Computing Inconsistency Measurements under Multi-Valued Semantics by Partial Max-SAT Solvers

Guohui Xiao[1,2], **Zuoquan Lin**[1], **Yue Ma**[3] and **Guilin Qi**[4]
[1]Department of Information Science, Peking University, China
[2] Institute of Information Systems, Vienna University of Technology, Austria
[3] Laboratoire d'Informatique de l'université Paris-Nord, Université Paris Nord - CNRS, France
[4]School of Computer Science and Engineering, Southeast University, China
{xgh,lz}@is.pku.edu.cn, yue.ma@lipn.univ-paris13.fr, gqi@seu.edu.cn

Abstract

Measuring the inconsistency degree of a knowledge base can help us to deal with inconsistencies. Several inconsistency measures have been given under different multi-valued semantics, including 4-valued semantics, 3-valued semantics, LPm and Quasi Classical semantics. In this paper, we first carefully analyze the relationship between these inconsistency measures by showing that the inconsistency degrees under 4-valued semantics, 3-value semantics, LPm are the same, but different from the one based on Quasi Classical semantics. We then consider the computation of these inconsistency measures and show that computing inconsistency measurement under multi-valued semantics is usually intractable. To tackle this problem, we propose two novel algorithms that respectively encode the problems of computing inconsistency degrees under 4-valued semantics (3-valued semantics, LPm) and under Quasi Classical semantics into the partial Max-SAT problems. We implement these algorithms and do experiments on some benchmark data sets. The preliminary but encouraging experimental results show that our approach is efficient to handle large knowledge bases.

1. Introduction

Inconsistency handling is one of the traditional topics in the field of knowledge representation and reasoning. Recently, there is an increasing interest in quantifying inconsistency in a knowledge base (KB). This is because it is not fine-grained enough to simply say that two inconsistent KBs contain the same amount of inconsistency. Indeed, quantifying inconsistency provides useful context information to resolve inconsistency (Hunter 2002; Hunter and Konieczny 2005; 2006). First, we can compare the quality of different knowledge bases based on their inconsistency degrees, i.e., those with less inconsistency degrees should be preferred (Hunter 2002). Second, we can decide how to act on inconsistency (Hunter 2006), i.e. to ignore or to resolve it, by considering the inconsistency degree of a knowledge base. Measuring inconsistency has several applications, such as ranking ontologies in the Semantic Web (Zhou et al. 2009).

Different approaches to measuring inconsistency were developed based on different views of atomic inconsistency (Hunter and Konieczny 2005). Syntactic views put the inconsistency atomicity to formulas, such as taking maximal consistent subsets of formulas (Knight 2002) or minimal inconsistent sets (Hunter and Konieczny 2008). Semantic ones put the inconsistency atomicity to propositional variables, such as considering the conflicting propositional variables based on some kind of multi-valued model (Grant 1978; Hunter 2002; Hunter and Konieczny 2005; Grant and Hunter 2006; Ma et al. 2007; Grant and Hunter 2008; Zhou et al. 2009). There are also ways to combine these two approaches such as the computation of the responsibility/contribution of each formula to the overall inconsistency in the knowledge base (Hunter and Konieczny 2006).

In this paper, we focus on the semantics based inconsistency measures which belong to the latter category. In this category, several types of multi-valued semantics have been used, including Quasi Classical semantics (Hunter 2000), 3-valued semantics (Levesque 1984), LP_m semantics (Priest 1991), and 4-valued semantics (Belnap 1977), which produce in turn various inconsistency measures denoted by ID_Q, ID_3, ID_{LP_m} and ID_4 in this paper, respectively. While all of these inconsistency measures are proposed separately, there is no work formally discussing the relationship among them. In this paper, we will show that inconsistency degrees under 3-valued, 4-valued and LP_m are the same, which means that we only need to consider ID_4 and ID_Q in the future.

In our previous work (Ma et al. 2009), we have shown that given a propositional knowledge base K, computing inconsistency degrees under 4-valued semantics is usually intractable. We extend this complexity result to the inconsistency measurements under 3-valued, LP_m and Quasi Classical semantics and show that all of these problems are **NP**-hard thus intractable.

One way to tackle such a complex problem is to develop an algorithm with heuristic search and then apply pruning strategies. Following the principle of approximating logical reasoning (Schaerf and Cadoli 1995), (Ma et al. 2009) proposed an anytime algorithm to compute approximating inconsistency degrees of a propositional knowledge base. However, when the size of a knowledge base becomes large, the execution time of that algorithm may be still unacceptable such that further optimizations are required. In this paper, we take another direction which is to reduce the problem of measuring inconsistency to some existing problems with highly optimized solvers. Particularly, we propose two novel algorithms encoding the problems of computing inconsistency degrees to partial Max-SAT problems so that we take fully use of the power of the state of the art partial Max-SAT solvers. Our experiment results show that this approach is efficient to handle large knowledge bases, and outperforms our previous approximating algorithm (Ma et al. 2009).

The remainder of this paper is structured as follows: In Section 2, we recall several inconsistency measures and some satisfiability problems. Section 3 discusses the relationship among different inconsistency measures. The complexity results of inconsistency measures are shown in Section 4. In Section 5, we propose two novel algorithms encoding the problems of computing various inconsistency degrees to the partial Max-SAT problems. Section 6 describes the implementation and evaluation. We conclude this paper and outlook our future work in Section 7.

2. Preliminaries

In this paper, we consider a propositional language $\mathcal{L}_{\mathcal{A}}$ with a finite set of propositional variables $\mathcal{A} = \{p_1, \ldots, p_n\}$. A literal is a variable p or its negation $\neg p$. A knowledge base is a set of propositional formulas built from $\mathcal{A}$. *Var*(K) denotes the set of variables occurring in K and $|S|$ denotes the cardinality of a set S.

A clause $\gamma = l_1 \vee l_2 \vee \ldots \vee l_k$ is a disjunction of literals. A CNF formula is a conjunction of clauses, which is usually represented as a set of clauses $K = \{\gamma_1, \gamma_2, \ldots, \gamma_m\}$.

2.1 Inconsistency Measures by Multi-Valued Semantics

Several important inconsistency measures have been defined by multi-valued semantics, such as four-valued semantics (4-semantics) (Hunter 2006; Ma et al. 2009), three-valued semantics (3-semantics) (Grant 1978), LP_m semantics (Grant and Hunter 2006), and quasi-classical semantics (Q-semantics) (Hunter 2002). All of these multi-valued semantics use a third truth value B to stand for the contradictory information.

To distinguish from multi-valued semantics, we call the original two-valued semantics of propositional logic as the classical semantics throughout the paper. And we use $\models$ for the entailment under the classical semantics.

We provide a uniform definition of an inconsistency degree under these semantics as follows:

Definition 1. *Suppose I is a multi-valued interpretation under i-semantics ($i = 3, 4, LP_m, Q$). The* inconsistency degree of knowledge base K with respect to I*, denoted $ID_i(K, I)$, is a value in $[0, 1]$ defined as*

$$ID_i(K, I) \quad = \quad \frac{|\{p \in Var(K) \mid p^I = B\}|}{|Var(K)|},$$

where the numerator $\{p \in Var(K) \mid p^I = B\}$ is called the conflicting set of I with respect to K, written Conflict(K, I).

The inconsistency degree of K under i-semantics, denoted $ID_i(K)$, is defined as

$$ID_i(K) = \min_{I \models_i K} \{ID_i(K, I)\},$$

where $I \models_i K$ means that I is a model of K under i-semantics.

We give a brief introduction to each of these multi-valued semantics below.

Four-valued Semantics Compared to two truth values used by classical semantics, the set of truth values for 4-valued semantics (Belnap 1977; Arieli and Avron 1998) 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 $\mathsf{FOUR} = (\{t, f, B, N\}, \preceq)$: $f \preceq N \preceq t, f \preceq B \preceq t, N \not\preceq B, B \not\preceq N$.

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 $\neg N = N$.

The designated set of FOUR is $\{t, B\}$. So a 4-valued interpretation I is a *4-model* of a knowledge base K denoted $I \models_4 K$ if and only if for each formula $\phi \in K$, $\phi^I \in \{t, B\}$. A knowledge base which has a 4-model is called *4-valued satisfiable*. A knowledge base K 4-valued entails a formula φ, written $K \models_4 \varphi$, if and only if each 4-model of K is a 4-model of φ.

Example 1. *Given a propositional knowledge base $K = \{p, \neg p \vee q, \neg q \vee r, \neg r, s \vee u\}$. Consider three 4-valued models I_1, I_2 and I_3 of K defined as:*

$$p^{I_1} = t, q^{I_1} = B, r^{I_1} = f, s^{I_1} = t, u^{I_1} = N;$$
$$p^{I_2} = B, q^{I_2} = f, r^{I_2} = B, s^{I_2} = t, u^{I_2} = N;$$
$$p^{I_3} = B, q^{I_3} = B, r^{I_3} = B, s^{I_3} = t, u^{I_3} = N.$$

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

Quasi-Classical Semantics (Q-semantics) Let $\mathcal{A}^+_-$ be a set of objects defined as follows:

$$\mathcal{A}^+_- = \{+p, -p \mid p \in \mathcal{A}\}.$$

We call any $I \in \wp(\mathcal{A}^+_-)$ a Q-interpretation, where $\wp(\mathcal{A}^+_-)$ is the power set of $\mathcal{A}^+_-$. Let $l_1, \ldots, l_n$ be literals. The focus of $l_1 \vee \ldots \vee l_n$ by l_i, denoted by $\otimes(l_1 \vee \ldots \vee l_n, l_i)$, is defined as follows (Hunter 2000):

$$\otimes(l_1 \vee \ldots \vee l_n, l_i) = l_1 \vee \ldots \vee l_{i-1} \vee l_{i+1} \vee \ldots \vee l_n.$$

For a knowledge base K in CNF (Conjunctive Normal Form), Q-semantics is defined as follows:

Definition 2 (Q-models(Besnard and Hunter 1995)). *Suppose p is an atomic propositional variable, $\gamma_1, \ldots, \gamma_m$ are propositional formulas and $l_1, \ldots, l_n$ are literals. Let $\models_Q$ be a satisfiability relation such that $\models_Q \subseteq \wp(\mathcal{A}^+_-) \times \mathcal{L}_{\mathcal{A}}$.*

For $I \in \wp(\mathcal{A}^{+}_{-})$ we define $\models_Q$ in the following way:

$$\begin{array}{lll} I \models_Q p & \textit{iff} & +p \in I; \\ I \models_Q \neg p & \textit{iff} & -p \in I; \\ I \models_Q l_1 \vee \ldots \vee l_n & \textit{iff} & [I \models_Q l_1 \textit{ or } \ldots \textit{ or } I \models_Q l_n] \\ & & \textit{and } [\textit{for all } i, \textit{ if } I \not\models_Q \neg l_i \\ & & \textit{then } I \models_Q \otimes(l_1 \vee \ldots \vee l_n, l_i)]; \\ I \models_Q \{\gamma_1, ..., \gamma_m\} & \textit{iff} & I \models_Q \gamma_i (1 \leq i \leq m). \end{array}$$

For a knowledge base K in arbitrary form, its Q-models are defined as the Q-models of $CNF(K)$, where $CNF(K)$ is the set of clauses obtained by the classical transformation of K into CNF.

Similarly to Belnap's 4-valued logic, 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 given in Definition 1.

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

Example 2 (Example 1 Continued). $K = \{p, \neg p \vee q, \neg q \vee r, \neg r, s \vee u\}$. *Consider the following 4-models I_1 , I_2 and I_3 of K:*

$$p^{I_1} = t, q^{I_1} = B, r^{I_1} = f, s^{I_1} = t, u^{I_1} = N;$$
$$p^{I_2} = B, q^{I_2} = f, r^{I_2} = B, s^{I_2} = N, u^{I_2} = t;$$
$$p^{I_3} = B, q^{I_3} = B, r^{I_3} = B, s^{I_3} = t, u^{I_3} = N.$$

By definition 2, I_1 and I_2 are not Q-models of K, although they are 4-models of K. In fact, I_3 is a Q-model of K and we have $ID_Q(K) = ID_Q(K, I_3) = \frac{3}{5}$.

Three-Valued and LP_m Semantics A 3-interpretation I is a 4-interpretation with the restriction that for every $p \in \mathcal{A}$, $p^I \neq N$(Levesque 1984). Similarly we can define satisfiability relation $\models_3$.

An LP_m interpretation (Priest 1991) is a 3-valued interpretation with the restriction that only "most classical" 3-valued models are considered as LP_m models. Formally, $I \models_{LP_m} \alpha$ if and only if $I \models_3 \alpha$ and there does not exist any other 3-valued model J of α such that $\{p \mid p^J = B\} \subsetneq \{p \mid p^I = B\}$.

2.2 Satisfiability Problems

Deciding if a knowledge base in CNF is satisfiable is called a satisfiability (SAT) problem which is NP-complete. Even though the SAT problem is intractable, the state of the art SAT solvers are highly optimized and can deal with large size inputs.

As an extension of SAT, partial Max-SAT (the partial maximum satisfiability problem) has gotten deep study recently. Formally, a partial MaxSAT problem is of the form $P = (H, S)$, where H is a set of clauses, called the hard part; And S is the other set of clauses, called the soft part. The objective is to ask for a classical variable assignment that satisfies all hard clauses in H together with the maximum number of the soft ones in S. That is, an answer should be a two-valued interpretation $\hat{I}$ such that $|\{\gamma \mid \gamma \in S, \hat{I} \models \gamma, \hat{I} \models H\}| = \max_I |\{\gamma \mid \gamma \in S, I \models \gamma, I \models H\}|$.

The state of the art partial MaxSAT solvers such as SAT4j MaxSAT (Berre 2009), MSUnCore (Marques-Silva 2009) and Clone (Pipatsrisawat and Darwiche 2007) are highly optimized and scalable as shown in the third[1] and fourth[2] MaxSAT Evaluations. Moreover, they are free to download and to use for academic research purpose.

3. Relationship among Different Inconsistency Measures

This section analyzes the relationship among different inconsistency measures. It turns out that ID_3, ID_4 and ID_{LP_m} are the same for any given knowledge base, but different from ID_Q.

Proposition 1. *Let K be a propositional knowledge base. Then $ID_3(K) = ID_4(K)$.*

Proof. (1) We first prove that $ID_4(K) \leq ID_3(K)$:
If $I \models_3 K$, then $I \models_4 K$.

$$\begin{array}{rcl} ID_4(K) & = & min\{ID_4(K, I) \mid I \models_4 K\} \\ & \leq & min\{ID_4(K, I) \mid I \models_3 K\} \\ & = & min\{ID_3(K, I) \mid I \models_3 K\} \\ & = & ID_3(K). \end{array}$$

(2) Then we show that $ID_4(K) \geq ID_3(K)$:
Given a 4-interpretation I of K, we can define a 3-interpretation I' as follows,

$$p^{I'} = \begin{cases} p^I & \text{if } p^I \neq N \\ t & \text{if } p^I = N \end{cases}$$

It is easy to see that if $I \models_4 K$ then $I' \models_3 K$. Moreover, we have $\{p \mid p^I = B\} = \{p \mid p^{I'} = B\}$, which in turn means $ID_4(K, I) = ID_3(K, I')$. Therefore, by the definition of $ID_4(K)$ and $ID_3(K)$, we have $ID_4(K) \geq ID_3(K)$.

In all, $ID_4(K) = ID_3(K)$ holds. □

Example 3 (Example 2 Continued). $K = \{p, \neg p \vee q, \neg q \vee r, \neg r, s \vee u\}$. *Consider a 4-model I_1 of K defined as follows:*

$$p^{I_1} = t, q^{I_1} = B, r^{I_1} = f, s^{I_1} = t, u^{I_1} = N.$$

By changing u^{I_1} from N to t, we can get the following 3-model I'_1 of K:

$$p^{I'_1} = t, q^{I'_1} = B, r^{I'_1} = f, s^{I'_1} = t, u^{I'_1} = t.$$

Clearly, $ID_4(K, I_1) = ID_3(K, I'_1)$.

Proposition 2. $ID_{LP_m}(K) = ID_3(K)$.

[1] `http://www.maxsat.udl.cat/08/`
[2] `http://www.maxsat.udl.cat/09/`

Proof. (1) $ID_3(K) \leq ID_{LP_m}(K)$.

Since every LP_m-model is also a 3-model, $ID_3(K) \leq ID_{LP_m}(K)$ follows.

(2) $ID_3(K) \geq ID_{LP_m}(K)$.

Assume that $ID_3(K) < ID_{LP_m}(K)$, then there exist two 3-interpretation I_0, J_0, s.t. $I_0 \models_3 K$, $J_0 \models_{LP_m} K$, $ID_3(K, I_0) = ID_3(K)$ and $ID_{LP_m}(K, J_0) = ID_{LP_m}(K)$. So $ID_3(K, I_0) < ID_{LP_m}(K, J_0)$. Then we have $|\{p \mid p^{I_0} = B\}| < |\{p \mid p^{J_0} = B\}|$, and we know no other 3-model can have less conflicting set by the definition of I_0, so I_0 is an LP_m model of K. $ID_{LP_m}(K, I_0) < ID_{LP_m}(K, J_0)$ leads to a contradiction. □

Example 4. $K = \{p, \neg p \vee q, \neg q \vee r, \neg r, s \vee u\}$. *Consider again the 3-model I_1' defined in Example 3. We can see that I_1' is an LP_m model of K because $I_1' \models_3 K$ and any other 3-models J satisfying $\{p \mid p^J = B\} \subseteq \{p \mid p^{I_1'} = B\}$ can only be a classical interpretation which in turn cannot not satisfy K. So $ID_{LP_m}(K) = \min_{I \models_{LP_m} K} ID_{LP_m}(K, I) = ID_{LP_m}(K, I_1') = 1/5 = ID_3(K)$.*

Proposition 3. $ID_4(K) \leq ID_Q(K)$.

Proof. Since every Q-model of K is also a 4-model of K, the conclusion is obvious. □

In example 2, we have seen that $ID_4(K) = \frac{1}{5} < \frac{3}{5} = ID_Q(K)$. This shows that $ID_4(K)$ can be strictly less than $ID_Q(K)$.

In summary, by Propositions 1, 2, and 3, we have the following theorem.

Theorem 4. $ID_3(K) = ID_{LP_m}(K) = ID_4(K) \leq ID_Q(K)$.

4. Computational Complexities

Apart from any particular algorithm, let us study the computational complexity of the inconsistency degree to see how hard the problem itself is. In (Ma et al. 2009), the complexity results of problems related to ID_4 have been discussed. These results can be extended to other measurements parallel as shown below.

We first define the following computation problems related to inconsistency degrees under i-semantics ($i = 3, 4, LPm, Q$):

- ID$_{i,\leq d}$ (resp. ID$_{i,<d}$, ID$_{i,\geq d}$,ID$_{i,>d}$): Given a propositional knowledge base K and a number $d \in [0, 1]$, is *ID*$_i(K) \leq d$ (resp. *ID*$_i(K) < d$, *ID*$_i(K) \geq d$, *ID*$_i(K) > d$)?
- EXACT-ID$_i$: Given a propositional knowledge base K and a number $d \in [0, 1]$, is *ID*$_i(K) = d$?
- ID$_i$: Given a propositional knowledge base K, what is the value of *ID*$_i(K)$?

Obviously, we have two trivial instances ID$_{i,\leq 1}$ and ID$_{i,\geq 0}$ with answer "yes" and another two trivial instances ID$_{i,<0}$ and ID$_{i,>1}$ with answer "no".

In general cases, the complexity of these computational problems is shown by following theorems.

Theorem 5. *ID$_{i,\leq d}$ and ID$_{i,<d}$ ($i = 3, 4, LP_m, Q$) are* **NP**-*complete ; ID$_{i,\geq d}$ and ID$_{i,>d}$ ($i = 3, 4, LP_m, Q$) are* **coNP**-*complete.*

Proof. We prove these results separately as follows:

ID$_{i,\leq d}$ is NP-complete:

The membership of ID$_{i,\leq d}$ ($i = 3, 4, Q$) in NP is achieved by the following non-deterministic algorithm:

1. Guess an i-interpretation I over *Var*(K);
2. Check that I is an i-model of K and $\frac{|\mathit{Conflict}(I|}{|\mathit{Var}(K)|} \leq d$, which can be done in deterministic polynomial time.

ID$_{LP_m,\leq d}$ is in NP follows from $ID_{LP_m}(K) = ID_4(K)$ by Theorem 4.

The NP-hardness comes from the following reduction from checking the satisfiability of K under classical 2-valued semantics, which is known to NP-complete, to this problem. The reduction is that K is 2-valued satisfiable if and only if *ID*$_i(K) \leq 0$ which is obvious by the definition of inconsistency degree.

ID$_{i,<d}$ is NP-complete:

Similarly to the case of ID$_{i,\leq d}$, the membership in **NP** holds obviously. The **NP**-hardness holds by the reduction that K is 2-valued satisfiable if and only if *ID*$_i(K) < \frac{1}{2|\mathit{Var}|}$. This is because, by the definition of *ID*$_i(K)$, the smallest value of *ID*$_i(K)$ for an inconsistent knowledge base is $\frac{1}{|\mathit{Var}|}$.

ID$_{i,\geq d}$ and ID$_{i,>d}$ are coNP-complete:

This is because ID$_{i,\geq d}$ is the complementary problem of ID$_{i,<d}$ and ID$_{i,>d}$ is the complementary problem of ID$_{i,\leq d}$. □

Theorem 6. *EXACT-ID$_i$ ($i = 3, 4, LP_m, Q$) is* **DP**-*complete* [3].

Proof. To show that it is in **DP**, we have to exhibit two languages $L_1 \in$ **NP** and $L_2 \in$ **coNP** such that the set of all "yes" instances of EXACT-ID$_i$ is $L_1 \cap L_2$. This is easy by setting $L_1 = \{K \mid ID_i(K) \leq d\}$ and $L_2 = \{K \mid ID_i(K) \geq d\}$.

To show the completeness, let $L = L_1 \cap L_2$ be any language in **DP**. We have to show that L can be reduced to EXACT-ID$_i$. To this end, recall that ID$_{i,\leq}$ is **NP**-complete and ID$_i, \geq$ is co**NP**-complete, that is, there is a reduction R_1 from L_1 to ID$_i, \leq$ and a reduction R_2 from L_2 to ID$_i, \geq$. Therefore, the reduction R from L to EXACT-ID$_i$ can be defined as follows, for any input x: $R(x) = (R_1(x), R_2(x))$. We have that $R(x)$ is a "yes" instance of EXACT-ID$_i$ if and only if $R_1(x)$ is a "yes" instance of ID$_{i,\leq}$ and $R_2(x)$ is a "yes" instance of ID$_{i,\geq}$, which is equal to $x \in L$. □

Due to the fact that $ID_4(K) = ID_3(K) = ID_{LP_m}(K)$, the complexity result of ID$_4$ (Ma et al. 2009) can be extended as the following theorem.

[3]A language L is in the class **DP** (Papadimitriou 1994) iff there are two languages $L_1 \in$ **NP** and $L_2 \in$ **coNP** such that $L = L_1 \cap L_2$.

Theorem 7. *ID_i(i=3, 4, LPm) is $\mathbf{FP}^{\mathbf{NP}[\log \mathbf{n}]}$-complete*[4].

However, the functional complexity of ID_Q is still an open problem. In spite of this, because of the DP-completeness result of Exact-ID_Q 6, we can conclude that the problem of computing ID_Q is intractable.

5. Encoding Algorithms

In previous section, we have shown that computing inconsistency degrees is an intractable task generally. In this section, we propose two novel algorithms which encode the problem of computing inconsistency degrees to the partial Max-SAT problem, so that we can take full advantage of the state of the art partial Max-SAT solvers.

In this section, without loss of generality, we assume that all the KBs are given in CNF, i.e. a set of clauses, because any knowledge base can be transformed to a CNF in polynomial time while preserving satisfiability. By Theorem 4, we only need to consider the computations of ID_4 and ID_Q.

5.1 Computing Inconsistency Degree under 4-valued Semantics

Given a knowledge base $K = \{\gamma_i \mid i = 1, \ldots, n\}$ over variables set $\mathcal{A}$, it is well-known that the 4-valued reasoning on K can be simulated by the 2-valued reasoning on $4(K)$, where $4(\cdot)$ is the transformation function from (a set of) clauses to (a set of) clauses defined as follows (Cadoli and Schaerf 1996):

$$\begin{aligned} 4(\{\gamma_1, \gamma_2, \ldots, \gamma_n\}) &= \{4(\gamma_1), 4(\gamma_2), \ldots, 4(\gamma_n)\} \,; \\ 4(l_1 \vee \ldots \vee l_k) &= 4(l_1) \vee \ldots \vee 4(l_k) \,; \\ 4(p) &= +p \,; \\ 4(\neg p) &= -p \,. \end{aligned}$$

That is, $4(K)$ is a knowledge base over variables $\mathcal{A}_-^+ = \{+p, -p \mid p \in Var(K)\}$. Obviously, computing $4(K)$ from K can be done in linear time.

A 4-valued interpretation I on $\mathcal{A}$ can also be seen as a 2-valued interpretation on variables $\mathcal{A}_-^+$. The corresponding relation can be described as follows:

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

In the rest, we will refer to either of these two views without explicit explanation.

Theorem 8. *(Cadoli and Schaerf 1996) Given a propositional knowledge base K and a 4-valued interpretation I, we have $I \models_4 K$ iff $I \models 4(K)$.*

[4]Complexity $\mathbf{P}^{\mathbf{NP}[\log \mathbf{n}]}$ is defined to be the class of all languages decided by a polynomial-time oracle machine which on input x asks a total of $\mathcal{O}(log\,|x|)$ SAT (or any other problem in $\mathbf{NP}$) queries. $\mathbf{FP}^{\mathbf{NP}[\log \mathbf{n}]}$ is the corresponding class of functions.

Example 5. *Let $K = \{\neg p, p \vee q, \neg q, r\}$. We have $4(K) = \{-p, +p \vee +q, -q, +r\}$. Consider the interpretation $I_1 = \{+p, -p, -q, +r\}$. I_1 can be seen as a 4-interpretation on $\{p, q, r\}$ with $p^{I_1} = B, q^{I_1} = f, r^{I_1} = t$. I_1 can also be viewed as a 2-interpretation on $\{+p, -p, +q, -q, +r, -r\}$ which assigns variables in I_1 true and other variables false, i.e. in the following way:*

$$\begin{aligned} &+p^{I_1} = t, -p^{I_1} = t, +q^{I_1} = f, \\ &-q^{I_1} = t, +r^{I_1} = t, -r^{I_1} = f. \end{aligned}$$

It is easy to check that $I_1 \models_4 K$ and $I_1 \models 4(K)$.

Corollary 9. *Given a knowledge base K over $\mathcal{A}$, the inconsistency degree of K under 4-valued semantics can be computed by 2-valued semantics over $\mathcal{A}_-^+$:*

$$\begin{aligned} ID_4(K, I) &= \frac{|b(K, I)|}{|Var(K)|} \,; \\ ID_4(K) &= \min_{I \models 4(K)} ID_4(K, I) = \frac{\min_{I \models 4(K)} |b(K, I)|}{|Var(K)|} \,. \end{aligned}$$

where $b(K, I) = \{p \in Var(K) \mid +p^I = t \text{ and } -p^I = t\}$.

Proof. By Definition 1 and the fact that $p^I = B$ iff $+p^I = t$ and $-p^I = t$, this corollary holds obviously. □

Based on Corollary 9, next we study an encoding algorithm that reduces the computation of four-value semantics based inconsistency degree to a partial Max-SAT instance. First of all, note that

$$\begin{aligned} &\min_{I \models 4(K)} |\{p \mid p \in Var(K), +p^I = t \text{ and } -p^I = t\}| \\ = &\min_{I \models 4(K)} |\{p \mid p \in Var(K), (\neg + p \vee \neg - p)^I = f\}| \\ = &\max_{I \models 4(K)} |\{p \mid p \in Var(K), (\neg + p \vee \neg - p)^I = t\}|. \end{aligned}$$

This motivates us to use partial Max-SAT problem solvers to compute ID_4 by considering the following partial Max-SAT instance:

Definition 3. *Given a propositional knowledge base $K = \{\gamma_1, \ldots, \gamma_n\}$, $Var(K) = \{p_1, \ldots, p_m\}$, the corresponding partial Max-SAT problem for the 4-semantics based inconsistency degree ID_4, written $P_4(K) = (H_4(K), S_4(K))$, is defined as follows:*

$$\begin{aligned} H_4(K) &= \{4(\gamma) \mid \gamma \in K\}; \\ S_4(K) &= \{\neg +p \vee \neg -p \mid p \in Var(K)\}. \end{aligned}$$

Then we have the following theorem.

Theorem 10. *Suppose I is a solution to the partial Max-SAT problem $P_4(K)$. Let $b(I, K) = |\{p \in Var(K) \mid +p^I = t, -p^I = t\}|$ and $m(K) = |Var(K)|$. Then we have that $ID_4(K) = b(I, K)/m(K)$.*

Proof. By the definition of $P_4(K)$, I satisfies that for any other J, $b(I, K) \leq b(J, K)$. By Corollary 9, this theorem holds. □

Theorem 10 can be described by the following algorithm. The algorithm first generates $P_4(K)$ in line 4 to line 9, then computes a solution of $P_4(K)$ by calling a partial Max-SAT solver in line 10, and computes the value of inconsistency degree by theorem 10 in line 11 to 12.

Algorithm 1 Computing ID_4 by Partial Max-SAT Solver

1: **procedure** $ID_4(K)$
2: $\quad P \leftarrow \{\}$
3: $\quad m \leftarrow |\mathit{Var}(K)|$
4: $\quad$ **for all** Clause $\gamma \in K$ **do**
5: $\qquad P.addHardClause(4(\gamma))$
6: $\quad$ **end for**
7: $\quad$ **for all** Variable $p \in \mathit{Var}(K)$ **do**
8: $\qquad P.addSoftClause(\neg + p \vee \neg - p)$
9: $\quad$ **end for**
10: $\quad I \leftarrow$ PartialMaxSATSolver(P)
11: $\quad b = |\{p \mid +p^I = t \wedge -p^I = t\}|$
12: $\quad$ **return** b/m
13: **end procedure**

Corollary 11 (Correctness of Algorithm 1). *For any given knowledge base K, Algorithm 1 is sound and complete for computing the four-value based inconsistency degree of K. That is, $Algorithm1(K) = ID_4(K)$, where $Algorithm1(K)$ is the value returned by Algorithm 1 with K as the input.*

Proof. This conclusion easily follows from Theorem 10. □

Next example gives a further illustration of Algorithm 1.

Example 6. *Let $K = \{p \vee q, \neg p, \neg q, r\}$. We have $4(K) = \{+p \vee +q, -p, -q, +r\}$. Then, by Definition 3, the hard clause set of $P_4(K)$ is $\{+p \vee +q, -p, -q, +r\}$, and the soft clause set is*

$$P_4(K) = \{\neg + p \vee \neg - p, \neg + q \vee \neg - q, \neg + r \vee \neg - r\}.$$

For $P_4(K)$, we have the following one optimized solution I_0 by a partial Max-SAT solver:

$$+\,p^{I_0} = t, -p^{I_0} = t, +q^{I_0} = f,$$
$$-\,q^{I_0} = t, +r^{I_0} = t, -r^{I_0} = f.$$

The corresponding 4-model of K is $p^{I_0} = B, q^{I_0} = f, r^{I_0} = t$, from which we have that $ID_4(K) = 1/3$ by Algorithm 1, coinciding with its theoretical value.

5.2 Computing Inconsistency Degree under QC Semantics

Since QC-semantics based inconsistency degree is different from that based on four-value semantics as discussed in Section 3.In this section, we study an algorithm for computing QC-based inconsistency degree.

Firstly, similar with 4-valued semantics, we have that reasoning under QC semantics can be reduced to 2-valued logic.

To simplify notations, for every literal l, we denote:

$$\begin{array}{llll} +l = +p & \text{if } l = p, & +l = -p & \text{if } l = \neg p, \\ -l = -p & \text{if } l = p, & -l = +p & \text{if } l = \neg p. \end{array}$$

Definition 4 (QC Transformation). *(Marquis and Porquet 2001) Given a knowledge base $K = \{\gamma_1, \ldots, \gamma_n\}$ in CNF. , the QC transformation of K is defined as follows,*

$$\begin{aligned} Q(\{\gamma_1, \ldots, \gamma_n\}) &= \{Q(\gamma_1), \ldots, Q(\gamma_n)\}, \\ Q(l_1 \vee \ldots \vee l_n) &= \bigvee_{i=1}^{n} (+l_i \wedge \neg - l_i) \vee \bigwedge_{i=1}^{n} (+l_i \wedge -l_i). \end{aligned}$$

Theorem 12. *(Marquis and Porquet 2001) Let K be a knowledge base and I be a QC interpretation. Then*

$$I \models_Q K \textit{ iff } I \models Q(K).$$

Example 7. *Let $K = \{\neg p, p \vee q, \neg q, r\}$. Then we have $4(K) = \{-p, +p \vee +q, -q, +r\}$, but $Q(K) = \{-p, (+p \wedge \neg - p) \vee (q \wedge \neg - q) \vee (+p \wedge -p \wedge +q \wedge -q), -q, +r\}$, where means that $4(K)$ is not the same as $Q(K)$ in general.*

Now we can compute ID_Q by classical semantics according to the following corollary. Its proof is similar to that of Corollary 9.

Corollary 13. *Given a knowledge base K, the inconsistency degree of K over the variable set $\mathcal{A}$ under Q-semantics can be computed by the 2-valued semantics over the variables set $\mathcal{A}^+_-$:*

$$\begin{aligned} ID_Q(K, I) &= \frac{|\{p \in Var(K) \mid +p^I = t \wedge -p^I = t\}|}{|Var(K)|}; \\ ID_Q(K) &= \min_{I \models Q(K)} ID_Q(K, I)\,. \end{aligned}$$

Compared with $4(\cdot)$, the transformation function $Q(\cdot)$ can not maintain CNF. Thus $Q(l_1 \vee \ldots \vee l_n)$ can not be directly used in a partial Max-SAT solver in general. Besides, direct transformation of $Q(l_1 \vee \ldots \vee l_n)$ into CNF by distribution laws can give a formula of exponential size. To avoid this problem, we adopt a technique given in(Baaz, Egly, and Leitsch 2001) that introduces new variables in the transformation to preserve equisatisfiability under 2-valued semantics in the following way:

$$\begin{aligned} y_i &:= +l_i \wedge \neg - l_i, i = 1, \ldots, n\,; \\ z &:= \wedge_{i=1}^{n} (+l_i \wedge -l_i)\,. \end{aligned}$$

Subsequently, we define the transformation function $Q'(\cdot)$:

$$\begin{aligned} Q'(\{\gamma_1, \ldots, \gamma_n\}) = &\ \{Q'(\gamma_1), \ldots, Q'(\gamma_n)\} \\ Q'(l_1 \vee \ldots \vee l_n) = &\ (\bigvee_{i=1}^{n} y_i \vee z) \wedge \bigwedge_{i=1}^{n} (\neg y_i \vee +l_i) \\ &\wedge \bigwedge_{i=1}^{n} (\neg y_i \vee \neg - l_i) \\ &\wedge \bigwedge_{i=1}^{n} (\neg z \vee +l_i) \wedge \bigwedge_{i=1}^{n} (\neg z \vee -l_i). \end{aligned}$$

Obviously, each clause of length n is transformed to $4n+1$ clauses by $Q'(\cdot)$. It is easy to check that $Q'(p) \equiv +p$ and $Q'(\neg p) \equiv -p$.

By the following proposition, we can see that the computation of ID_Q can be simulated by 2-valued logic.

Proposition 14. *For any knowledge base K, we have*

$$ID_Q(K) = \frac{\min\limits_{I \models Q'(K)} |\{p \in Var(K) \mid +p^I = t, -p^I = t\}|}{|Var(K)|}.$$

Proof. Given an interpretation I on variables $\{+p, -p \mid p \in Var(K)\}$, s.t. $I \models Q(K)$, we can extend I to I' on variables $\{+p, -p \mid p \in Var(K)\} \cup \{y_i\} \cup \{z\}$ s.t. $I' \models Q'(K)$ by

$$\begin{aligned} y_i^{I'} &= (+l_i \wedge \neg - l_i)^I, i = 1, \ldots, n\,; \\ z^{I'} &= (\wedge_{i=1}^n (+l_i \wedge -l_i))^I\,. \end{aligned}$$

On the other hand, if $J \models Q'(K)$, then J can also be viewed as an interpretation for $Q(K)$ and $J \models Q(K)$.

So $\{p \mid p \in Var(K), +p^I = t, -p^I = t, I \models Q'(K)\} = \{p \mid p \in Var(K), +p^I = t, -p^I = t, I \models Q(K)\}$.

Then by corollary 13, the conclusion follows. ☐

Definition 5. *Given a propositional knowledge base $K = \{\gamma_1, \ldots, \gamma_n\}$, the corresponding partial Max-SAT problem $P_Q(K) = (H_Q(K), S_Q(K))$ for ID_Q is defined as follows:*

$$\begin{aligned} H_Q(K) =& \{Q'(\gamma) \mid \gamma \in K\}; \\ S_Q(K) =& \{\neg +p \vee \neg -p \mid p \in \mathit{Var}(K)\}. \end{aligned}$$

Similar to Theorem 10, we have the following theorem holds which gives a reduction from the computation of Q-semantics based inconsistency degree to the partial Max-SAT problem.

Theorem 15. *Given a knowledge base K, suppose I is a solution to the partial Max-SAT problem $P_Q(K)$. Let $b(I, K) = |\{p \mid +p^I = t \wedge -p^I = t\}|$, $m(K) = |\mathit{Var}(K)|$. Then $ID_Q(K) = b(I, K)/m(K)$.*

Example 8. *Let $K = \{\neg p, p \vee q, \neg q, r\}$. Then the hard part of $P_Q(K)$ is $Q'(K) = \{Q'(\neg p), Q'(p \vee q), Q'(\neg q), Q'(r)\}$, where $Q'(\neg p) = -p, Q'(\neg q) = -q, Q'(r) = +r$, and*

$$\begin{aligned} Q'(p \vee q) =& (y_p \vee y_q \vee z) \wedge (\neg y_p \vee +p) \wedge (\neg y_p \vee \neg - p) \\ & \wedge (\neg y_q \vee +q) \wedge (\neg y_q \vee \neg - q) \wedge (\neg z \vee +p) \\ & \wedge (\neg z \vee -p) \wedge (\neg z \vee +q) \wedge (\neg z \vee -q). \end{aligned}$$

The soft part of $P_Q(K)$ is $\{\neg + p \vee \neg - p, \neg + q \vee \neg - q, \neg + r \vee \neg - r\}$. One solution to $P_Q(K)$ is I_0 such that $+p^{I_0} = t, -p^{I_0} = t, +q^{I_0} = t, -q^{I_0} = t, +r^{I_0} = t, -r^{I_0} = f, {y_p}^{I_0} = f, {y_q}^{I_0} = f, z^{I_0} = t$. So $ID_Q(K) = \frac{2}{3}$ by Theorem 15.

Theorem 15 motivates the following algorithm. The propositional variables y_i and z are introduced by the transformation function $Q'(\cdot)$.

Algorithm 2 Computing ID_Q by Partial Max-SAT Solver

```
procedure ID_Q(K)
    P ← {}
    m ← |Var(K)|
    for all Clause γ = {l_1, ..., l_n} ∈ K do
        P.addHardClause(y_1 ∨ ... ∨ y_n ∨ z)
        for i = 1 to n do
            P.addHardClause(¬y_i ∨ +l_i)
            P.addHardClause(¬y_i ∨ ¬ − l_i)
            P.addHardClause(¬z ∨ +l_i)
            P.addHardClause(¬z ∨ −l_i)
        end for
    end for
    for all p ∈ Var(K) do
        P.addSoftClause(¬ +p ∨ ¬ −p)
    end for
    I ← PartialMaxSATSolver(P)
    b = |{p | +p^I = t ∧ −p^I = t}|
    return b/m
end procedure
```

Corollary 16 (Correctness of Algorithm 2). *For any given knowledge base K, Algorithm 2 is sound and complete for computing the QC-based inconsistency degree of K. That is, $Algorithm2(K) = ID_Q(K)$, where $Algorithm2(K)$ is the value returned by Algorithm 2 with K as the input.*

Proof. This conclusion easily follows from Theorem 15. ☐

6. Experimental Evaluation

This section describes the experimental results to show the efficiency of our encoding algorithms. To this end, we used three state of the art partial Max-SAT solvers, namely SAT4j MaxSAT (Berre 2009), MsUncore (Marques-Silva 2009) and Clone (Pipatsrisawat and Darwiche 2007), to implement our encoding algorithms.

The experiments were performed on an Intel Pentium 4 (3.00GHz) machine with 1G Memory running OpenSuse and the results were shown in Tables 1, 2 and 3. Both the program and test data can be found online [5]. We ran every instance against each solver with a timeout of 240 seconds and used "*" to indicate the occurrence of a timeout. The meaning of each column of these tables is given as follows:

- "name": the name of the instance used as test datum;
- "#V" and "#C": the number of variables and clauses in the instance;
- "ID_4" and "ID_Q": the values of inconsistency degrees under 4-semantics and Q-semantics, respectively;
- "AnyTime": the final time in seconds that produces the exact value by the any time algorithm in (Ma et al. 2009);
- "Encoding Algorithm": time consumed in seconds by encoding algorithms based on each partial Max-SAT solver.

[5] http://www.is.pku.edu.cn/~xgh/id/

Table 1 shows the comparison of Algorithm 1 with the anytime algorithm proposed in (Ma et al. 2009). The anytime algorithm (Ma et al. 2009) computes upper and lower bounds of inconsistency degrees (under four-valued semantics) by polynomial times invoking of a polynomial procedure that decides the satisfiability of a set of CNFs in restricted forms. The approximating inconsistency degrees are shown converging to exact inconsistency degrees as more and more computing resource is available. Please refer to (Ma et al. 2009) for more details. For the comparison, the data set we test is the same as that used in (Ma et al. 2009), that is, inputs are $K_N = \{p_i, q_j, \neg p_i \vee \neg q_j \mid 1 \leq i, j \leq N\}$ for $N = 1, 2, 5, 7, 10, 20, 50, 100$. Obviously, $|Var(K_N)| = 2N$ and $|K_N| = N^2 + 2N$. From Table 1, we can see that our encoding algorithm outperforms the anytime algorithm in (Ma et al. 2009) when $N > 10$. Furthermore, the anytime algorithm cannot deal with the inputs with $N > 20$, whilst our encoding algorithm can handle them easily. Note that the anytime algorithm cannot handle even one instance in the data sets used to test our encoding algorithms given in Tables 2 and 3. This shows the advantage of taking existing optimized partial Max-SAT solvers.

Instance				AnyTime	Encoding Algorithm		
name	#V	#C	ID_4		sat4j	msuncore	clone
001.cnf	2	3	0.500	0.001	0.351	0.016	0.566
002.cnf	4	8	0.500	0.003	0.351	0.016	0.571
005.cnf	10	35	0.500	0.268	0.365	0.016	0.635
007.cnf	14	63	0.500	4.477	0.360	0.017	0.732
010.cnf	20	120	0.500	228.754	0.353	0.018	0.960
020.cnf	40	440	0.500	*	0.457	0.031	1.396
050.cnf	100	2600	0.500	*	0.858	0.188	4.209
100.cnf	200	10200	0.500	*	3.513	1.570	17.876

Table 1: Comparison of AnyTime and Encoding Algorithm

Table 2 gives the results of Algorithm 1 performing on two groups of data set. One group (group A), with the prefix "uuf" of each instance, is obtained from the SAT benchmark SATLIB [6]. The other group (group B), with the prefix "C", is a large set of unsatisfiable CNF benchmarks from automotive product configuration (Sinz, Kaiser, and Küchlin 2003), each of which encodes a set of available configurations for a product, along with constraints enforcing a specific property to be checked. Due to space limitations, only part of the results in groups A and B are shown. Observed from Table 2, we can see that nearly all the instances can be handled by the implementation based on any partial Max-SAT solver, except uuf100-0103 and C168_FW_SZ_107 which cannot be handled by that based on Clone before timeout.

Table 3 describes the computation of ID_Q by the encoding algorithm (Algorithm 2) on the same data sets as those used in Table 2. We can see that implementation based on SAT4j can handle all the instances of group A in less than 1 second and handle all the data of group B in 9 seconds to 14 seconds; the implementation based on MsUncore cannot handle even one instance; the implementation based on Clone can deal with all the instances in less than 2 seconds.

[6] http://www.satlib.org

Instance				Encoding Algorithm		
name	#V	#C	ID_4	sat4j	msuncore	clone
uuf50-0101	50	218	0.02000	0.396	0.026	1.119
uuf50-0102	50	218	0.02000	0.398	0.020	1.121
uuf50-0103	50	218	0.02000	0.450	0.044	1.142
uuf50-0104	50	218	0.02000	0.397	0.027	1.279
uuf75-011	75	325	0.01330	0.496	0.031	1.379
uuf75-012	75	325	0.01330	0.447	0.030	1.355
uuf75-013	75	325	0.01330	0.443	0.033	1.333
uuf75-014	75	325	0.01333	0.494	0.029	1.372
uuf100-0101	100	430	0.01000	0.545	0.045	1.748
uuf100-0102	100	430	0.01000	0.918	0.053	2.088
uuf100-0103	100	430	0.02000	3.951	2.592	*
C168_FW_SZ_107	1698	5401	0.00059	0.698	0.120	*
C168_FW_SZ_128	1698	5422	0.00059	0.601	0.090	13.191
C168_FW_SZ_41	1698	7489	0.00059	0.849	0.085	11.939

Table 2: Computing ID_4 by Encoding Algorithm

Instance				Encoding Algorithm		
name	#V	#C	ID_Q	sat4j	msuncore	clone
uuf50-0101	50	218	1.000	0.445	*	0.428
uuf50-0102	50	218	1.000	0.444	*	0.446
uuf50-0103	50	218	1.000	0.449	*	0.246
uuf50-0104	50	218	1.000	0.494	*	0.433
uuf75-011	75	325	1.000	0.544	*	0.434
uuf75-012	75	325	1.000	0.548	*	0.435
uuf75-013	75	325	1.000	0.455	*	1.338
uuf75-014	75	325	1.000	0.646	*	0.437
uuf100-0101	100	430	1.000	0.709	*	0.478
uuf100-0102	100	430	1.000	0.803	*	0.438
uuf100-0103	100	430	1.000	0.749	*	0.445
C168_FW_SZ_107	1698	5401	0.124	9.269	*	1.487
C168_FW_SZ_128	1698	5422	0.107	9.916	*	0.792
C168_FW_SZ_41	1698	7489	0.117	13.627	*	0.738

Table 3: Computing ID_Q by Encoding Algorithm

From all of the tests given above, we can get the following conclusions for tested data sets:

- For most of these large sized instances, our algorithms can terminate in short time, which indicates the efficiency of our approach.
- The performance of the implementation of each of our algorithms relies heavily on the underlying partial Max-SAT solver. For example, in our experiment, the implementation based on MsUnCore is the fastest solver that can handle all the instances for ID4. In constrast, the implementation based on Clone performs best for most of the instances for IDQ. Compared with other solvers, SAT4j based implementation can handle all the instances for both ID_4 and ID_Q

One observation is that the values of ID_4 and ID_Q are usually different. Which measurement is more useful depends on the concrete context and the application. In our experiment, we found that the computation of ID_4 ran faster than that of ID_Q in most cases. This can be explained by the more complex transformation function $Q'(\cdot)$ used by Algorithm 2 than $4(\cdot)$ used by Algorithm 1.

7. Conclusion and Future Work

Several inconsistency measures under different multi-valued semantics, including 4-valued semantics, 3-valued semantics, LPm and Quasi Classical semantics were proposed in the literature. In this paper, we first carefully analyzed the relationship among all of these different inconsistency measures. We showed that the inconsistency measures under 4-valued semantics, 3-value semantics, and LPm are the same. Moreover, the inconsistency degree of an arbitrary inconsistent knowledge base under these semantics is less than or equal to that under Quasi Classical semantics.

The complexity analysis showed that the computation of the inconsistency degrees is a hard task. In order to use these inconsistency degrees in practice, an efficient algorithm for the computation of the inconsistency measures is essential. To tackle this problem, in this paper, we made some effort to explore a linear encoding of the computation of inconsistency degrees to the partial Max-SAT problem. Our encoding algorithms for computing i-semantics based inconsistency degrees ($i = 4, 3, LP_m, Q$) were tested on several benchmarks and the experiment results showed the efficiency of this approach. The advantage of our algorithms is that they can benefit from the high optimizations of the state of the art partial Max-SAT problem solvers.

In the future, we will extend our algorithms with the ability to compute approximating inconsistency degrees. This is possible because according to the output specification of the SAT competition[7], partial Max-SAT solvers should output the current optimal solution as soon as they find a new one, which can be used to get an upper bound of the inconsistency degrees. Additionally, we will study other methods for the encodings of inconsistency degrees, such as the encoding of the computation of inconsistency degrees to the pseudo boolean problem which has mature solvers (Berre 2009). Moreover, since there are several powerful partial Max-SAT solvers, we are interested in training a meta framework which can automatically choose proper solvers to the computation of different inconsistency degrees of a given knowledge base. Finally, we plan to apply our method to measure inconsistency in other logic systems such as Description Logics and Logic Programming.

Acknowledgments We acknowledge support by the European Commission under the project OntoRule (IST-2009-231875), by the National Natural Science Foundation of China under number 60973003, and by The Ph.D. Programs Foundation of Ministry of Education of China. This work was partly realized as part of the Quaero Programme, funded by OSEO, French State agency for innovation. Guilin Qi is partially supported by Excellent Youth Scholars Program of Southeast University under grant 4009001011 and National Science Foundation of China under grant 60903010.

References

Arieli, O., and Avron, A. 1998. The value of the four values. *Artificial Intelligence* 102:97–141.

Baaz, M.; Egly, U.; and Leitsch, A. 2001. *Normal Form Transformations*. Elsevier Science. chapter 5, 273–333.

Belnap, N. D. 1977. A useful four-valued logic. In *Modern uses of multiple-valued logics*, 7–73. Reidel.

Berre, D. L. 2009. SAT4J: A Satisfiability Library for Java. http://www.sat4j.org.

Besnard, P., and Hunter, A. 1995. Quasi-classical logic: Non-trivializable classical reasoning from incosistent information. In *Proc. of ECSQARU'95*, 44–51. Springer.

Cadoli, M., and Schaerf, M. 1996. On the complexity of entailment in propositional multivalued logics. *Annals of Mathematics and Artificial Intelligence* 18(1):29–50.

Grant, J., and Hunter, A. 2006. Measuring inconsistency in knowledgebases. *Journal of Intelligent Information Systems* 27(2):159–184.

Grant, J., and Hunter, A. 2008. Analysing inconsistent first-order knowledge bases. *Artificial Intelligence* 172:1064–1093.

Grant, J. 1978. Classifications for inconsistent theories. *Notre Dame Journal of Formal Logic* 19(3):435–444.

Hunter, A., and Konieczny, S. 2005. Approaches to measuring inconsistent information. In *Inconsistency Tolerance*, 191–236. Springer.

Hunter, A., and Konieczny, S. 2006. Shapley inconsistency values. In *Proc. of KR'06*, 249–259. AAAI Press.

Hunter, A., and Konieczny, S. 2008. Measuring inconsistency through minimal inconsistent sets. In *Proc. of KR'08*, 358–366. AAAI Press.

Hunter, A. 2000. Reasoning with contradictory information using quasi-classical logic. *Journal of Logic and Computation* 10(5):677–703.

Hunter, A. 2002. Measuring inconsistency in knowledge via quasi-classical models. In *Proc. of AAAI'02*, 68–73. AAAI Press.

Hunter, A. 2006. How to act on inconsistent news: Ignore, resolve, or reject. *Data & Knowledge Engineering* 57(3):221–239.

Knight, K. 2002. Measuring inconsistency. *Journal of Philosophical Logic* 31(1):77–98.

Levesque, H. J. 1984. A logic of implicit and explicit belief. In *Proc. of AAAI'84*, 198–202. AAAI Press.

Ma, Y.; Qi, G.; Hitzler, P.; and Lin, Z. 2007. Measuring inconsistency for description logics based on paraconsistent semantics. In *Prof. of ECSQARU'07*, 30–41. Springer.

Ma, Y.; Qi, G.; Xiao, G.; Hitzler, P.; and Lin, Z. 2009. An anytime algorithm for computing inconsistency measurement. In *Proc. of KSEM'09*, 29–40. Springer.

Marques-Silva, J. 2009. The msuncore maxsat solver. Technical report, CASL/CSI, University College Dublin.

Marquis, P., and Porquet, N. 2001. Computational aspects of quasi-classical entailment. *Journal of Applied Non-Classical Logics* 11(3–4):295–312.

Papadimitriou, C., ed. 1994. *Computational Complexity*. Addison Wesley.

[7] www.maxsat.udl.cat/09/index.php?disp=requirements

Pipatsrisawat, K., and Darwiche, A. 2007. Clone: Solving weighted max-sat in a reduced search space. In *Proc. of AI'07*, 223–233. Springer.

Priest, G. 1991. Minimally inconsistent LP. *Journal Studia Logica* 50(2):321–331.

Schaerf, M., and Cadoli, M. 1995. Tractable reasoning via approximation. *Artificial Intelligence* 74(2):249–310.

Sinz, C.; Kaiser, A.; and Küchlin, W. 2003. Formal methods for the validation of automotive product configuration data. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 17(1):75–97.

Zhou, L.; Huang, H.; Qi, G.; Ma, Y.; Huang, Z.; and Qu, Y. 2009. Measuring inconsistency in DL-Lite ontologies. In *Proc. of WI'09*, 349–356. Springer.