moreover, the computational complexities related to our new inconsistency measurements are studied. As it turns out that computing the exact inconsistency degrees is intractable in general, we then propose and evaluate an anytime algorithm to make  $ID_{MUS}$  and  $ID_{MCS}$  usable in knowledge management applications. In particular, as most of syntax based measures tend to be difficult to compute in reality due to the exponential number of MUSes, our new inconsistency measures are practical because the numbers of variables in MUSes are often limited or easily to be approximated.

We evaluate our approach on the DC benchmark. Our encouraging experimental results show that these new inconsistency measurements or their approximations are efficient to handle large knowledge bases and to better distinguish inconsistent knowledge bases.

## 1 Introduction

Inconsistency handling has been recognized as an important issue in the field of artificial intelligence. Recently, as the ever expanding amount of logic-based data available in diverse information systems, there is an increasing interests in quantifying inconsistency. This is because it is not fine-grained enough to simply say that two inconsistent knowledge bases contain the same amount of inconsistency. Indeed, it has been shown that analyzing inconsistency can provide useful context information to resolve inconsistency [8, 12, 10, 9, 5]. Furthermore, measuring inconsistency in a knowledge base proves meaningful in different scenarios such as news reports [9], integrity constraints [4], software engineering [20], and semantic annotation [17].

Having been studied for inconsistency handling, minimal inconsistency subsets theories and multi-valued logics are used as two main distinct approaches to define inconsistency metrics, which focus on different views of atomic inconsistency [12]. The former puts atomicity to formulae touched by inconsistency [14, 11, 22, 21].

While the latter puts atomicity to the propositional letters valued as conflictions under the corresponding multi-valued models [3, 8, 9, 4, 19, 24, 18]. Usually, the above two approaches are assumed to have their own suitable application scenarios [12]. However, there have been an increasing requirement recently to define inconsistency measures by combining these two aspects [10, 11]. To achieve this, we propose, in this paper, a novel approach by considering the number of conflicting atoms in MUSes and MCSes of a knowledge base.

Our work is based on the observations that MUSes and MCSes are cornerstones of analyzing thus measuring inconsistencies [16] and that various multi-valued semantics show interesting properties for measuring inconsistency [8, 4, 9, 19]. Indeed, we find that the measurements merely by multi-valued semantics can easily give a same degree for many different knowledge bases because of the neglect of syntax differences (see Section 6 for examples). Similarly, measurements merely based on the number of MUSes or the cardinalities of MUSes, such as  $MIV_D(K)$  and  $MIV_C(K)$  in [11], are blunt to evaluate inconsistency values of some knowledge bases. For example, suppose that there are  $n$  groups  $\{g_1, \dots, g_n\}$  of policies to poll on. The poll result is represented by the set  $\{\gamma_1, \dots, \gamma_n\}$ , where  $\gamma_i$  is defined as follows: if a policy  $a$  in  $g_i$  is supported (resp. denied) by some people,  $a$  (resp.  $\neg a$ ) is a conjunct of  $\gamma_i$ ; otherwise,  $a$  and  $\neg a$  are left out of  $\gamma_i$ . For simplicity, we consider only one group with two policies  $\{a, b\}$ . So  $K = \{a \wedge \neg a\}$  represents a poll that there are people supporting and people denying  $a$ , but no opinion on  $b$ . And  $K' = \{a \wedge \neg a \wedge b \wedge \neg b\}$  indicates conflicting opinions on both  $a$  and  $b$ . So we consider the first poll, without explicit confliction on  $b$ , is less contradictory than the second. However, both  $K$  and  $K'$  have only one minimal inconsistent subset with the same cardinality. So they cannot be distinguished by  $MIV_D(K)$  or  $MIV_C(K)$ . But if we consider the conflicting atoms in each, we get that  $K'$  is more inconsistent, which accords with our intuition. Additionally, as most of syntax based measures tend to be difficult to compute in reality due to the exponential number of MUSes, our new inconsistency measures are experimentally shown practical because the numbers of variables in MUSes are often limited or easily to be approximated (see Section 6).

Although many inconsistency measures have been studied, the complete comparison of them is challenging. We have known some positive answers: for example, the inconsistency degrees under 4-valued semantics, 3-valued semantics, LPM are the same, but different from the one based on Quasi Classical semantics [24]. Also, we have negative answers that many inconsistency measures are incompatible [5]. In this paper, we show that the two proposed inconsistency degrees are equivalent to each other. Moreover, they are compatible with multi-valued based inconsistency measures: they are

<sup>1</sup> Vienna University of Technology, Austria, email: xiao@kr.tuwien.ac.at

<sup>2</sup> Université Paris Nord – CNRS, France, email: yue.ma@lipn.univ-paris13.fr

always between the inconsistency degrees based on 4-valued semantics and QC semantics. By examples, we can see that the new novel inconsistency degrees can compensate the bluntness of the widely studied multi-value based inconsistency degrees.

For computing the inconsistency degrees, we first study the computational complexities of the proposed inconsistency degrees and find that they are theoretically harder than multi-valued based inconsistency degrees. To handle this problem, we finally propose an anytime approximating algorithm which is shown efficient to handle large knowledge bases, and even outperforms the state-of-the-art approaches for computing multi-valued based inconsistency degrees [24] on some real use case data from the DC benchmark [23].

The rest of the paper is structured as follows: preliminaries are given in the next section, followed by the definition and properties of our inconsistency measurements. The theoretical complexities are given in Section 4 and an anytime algorithm for practical computing the proposed inconsistency measures are detailed in Section 5. The evaluation of our approach is done on the DC benchmark data in Section 6. Related work is discussed in Section 7. Due to space limitation, we only give proof sketches through the paper.

## 2 Preliminaries

Given a finite set of propositional variables  $\mathcal{A} = \{p_1, \dots, p_n\}$ , a literal  $l$  is a variable  $p$  or its negation  $\neg p$ . A clause  $C = l_1 \vee l_2 \vee \dots \vee l_k$  is a disjunction of literals. W.o.l.g, we assume that a knowledge base (KB) is a CNF formula, i.e., a conjunction of clauses, represented as a set of clauses  $K = \{C_1, C_2, \dots, C_m\}$ . Indeed, all the definitions and results in the following paper could be extended into KBs of arbitrary propositional formulas in a straightforward way. We denote by  $Var(K)$  the set of variables occurring in  $K$  and  $|S|$  the cardinality of a set  $S$ .

### 2.1 MUS and MCS

A Minimal Unsatisfiable Subset (MUS) is a subset of a KB which is both unsatisfiable and cannot be made smaller without becoming satisfiable. A Minimal Correction Subset (MCS) is a subset of a KB whose removal from that system results in a satisfiable set of constraints (“correcting” the infeasibility) and which is minimal in the same sense that any proper subset does not have that defining property. Any KB  $K$  can have multiple MUSes and MCSes, potentially exponential in  $|K|$  [16]. Formally, given a KB  $K$ , its MUSes and MCSes are defined as follows:

**Definition 1** A subset  $U \subseteq K$  is an MUS if  $U$  is unsatisfiable and  $\forall C_i \in U, U \setminus \{C_i\}$  is satisfiable.

**Definition 2** A subset  $M \subseteq K$  is an MCS if  $K \setminus M$  is satisfiable and  $\forall C_i \in M, K \setminus (M \setminus \{C_i\})$  is unsatisfiable.

#### 2.1.1 MUS/MCS Duality

For a KB, the relationship of MCSes and MUSes can be stated simply: the set of its MUSes and the set of its MCSes are “hitting set duals” of one another, where the hitting set is defined as follows:

**Definition 3**  $H$  is a hitting set of a set of sets  $\Omega$  if  $\forall S \in \Omega, H \cap S \neq \emptyset$ . A hitting set  $H$  is irreducible if there is no other hitting set  $H'$ , s.t.  $H' \subsetneq H$ .

**Proposition 1 ([16])** Given an inconsistent knowledge base  $K$ :

- A subset  $M$  of  $K$  is an MCS of  $K$  iff  $M$  is an irreducible hitting set of  $MUSes(K)$ ;
- A subset  $U$  of  $K$  is an MUS of  $K$  iff  $U$  is an irreducible hitting set of  $MCSes(K)$ .

**Example 1** Let  $K = \{p, \neg p, p \vee q, \neg q, \neg p \vee r\}$ . Then  $MUSes(K) = \{\{p, \neg p\}, \{\neg p, p \vee q, \neg q\}\}$  and  $MCSes(K) = \{\{\neg p\}, \{p, p \vee q\}, \{p, \neg q\}\}$ . Clearly,  $MUSes(K)$  and  $MCSes(K)$  are hitting set duals of each other.

A free formula of a knowledge base  $K$  is a formula of  $K$  that does not belong to any MUS of  $K$ . This means that this formula has nothing to do with the conflicts of the KB. In [11], an inconsistency measure by number of MUSes of  $K$  is defined as  $I_{MI}(K) = |MUSes(K)|$ .

The state-of-the-art MCS/MUS finders are highly optimized and scalable. Some of them are CAMUS<sup>3</sup>[16], and HycAM<sup>4</sup>[7].

## 2.2 Inconsistency Measures by Multi-Valued Semantics

Different from classical two-valued (true, false) semantics, multi-valued semantics (3-valued, 4-valued, Lpm, and Quasi Classical), use a third truth value  $B$  to stand for the contradictory information, thus able to measure inconsistency. Since 3-valued, 4-valued, and Lpm based inconsistency degrees are the same, but different from the one based on Quasi Classical [24], only 4-valued and Quasi Classical inconsistency degree are necessarily discussed and denoted by  $ID_4$  and  $ID_Q$ , respectively.

Let  $\mathcal{I}$  be a multi-valued interpretation under  $i$ -semantics ( $i = 4, Q$ ). Then  $Conflict(K, \mathcal{I}) = \{p \in Var(K) \mid p^{\mathcal{I}} = B\}$  is called the *conflicting set* of  $\mathcal{I}$  with respect to  $K$ , simply written  $Conflict(\mathcal{I})$  when  $K$  is clear from the context. The *preferred  $i$ -model* set, written  $PM_i(K)$ , is defined as  $PM_i(K) = \{\mathcal{I} \mid \mathcal{I} \models_i K \text{ and } \forall \mathcal{J} \models_i K : |Conflict(\mathcal{J})| \geq |Conflict(\mathcal{I})|\}$ , where  $\mathcal{I} \models_i K$  means that  $\mathcal{I}$  is a model of  $K$  under  $i$ -semantics as defined in subsequent subsections. Then the *inconsistency degree of a KB  $K$*  w.r.t.  $\mathcal{I}$  is defined as  $ID_i(K, \mathcal{I}) = \frac{|Conflict(K, \mathcal{I})|}{|Var(K)|}$ . Finally The *inconsistency degree of  $K$*  under  $i$ -semantics is defined as  $ID_i(K) = \frac{|Conflict(K, \mathcal{I})|}{|Var(K)|}$ , for some  $\mathcal{I} \in PM_i(K)$ .

### 2.2.1 Four-valued Semantics

The set of truth values for 4-valued semantics [1] contains four elements: *true*, *false*, *unknown* and *both*, written by  $t, f, N, B$ , respectively. The truth value  $N$  allows to express incompleteness of information. The four truth values together with the ordering  $\preceq$  defined below form a lattice  $FOUR = (\{t, f, B, N\}, \preceq)$ :  $f \preceq N \preceq t, f \preceq B \preceq t, N \not\preceq B, B \not\preceq N$ . The 4-valued semantics of connectives  $\vee, \wedge$  are defined according to the upper and lower bounds of two elements based on the ordering  $\preceq$ , respectively, and the operator  $\neg$  is defined as  $\neg t = f, \neg f = t, \neg B = B$ , and  $\neg N = N$ .

A 4-valued interpretation  $\mathcal{I}$  is a 4-model of a KB  $K$ , denoted  $\mathcal{I} \models_4 K$ , if and only if for each formula  $\phi \in K, \phi^{\mathcal{I}} \in \{t, B\}$ .

**Example 2** Let  $K = \{p, \neg p \vee q, \neg q \vee r, \neg r, s \vee u\}$ . Consider three 4-valued models  $\mathcal{I}_1, \mathcal{I}_2$  and  $\mathcal{I}_3$  of  $K$  defined as:

$$\begin{aligned} p^{\mathcal{I}_1} &= t, q^{\mathcal{I}_1} = B, r^{\mathcal{I}_1} = f, s^{\mathcal{I}_1} = t, u^{\mathcal{I}_1} = N; \\ p^{\mathcal{I}_2} &= B, q^{\mathcal{I}_2} = f, r^{\mathcal{I}_2} = B, s^{\mathcal{I}_2} = t, u^{\mathcal{I}_2} = N; \\ p^{\mathcal{I}_3} &= B, q^{\mathcal{I}_3} = B, r^{\mathcal{I}_3} = B, s^{\mathcal{I}_3} = t, u^{\mathcal{I}_3} = N. \end{aligned}$$

<sup>3</sup> <http://www.eecs.umich.edu/~lifffiton/camus/>

<sup>4</sup> <http://www.cril.univ-artois.fr/~piette/#resources>

Obviously,  $ID_4(K, I_1) = 1/5$ ,  $ID_4(K, I_2) = 2/5$ ,  $ID_4(K, I_3) = 3/5$ . Moreover, since  $K$  is 2-valued unsatisfiable, every 4-model of  $K$  contains at least one contradiction. So  $ID_4(K) = 1/5$ .

### 2.2.2 Quasi-Classical Semantics (Q-semantics)

For the propositional variables set  $\mathcal{A}$ , let  $\mathcal{A}^\pm$  be a set of objects defined as  $\mathcal{A}^\pm = \{+p, -p \mid p \in \mathcal{A}\}$ .

**Definition 4 (Q-models[2])** Suppose  $p \in \mathcal{A}$ ,  $C_1, \dots, C_m$  are clauses and  $l_1, \dots, l_n$  are literals. For  $\mathcal{I} \subseteq \mathcal{A}^\pm$ , the Q-satisfiability relation  $\models_Q$  is defined as follows:

$$\begin{aligned} \mathcal{I} \models_Q p & \quad \text{iff } +p \in \mathcal{I}; \\ \mathcal{I} \models_Q \neg p & \quad \text{iff } -p \in \mathcal{I}; \\ \mathcal{I} \models_Q l_1 \vee \dots \vee l_n & \quad \text{iff } [\mathcal{I} \models_Q l_1 \text{ or } \dots \text{ or } \mathcal{I} \models_Q l_n] \\ & \quad \text{and } [\text{for all } i, \mathcal{I} \models_Q \neg l_i \text{ implies} \\ & \quad \quad \mathcal{I} \models_Q l_1 \vee \dots \vee l_{i-1} \vee l_{i+1} \vee \dots \vee l_n]; \\ \mathcal{I} \models_Q \{C_1, \dots, C_m\} & \quad \text{iff } \mathcal{I} \models_Q C_i (1 \leq i \leq m). \end{aligned}$$

Q-semantics can also be regarded as assigning one of the four truth values  $\{B, t, f, N\}$  to symbols in  $\mathcal{A}$  in the following way, which enables the uniform way to define inconsistency degrees as above.

$$p^{\mathcal{I}} = \begin{cases} t & \text{iff } +p \in \mathcal{I} \text{ and } -p \notin \mathcal{I}; \\ f & \text{iff } +p \notin \mathcal{I} \text{ and } -p \in \mathcal{I}; \\ B & \text{iff } +p \in \mathcal{I} \text{ and } -p \in \mathcal{I}; \\ N & \text{iff } +p \notin \mathcal{I} \text{ and } -p \notin \mathcal{I}. \end{cases}$$

**Example 3 (Example 2 Contd.)** Consider again the 4-models  $\mathcal{I}_1$ ,  $\mathcal{I}_2$  and  $\mathcal{I}_3$  of  $K$ . By definition 4,  $\mathcal{I}_1$  and  $\mathcal{I}_2$  are not Q-models of  $K$ , although they are 4-models of  $K$ . In fact,  $\mathcal{I}_3$  is a preferred Q-model of  $K$  and we have  $ID_Q(K) = ID_Q(K, \mathcal{I}_3) = 3/5$ .

## 3 Inconsistency Degrees by MUS and MCS

MUSes and MCSes are fundamental features in characterizing the inconsistency of a knowledge base. In this section, we propose two inconsistency degrees through MUSes and MCSes respectively. We prove that these two inconsistency degrees are actually equivalent to each other and with desirable properties. More interestingly, we find the relation between the proposed syntax-semantics combined measures and purely semantic based inconsistency degrees  $ID_4$  and  $ID_Q$ . Their experimental comparison is given in Section 6.

As we have seen from the example given in the introduction, considering the cardinality of variables occurring in MUSes can provide a more fine-grained way for measuring inconsistency. This intuition is formalized by the following definition.

**Definition 5** For a given set of variables  $S$  and a given knowledge base  $K$  such that  $\text{Var}(K) \subseteq S$ , its MUS-variable based inconsistency degree, written  $ID_{MUS}(K)$ , is defined as:

$$ID_{MUS}(K) = \frac{|\text{Var}(MUSes(K))|}{|S|}.$$

That is,  $ID_{MUS}(K)$  is the ratio of the number of variables occurring in some MUSes divided by the amount of all concerned variables in  $S$ . Obviously, this is a new way to measure the proportion of the language touched by the inconsistency in the knowledge base  $K$ . Note that  $S$  is provided to compare different knowledge bases, as shown in Example 5. When  $S$  is not explicitly given, we assume that  $S = \text{Var}(K)$ , that is, we only consider variables occurring in the KB.

**Example 4 (Example 1 contd.)** Let  $K = \{p, \neg p, p \vee q, \neg q, \neg p \vee r\}$  and  $S = \text{Var}(K)$ ,  $MUSes(K) = \{\{p, \neg p\}, \{\neg p, p \vee q, \neg q\}\}$ . Then  $ID_{MUS}(K) = 2/3$ .

**Example 5** For  $K = \{a \wedge \neg a\}$  and  $K' = \{a \wedge \neg a \wedge b \wedge \neg b\}$  as given in the introduction, let  $S = \text{Var}(K) \cup \text{Var}(K') = \{a, b\}$ . Then we have  $MUSes(K) = \{\{a \wedge \neg a\}\}$  and  $MUSes(K') = \{\{a \wedge \neg a \wedge b \wedge \neg b\}\}$ ,  $ID_{MUS}(K) = 1/2$  and  $ID_{MUS}(K') = 1$ . So under  $ID_{MUS}$ ,  $K'$  is more inconsistent than  $K$ .

Similarly to  $ID_{MUS}(K)$ , we can define another inconsistency degree through MCS as follows:

**Definition 6** For a given set of variables  $S$  and a given knowledge base  $K$  such that  $\text{Var}(K) \subseteq S$ , its MCS-variable based inconsistency degree, written  $ID_{MCS}(K)$ , is defined as follows:

$$ID_{MCS}(K) = \frac{|\text{Var}(MCSes(K))|}{|S|}.$$

**Example 6 (Example 1 contd.)** Let  $K = \{p, \neg p, p \vee q, \neg q, \neg p \vee r\}$  and  $S = \text{Var}(K)$ ,  $MCSes(K) = \{\{-p\}, \{p, p \vee q\}, \{p, \neg q\}\}$ , then  $ID_{MCS}(K) = 2/3$ .

In the examples 4 and 6, the MUS-variable and the MCS-variable based inconsistency degrees are equal. Actually, this is not a coincidence as shown by the following proposition followed by the duality of MUS and MCS.

**Proposition 2** For any CNF KB  $K$ ,  $ID_{MUS}(K) = ID_{MCS}(K)$ .

*Proof.* From the duality of MUS and MCS, we have  $\bigcup MUSes(K) = \bigcup MCSes(K)$ . Then the conclusion follows directly.  $\square$

By this proposition, in the rest of the paper, the discussion is only about  $ID_{MUS}(K)$ , unless otherwise stated.

**Proposition 3** The  $ID_{MUS}(K)$  satisfies the following properties, for any knowledge base  $K$  and any formulae  $\alpha, \beta$  with  $\text{Var}(\alpha), \text{Var}(\beta) \subseteq \text{Var}(K)$ :

- $ID_{MUS}(K) = 0$  iff  $K$  is consistent;
- $ID_{MUS}(K \cup \{\alpha\}) \geq ID_{MUS}(K)$ ;
- If  $\alpha$  is a free formula of  $K \cup \{\alpha\}$ ,  $ID_{MUS}(K \cup \{\alpha\}) = ID_{MUS}(K)$ .

The above three properties are called *consistency*, *monotony* and *free formula independence* respectively in [11].

### 3.1 Relationship between $ID_4(K)$ and $ID_{MUS}(K)$

**Lemma 4** Let  $U$  be an MUS, and  $p \in \text{Var}(U)$ . Then there exists a 4-valued model  $\mathcal{I}$  of  $U$ , such that  $p^{\mathcal{I}} = B$  and  $x^{\mathcal{I}} \in \{t, f\}$ , if  $x \neq p$ .

*Proof (sketch).* Suppose that  $p \in \text{Var}(C)$ , for some  $C \in U$ . Since  $U$  is minimal unsatisfiable, there exists a classical model  $\mathcal{J}$  for  $U \setminus \{C\}$ . By changing the assignment of  $p$  to  $B$ , we get a 4-model  $\mathcal{I}$  of  $U$ .  $\square$

**Corollary 5** Let  $MUSes(K) = \{U_1, \dots, U_n\}$ , and  $H$  be a hitting set of  $\{\text{Var}(U_1), \dots, \text{Var}(U_n)\}$ . Then there exists a 4-model  $\mathcal{I}$  of  $K$ , such that  $x^{\mathcal{I}} = B$ , if  $x \in H$ ; and  $x^{\mathcal{I}} \in \{t, f\}$ , otherwise.

**Corollary 6** Let  $K$  be a KB and interpretation  $\mathcal{I} \in PM_4(K)$ , then  $\text{Conflict}(\mathcal{I}, K) \subseteq \text{Var}(MUSes(K))$ .

**Corollary 7** Let  $U$  be an MUS, then  $ID_4(U) = 1/|\text{Var}(U)|$ .

The following theorem shows that  $ID_4(K)$  can be determined by the cardinality minimal hitting sets of  $MUSes(K)$ .

**Theorem 8** For a given KB  $K$ ,

$$ID_4(K) = \frac{\min_H \{ |H| \mid \forall U \in MUSes(K), \text{Var}(U) \cap H \neq \emptyset \}}{|\text{Var}(K)|}.$$

*Proof (sketch).* Note that the variables in the conflicting set of the preferred 4-models are the cardinality minimal hitting sets of  $\{\text{Var}(U) \mid U \in MUSes(K)\}$ .  $\square$

**Corollary 9**  $ID_{MUS}(K) \geq ID_4(K)$ .

**Example 7** Let  $K = \{r, \neg r, \neg p, p \vee q, \neg q\}$ , then  $MUSes(K) = \{U_1 = \{r, \neg r\}, U_2 = \{\neg p, p \vee q, \neg q\}\}$ . So we have two cardinality minimal hitting sets of  $\{\text{Var}(U_1), \text{Var}(U_2)\}$ , i.e.  $\{r, p\}$  and  $\{r, q\}$ . Hence  $ID_4(K_2) = 2/3 < 1 = ID_{MUS}(K)$ .

### 3.2 Relationship between $ID_Q(K)$ and $ID_{MUS}(K)$

Firstly, we introduce necessary notations. Let  $S$  be a set of clauses, the resolution closure of  $S$ , denoted  $RC(S)$ , is the minimal set of clauses satisfying the following conditions:

1. If  $C = l_1 \vee \dots \vee l_n \in S$ , then  $C \in RC(S)$ .
2. If  $C_1 = l_1 \vee \dots \vee l_n \vee c \in RC(S)$ ,  $C_2 = l'_1 \vee \dots \vee l'_m \vee \neg c \in RC(S)$ , then  $Res(C_1, C_2, c) = l_1 \vee \dots \vee l_n \vee l'_1 \vee \dots \vee l'_m \in RC(S)$ . In particular, if  $C_1 = c \in RC(S)$  and  $C_2 = \neg c \in RC(S)$ , then the empty clause  $\square \in RC(S)$ .
3. Every clause in  $RC(S)$  can be formed by the above rules.

Note that here we do not allow the resolve of the empty clause  $\square$  with a non-empty clause.  $\square$

**Proposition 10** If  $U$  is an MUS, then the resolution closure of  $U$  contains all the literals formed by atomic letters occurring in  $U$ , i.e.  $RC(U) \supseteq \{p, \neg p \mid p \in \text{Var}(U)\}$ .

*Proof (sketch).* The existence of a resolution path from a clause  $C$  to  $\square$  in the resolution sequence  $ReSeq(\square)$  is defined inductively:

- There exists a resolution path from  $\square$  to  $\square$ ;
- If  $\square = Res(p, \neg p, p)$ , then there is a resolution path from  $p$  to  $\square$  and a resolution path from  $\neg p$  to  $\square$  in  $ReSeq(\square)$ ;
- If  $C = Res(C_1, C_2, p)$  and there exists a path from  $C$  to  $\square$ , then there exists a resolution path from  $C_i$  to  $\square$  for  $i = 1, 2$ .

By noticing that  $U$  is an MUS, we have that all clauses  $C$  in  $U$  should have a resolution path to  $\square$ . For each clause  $C \in U$ , since there is a path from  $C$  to  $\square$ , w.l.o.g, suppose the resolved atoms along this path are  $path_C = \{p_1, \dots, p_m\}$ . The conclusion holds by induction on  $p_i$  ( $1 \leq i \leq m$ ).  $\square$

**Lemma 11** Let  $U$  be an MUS, then  $U$  has only one  $Q$ -model which assigns  $B$  to all of its variables. Hence  $ID_Q(U) = 1$ .

*Proof (sketch).* By Proposition 10 and the fact that a resolution sequence defined above is also a valid resolution sequence under  $Q$ -semantics, we have  $U \models_Q p, U \models_Q \neg p$ , for all  $p \in \text{Var}(U)$ , that is, for any  $Q$ -model  $\mathcal{I}$  of  $U$ ,  $p^{\mathcal{I}} = B$ .  $\square$

**Proposition 12** Let  $K$  be a KB and  $\mathcal{I} \in PM_Q(K)$ , then  $\text{Conflict}(\mathcal{I}, K) \supseteq \text{Var}(MUSes(K))$ .

**Corollary 13** Let  $K$  be a KB, then  $ID_Q(K) \geq ID_{MUS}(K)$ .

**Example 8** Let  $K = \{p, \neg p, p \vee r, \neg p \vee \neg r\}$ . By Definition 4, we have  $ID_Q(K) = 1$ . However,  $K$  has only one MUS:  $MUSes(K) = \{\{p, \neg p\}\}$ . So  $ID_{MUS}(K) = 1/2 < ID_Q(K)$ .

## 4 Computational Complexities

Given a KB  $K$ , the  $ID_{MUS}$  related computational problems can include:

- *Var-in-MUSes*: Given a variable  $x$ , deciding  $x \in \text{Var}(MUSes(K))$ .
- *Size-Vars-in-MUSes $_{\geq k}$*  (resp.  $_{\leq k}$ ,  $_{=k}$ ): Given an integer  $k$ , deciding  $|\text{Var}(MUSes(K))| \geq k$  (resp.  $|\text{Var}(MUSes(K))| \leq k$ ,  $|\text{Var}(MUSes(K))| = k$ ).
- *Size-Vars-in-MUSes*: Computing  $|\text{Var}(MUSes(K))|$ .
- *ID-MUS $_{\geq k}$*  (resp. *ID-MUS $_{\leq k}$* , *ID-MUS $_{=k}$* ): Given a number  $k$ , deciding  $ID_{MUS}(K) \geq k$  (resp.  $ID_{MUS}(K) \leq k$ ,  $ID_{MUS}(K) = k$ ).
- *ID-MUS*: Computing  $ID_{MUS}(K)$ .

**Theorem 14** *Var-in-MUSes* is  $\Sigma_2^p$ -complete.

*Proof (sketch).* This is immediate from the proof of Theorem 5 in [15], where the result is proved for a clause belonging to an MUS, but the query clause is a variable  $w$ .  $\square$

**Theorem 15** *Size-Vars-in-MUSes $_{\geq k}$*  is  $\Sigma_2^p$ -complete.

*Proof (sketch).* Membership is trivial by guess and check. For hardness, we use the same reduction as in theorem 14. There  $w \in \text{Var}(MUSes(K))$  is equivalent to  $|\text{Var}(MUSes(K))| \geq n + 1$ .  $\square$   
By Theorem 15, the following result is straightforward.

**Corollary 16** *Size-Vars-in-MUSes $_{\leq k}$*  is  $\Pi_2^p$ -complete; *Size-Vars-in-MUSes $_{=k}$*  is  $D_2^p$ -complete. <sup>5</sup>

**Theorem 17** *Size-Vars-in-MUSes* is in  $FP^{\Sigma_2^p[\log]}$ . <sup>6</sup>

*Proof (sketch).* We can easily develop an algorithm accessing logarithmically many times to an oracle solving *Size-Vars-in-MUSes $_{\leq k}$*  via binary search.  $\square$

**Corollary 18** *ID-MUS $_{\geq k}$*  (resp. *ID-MUS $_{\leq k}$* , *ID-MUS $_{=k}$* ) is  $\Sigma_2^p$  (resp.  $\Pi_2^p$ ,  $D_2^p$ )-complete. *ID-MUS* is in  $FP^{\Sigma_2^p[\log]}$ .

The tight complexity bound of problems *Size-Vars-in-MUSes* and *ID-MUS* are still open. We conjecture that they are  $FP^{\Sigma_2^p[\log]}$ -complete. There is a reduction of a generalization of a problem that is  $FP^{NP[\log]}$  complete, viz. computing given Boolean formulas  $F_1, \dots, F_n$ , the number of satisfiable formulas among them [13]. That proof might be lifted to  $\exists \forall$  QBFs.

It turns out that all of these problems are in the second layer of polynomial hierarchy as given below. Recall that the complexity of multi-valued based inconsistency degrees are in the first layer [19, 24]. Therefore  $ID_{MUS}$  is theoretically harder than  $ID_4$  and  $ID_Q$ .

## 5 Algorithm

To solve the high computational complexity of  $ID_{MUS}$ , in this section, we present an anytime algorithm based on existing MUS finder. The state-of-the-art MUS finders usually take two steps: first finding MCSes, then using a hitting set algorithm to find MUSes [16, 7]. For computing  $ID_{MUS}$ , we only need MCSes generated in the first step, as  $ID_{MCS}(K) = ID_{MUS}(K)$ .

MCS finders generate MCSes one by one. These intermediate results can be used to approximate the value of  $ID_{MUS}$ . Every time we get a new MCS, we can update the lower bound of  $ID_{MUS}(K)$ . The pseudo code is described in Algorithm 1.

<sup>5</sup>  $D_2^p$  is, similar to  $D^p$ , the ‘‘conjunction’’ of  $\Sigma_2^p$  and  $\Pi_2^p$ ; e.g. solve a pair  $(\Phi, \Psi)$  of QBFs  $\Phi = \exists \forall E$  and  $\Psi = \forall \exists E'$ .

<sup>6</sup>  $FP^{\Sigma_2^p[\log]}$  is for polynomial time computations with an oracle in  $\Sigma_2^p$ , where the oracle can be accessed only log time often in the size of the input.

**Algorithm 1:** Anytime Algorithm for  $ID_{MUS}(K)$ 


---

**Input:**  $K$ : KB as a set of clauses  
**Output:**  $ID_{MUS}(K)$

```

1  $B \leftarrow \{\}$  // variable set
2  $N \leftarrow |\text{Var}(K)|$ 
3 foreach  $M \in \text{MCSES}(K)$  // call MCS finder
4 do
5    $B \leftarrow B \cup \text{Var}(M)$  // update B
6    $id \leftarrow |B|/N$  // new idmus lower bound
7   print 'id_mus(K)  $\geq$  ',  $id$ 
8 end
9 print 'id_mus(K) = ',  $id$ 
10 return  $id$ 

```

---

## 6 Experimental Evaluation

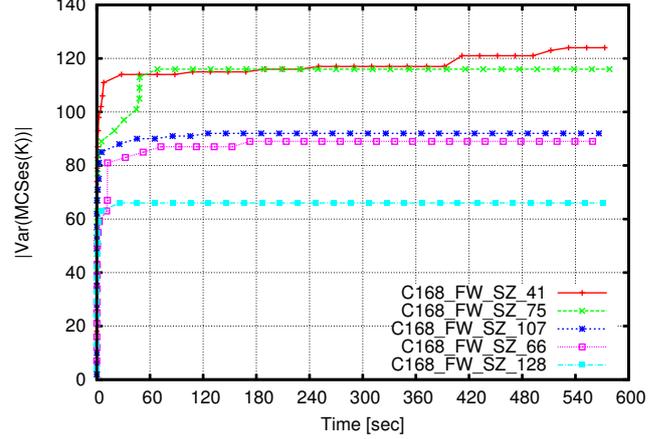
To demonstrate the feasibility of Algorithm 1, we implemented a prototype, called CAMUS\_IDMUS, by adapting the source code of CAMUS\_MCS 1.02<sup>7</sup>. All the tests were performed on a machine running Mac OS X 10.6.6, with 4G memory and 2.4G Intel Core 2 Duo CPU.

**Table 1.** Evaluation of CAMUS\_IDMUS on DC Benchmark

Instance	#V	#C	#M	#4	#Q	#VM	T
C168_FW_SZ_41	1,698	5,387	>30,104	1	211	> 124	600.00
C168_FW_SZ_66	1,698	5,401	>16,068	1	182	> 69	600.00
C168_FW_SZ_75	1,698	5,422	>37,317	1	198	> 116	600.00
C168_FW_SZ_107	1,698	6,599	>51,597	1	189	> 92	600.00
C168_FW_SZ_128	1,698	5,425	>25,397	1	211	> 66	600.00
C168_FW_UT_2463	1,909	7,489	>109,271	1	436	> 168	600.00
C168_FW_UT_2468	1,909	7,487	>54,845	1	436	> 138	600.00
C168_FW_UT_2469	1,909	7,500	>56,166	1	436	> 150	600.00
C168_FW_UT_714	1,909	7,487	>84,287	1	436	> 92	600.00
C168_FW_UT_851	1,909	7,491	30	1	436	11	0.35
C168_FW_UT_852	1,909	7,489	30	1	436	11	0.35
C168_FW_UT_854	1,909	7,486	30	1	436	11	0.35
C168_FW_UT_855	1,909	7,485	30	1	436	11	0.35
C170_FR_SZ_58	1,659	5,001	177	1	157	54	0.46
C170_FR_SZ_92	1,659	5,082	131	1	163	46	0.10
C170_FR_SZ_95	1,659	4,955	175	1	23	23	0.20
C170_FR_SZ_96	1,659	4,955	1,605	1	125	43	0.36

**Data Set** We use the DC benchmark from an automotive product configuration domain [23] that has been shown having a wide range of characteristics with respect to each instance’s MCSes and MUSes. There are 6 groups of test data in DC benchmark<sup>8</sup>. Due to space limitation, we only present results on the following two groups here: *SZ* (stability of the order completion process) and *UT* (superfluous parts). The results are shown in Table 1 with the following columns:

*instance*: name of the instance  $K$ ;  
*#V (#C)*: number of variables (clauses);  
*#M*: number of MCSes computed by CAMUS\_IDMUS;  
*#4 (#Q)*: number of contradictory variables in preferred 4-model (Q-model), i.e.,  $ID_4(K) \times |\text{Var}(K)|$  ( $ID_Q(K) \times |\text{Var}(K)|$ ), computed by the reduction to partial Max-SAT [24];  
*#VM*:  $|\text{Var}(\text{MUSes}(K))|$ , i.e.,  $ID_{MUS}(K) \times |\text{Var}(K)|$ , computed by CAMUS\_IDMUS;

<sup>7</sup> <http://www.eecs.umich.edu/~lifffiton/camus/><sup>8</sup> <http://www-sr.informatik.uni-tuebingen.de/~sinz/DC/>**Figure 1.** Anytime Property of CAMUS\_IDMUS

$T$ : running time of CAMUS\_IDMUS in seconds, with a timeout 600s.

To demonstrate the anytime property of CAMUS\_IDMUS, we also visualize some of the computing results in Figure 1. From Table 1 and Figure 1, we get the following interesting observations:

- For all the instances  $K$ , we have  $ID_4(K) = 1/|\text{Var}(K)|$  which only can indicate that some contradiction exists in the knowledge bases but can not help us distinguish the different amounts of contradictions in them.
- For many instances,  $ID_Q$  is much larger than  $ID_{MUS}$  (e.g. C168\_FW\_UT\_851—C170\_FR\_SZ\_92). This indicates that  $ID_Q$  often overestimates inconsistency than  $ID_{MUS}$ . In particular, C168\_FW\_UT\_2463 and C168\_FW\_UT\_851 have same  $ID_Q$  but far different  $ID_{MUS}$ , that is,  $ID_{MUS}$  can distinguish them well but  $ID_Q$  cannot.
- About half of the instances (C168\_FW\_UT\_851 — 855, C170\_FR\_SZ\_58 — 96) do not have many MCSes ( $< 2000$ ), so their  $ID_{MUS}$  can be computed very efficiently ( $< 0.5s$ ). Sometimes it is even much faster than computing  $ID_4$  and  $ID_Q$ , e.g., for  $K = \text{C170\_FR\_SZ\_96}$ ,  $ID_4(K)$  took 1.86s and  $ID_Q(K)$  took 47.1s by the approach in [24], while  $ID_{MUS}(K)$  just took 0.35s.<sup>9</sup>
- For the instances with many MCSes, although CAMUS\_IDMUS can not terminate before the timeout, as shown in Figure 1, we can get the approximated value of  $ID_{MUS}$  quickly. Moreover, after a short time of computation (60s for the example instances), the ranking of  $ID_{MUS}$  for the instances are almost stable.
- In particular, our evaluation shows that using the number of variables rather than the number of MUSes as inconsistency measure is often more practical, as there may be exponentially many MUSes, thus very hard to approximate; but the number of variables in MUSes is often limited or easily to be approximated.

## 7 Related Work

Inconsistency measurement is an active research field, and many measurements have been proposed. They can be roughly divided into syntax based approaches and semantics based ones. The former approaches are usually based on number of clauses in MUSes [14, 11,

<sup>9</sup> Due to space limitation, we did not list the running time for  $ID_4$  and  $ID_Q$  in Table 1.

22]. Whilst, the latter approaches are usually based on truth values in the “most classical” models under some multi-valued logics such as four-valued semantics [9], three-valued semantics [3],  $LP_m$  semantics [4], and quasi-classical semantics [8]. Our new approaches are based on the number of variables in the MUSes, and can be seen as a combination of the two approaches.

Shapley inconsistency measures are another attempt of combining two approaches, which allow us to see the distribution of the contradiction among different formulas in a KB [10]. Our measure  $ID_{MUS}$  is “orthogonal” to them: we can define  $ID_{MUS}$  Shapley inconsistency value to see how contradictions distribute in the KB under  $ID_{MUS}$ .

To make the inconsistency measurements practically useful, efficient algorithms are important, but there is not much work on it. Ma et al. attempted to develop an anytime algorithm for the four-valued semantics based inconsistency degree  $ID_4$  [19]. Later Xiao et al. developed reductions of  $ID_4$  and  $ID_Q$  to Max-SAT problems [24], so that existing powerful Max-SAT solvers can be used. To the best of our knowledge, there is no work concerning and implementing the computation of syntax based inconsistency measures, even though efficient MUS finders, e.g. CAMUS [16], and HYCAM [7] are available. One possible reason is that the number of MUSes for a given KB can be exponentially large, thus very hard to count or approximate. In contrast, this is possible for the inconsistency degrees proposed in this paper because the numbers of variables in MUSes are often limited or easier to be approximated in practice.

Grant and Hunter recently proposed a methodology about stepwise inconsistency resolution [5, 6]. Since values of semantics based measures are invariant in the splitting step, syntax based approaches are more suited in this framework. As our approach is the combination of syntax and semantics, the splitting operator can be used for resolving inconsistency under our measurements.

## 8 Conclusion and Perspectives

We proposed two new inconsistency measurements, named  $ID_{MUS}$  and  $ID_{MCS}$ , based on counting variables of MUSes and MCSes. We proved that they are equivalent to each other and they have preferred properties than existing multi-valued inconsistency degrees. Take the example given in the introduction,  $K$  and  $K'$  cannot be distinguished neither by purely syntax based measures ( $MIV_D$  and  $MIV_C$ ) or semantic based measures ( $ID_4$ ,  $ID_Q$ ). For the comparison of  $ID_{MUS}$  with  $ID_4$  and  $ID_Q$ , we discovered an interesting relationship between multi-valued logics and MUSes: for a given KB, the set of variables in its MUSes is the super set of the contradict variables in each preferred 4-model of KB, and is the subset of the contradict variables in each preferred Q-model of KB.

Our complexity analysis showed that all the  $ID_{MUS}$  and  $ID_{MCS}$  related problems are in the second layer of polynomial hierarchy, and thus theoretically harder than  $ID_4$  and  $ID_Q$ . However, the evaluation of our prototype CAMUS\_IDMUS on the DC benchmark showed that our anytime algorithm makes  $ID_{MUS}$  ( $ID_{MCS}$ ) or its approximations practically useful and efficient even for large knowledge bases and work better to distinguish inconsistent KBs.

In the future, we plan to develop more efficient algorithms for  $ID_{MUS}$  based on the relation between  $ID_4$  and  $ID_{MUS}$ , and try to avoid the generation of exponentially many MCSes when computing  $ID_{MUS}$ . Moreover, the relationship between multi-valued logics and MUSes/MCSes itself is also interesting which could be potentially useful for boosting MUSes (MCSes) finding algorithms.

## Acknowledgments

We thanks Prof. Thomas Eiter for his support, in particular, on the complexity results. This work was supported by the Austrian Science Fund (FWF) grants P20840, by the Quaero Programme (funded by OSEO), and by the DFG under grant BA 1122/16-1.

## REFERENCES

- [1] Ofer Arieli and Arnon Avron, ‘The value of the four values’, *Artificial Intelligence*, **102**, 97–141, (1998).
- [2] Philippe Besnard and Anthony Hunter, ‘Quasi-classical logic: Non-trivializable classical reasoning from inconsistent information’, in *Proc. of ECSQARU’95*, pp. 44–51, (1995).
- [3] John Grant, ‘Classifications for inconsistent theories’, *Notre Dame Journal of Formal Logic*, **19**(3), 435–444, (1978).
- [4] John Grant and Anthony Hunter, ‘Measuring inconsistency in knowledgebases’, *Journal of Intelligent Information Systems*, **27**(2), 159–184, (2006).
- [5] John Grant and Anthony Hunter, ‘Measuring consistency gain and information loss in stepwise inconsistency resolution’, in *ECSQARU*, volume 6717 of *LNCS*, pp. 362–373. Springer, (2011).
- [6] John Grant and Anthony Hunter, ‘Measuring the good and the bad in inconsistent information’, in *IJCAI*, ed., Toby Walsh, pp. 2632–2637. IJCAI/AAAI, (2011).
- [7] Éric Grégoire, Bertrand Mazure, and Cédric Piette, ‘Boosting a complete technique to find MSS and MUS thanks to a local search oracle’, in *IJCAI*, ed., Manuela M. Veloso, pp. 2300–2305, (2007).
- [8] Anthony Hunter, ‘Measuring inconsistency in knowledge via quasi-classical models’, in *Proc. of AAAI’02*, pp. 68–73, (2002).
- [9] Anthony Hunter, ‘How to act on inconsistent news: Ignore, resolve, or reject’, *Data Knowl. Eng.*, **57**(3), 221–239, (2006).
- [10] Anthony Hunter and S. Konieczny, ‘Shapley inconsistency values’, in *Proc. of KR’06*, pp. 249–259, (2006).
- [11] Anthony Hunter and S. Konieczny, ‘Measuring inconsistency through minimal inconsistent sets’, in *Proc. of KR’08*, pp. 358–366, (2008).
- [12] Anthony Hunter and Sébastien Konieczny, ‘Approaches to measuring inconsistent information.’, in *Inconsistency Tolerance*, pp. 191–236, (2005).
- [13] Birgit Jenner and Jacobo Torán, ‘Computing functions with parallel queries to NP’, *Theor. Comput. Sci.*, **141**(1&2), 175–193, (1995).
- [14] Kevin Knight, ‘Measuring inconsistency’, *Journal of Philosophical Logic*, **31**(1), 77–98, (2002).
- [15] Paolo Liberatore, ‘Redundancy in logic I: CNF propositional formulae’, *Artif. Intell.*, **163**(2), 203–232, (2005).
- [16] Mark H. Liffiton and Kareem A. Sakallah, ‘Algorithms for computing minimal unsatisfiable subsets of constraints’, *J. Autom. Reasoning*, **40**(1), 1–33, (2008).
- [17] Yue Ma and Laurent Audibert, ‘Détection des contradictions dans les annotations sémantiques’, in *Proc. of TALN’09*, pp. 609–618, (2009).
- [18] Yue Ma, Guilin Qi, and Pascal Hitzler, ‘Computing inconsistency measure based on paraconsistent semantics’, *J. Log. Comput.*, **21**(6), 1257–1281, (2011).
- [19] Yue Ma, Guilin Qi, Guohui Xiao, Pascal Hitzler, and Zuoquan Lin, ‘Computational complexity and anytime algorithm for inconsistency measurement’, *IJAI*, **4**(1), 3–21, (2010).
- [20] Kedian Mu, Zhi Jin, Ruqian Lu, and Weiru Liu, ‘Measuring inconsistency in requirements specifications.’, in *Proc. of ECSQARU’05*, pp. 440–451, (2005).
- [21] Kedian Mu, Weiru Liu, and Zhi Jin, ‘Measuring the blame of each formula for inconsistent prioritized knowledge bases’, *Journal of Logic and Computation*, (February 2011).
- [22] Kedian Mu, Weiru Liu, Zhi Jin, and David A. Bell, ‘A syntax-based approach to measuring the degree of inconsistency for belief bases’, *Int. J. Approx. Reasoning*, **52**(7), 978–999, (2011).
- [23] Carsten Sinz, Andreas Kaiser, and Wolfgang Kuchlin, ‘Formal methods for the validation of automotive product configuration data’, *AIEDAM*, **17**(1), 75–97, (2003).
- [24] Guohui Xiao, Zuoquan Lin, Yue Ma, and Guilin Qi, ‘Computing inconsistency measurements under multi-valued semantics by partial max-SAT solvers’, in *Proc. of KR’10*, pp. 340–349, (2010).