Computing Inconsistency Measurements under Multi-Valued Semantics by Partial Max-SAT Solvers
Guohui Xiao, Zuoquan Lin, Yue Ma, Guilin Qi

To cite this version:

HAL Id: hal-00528911
https://hal.archives-ouvertes.fr/hal-00528911
Submitted on 22 Oct 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Computing Inconsistency Measurements under Multi-Valued Semantics by Partial Max-SAT Solvers

Guohui Xiao\textsuperscript{1,2}, Zuoquan Lin\textsuperscript{1}, Yue Ma\textsuperscript{3} and Guilin Qi\textsuperscript{4}
\textsuperscript{1}Department of Information Science, Peking University, China
\textsuperscript{2}Institute of Information Systems, Vienna University of Technology, Austria
\textsuperscript{3}Laboratoire d’Informatique de l’université Paris-Nord, Université Paris Nord - CNRS, France
\textsuperscript{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, \textit{LP}_m 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, \textit{LP}_m 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, \textit{LP}_m) 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), \textit{LP}_m semantics (Priest 1991), and 4-valued semantics (Belnap 1977), which produce in turn various inconsistency measures denoted by $ID_3$, $ID_{\textit{LP}_m}$ 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 \textit{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, \textit{LP}_m and Quasi Classical semantics and show that all of these problems are \textbf{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 (Schaefer 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 $\mathcal{L}_A$ with a finite set of propositional variables $A = \{p_1, \ldots, p_n\}$. A literal is a variable $p$ or its negation $\neg p$. A knowledge base is a set of propositional formulas built from $A$. $\text{Var}(K)$ denotes the set of variables occurring in $K$ and $|S|$ denotes the cardinality of a set $S$.

A clause $\gamma = l_1 \lor l_2 \lor \cdots \lor l_k$ 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

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

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

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

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

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 $\text{FOUR} = \{(t, f, B, N), \preceq\}: f \preceq N \preceq t, f \preceq B \preceq t, N \not\preceq B, B \not\preceq N$.

The 4-valued semantics of connectives $\lor, \land$ 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 $\neg t = f, \neg f = t, \neg B = B, \text{and } \neg N = N$.

The designated set of $\text{FOUR}$ 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 $\phi \in K_0, \phi^I \in \{t, B\}$. A knowledge base which has a 4-model is called 4-valued satisfiable. A knowledge base $K$ 4-valued entails a formula $\varphi$, written $K \models_4 \varphi$, if and only if each 4-model of $K$ is a 4-model of $\varphi$.

**Example 1.** Given a propositional knowledge base $K = \{p, \neg p \lor q, \neg q \lor r, \neg r, s \lor u\}$. 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, $ID_4(K, I_1) = \frac{1}{6}, ID_4(K, I_2) = \frac{7}{6}$ and $ID_4(K, I_3) = \frac{8}{16}$. Moreover, since $K$ is 2-valued unsatisfiable, every 4-model of $K$ contains at least one contradiction. So $ID_4(K) = \frac{1}{3}$.

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

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

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

$$\otimes(l_1 \lor \cdots \lor l_n, l_i) = l_1 \lor \cdots \lor l_{i-1} \lor l_{i+1} \lor \cdots \lor l_n.$$
For $I \in \varphi (\mathcal{A}^\subset)$ we define $|=Q$ in the following way:
\[
I |= Q p \quad \text{iff} \quad p \in I; \\
I |= Q \neg p \quad \text{iff} \quad p \notin I; \\
I |= Q l_1 \lor \ldots \lor l_n \quad \text{iff} \quad \{I |= Q l_i \quad \text{or} \quad I \notin|=Q l_n\} \\
\text{and} \quad \{\text{for all } i, \text{ if } I \notin|=Q l_i \quad \text{then } I \notin|=Q \otimes (l_1 \lor \ldots \lor l_n, l_i)\}; \\
I |= Q \{\gamma_1, \ldots, \gamma_m\} \quad \text{iff} \quad I |= Q \gamma_i (1 \leq i \leq m).
\]

For a knowledge base $K$ in arbitrary form, its $Q$-models are defined as the $Q$-models of $\text{CNF}(K)$, where $\text{CNF}(K)$ 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 $\{T, F, I, N\}$ to symbols in $\mathcal{A}$ in the following way, which enables the uniform way to define inconsistency degrees as given in Definition 1.

\[
p^I = \begin{cases} 
1 & \text{iff } p \in I \quad \text{and} \quad -p \notin I; \\
0 & \text{iff } p \notin I \quad \text{and} \quad -p \in I; \\
B & \text{iff } p \in I \quad \text{and} \quad -p \in I; \\
N & \text{iff } p \notin I \quad \text{and} \quad -p \notin I.
\end{cases}
\]

**Example 2 (Example 1 Continued).** $K = \{p, \neg p \lor q, \neg q \lor r, \neg r, s \lor u\}$. 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 $ID_Q(K) = ID_Q(K, I_3) = \frac{3}{4}$.

**Three-Valued and $L_{P_m}$ 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 $|=3$.

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

**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 Max-SAT 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 $I$ such that $\{\gamma \mid \gamma \in S, I |= \gamma, I |= H\} = \max_I \{\gamma \mid \gamma \in S, I |= \gamma, I |= H\}$.

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 $I |=_3 K$, then $I |\neq_4 K$.

\[
ID_4(K) = \min\{ID_4(K, I) \mid I |\neq_4 K\} \leq \min\{ID_4(K, I) \mid I |=_3 K\} = \min\{ID_3(K, I) \mid I |=_3 K\} = ID_3(K).
\]

(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 $I |\neq_4 K$ then $I' |=_3 K$. 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. \(\square\)

**Example 3 (Example 2 Continued).** $K = \{p, \neg p \lor q, \neg q \lor r, \neg r, s \lor u\}$. 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 $t$ to $N$, 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/
LP not satisfy K. So ID can only be a classical interpretation which in turn cannot 3-models.

EXACT-ID

• computational complexity of the inconsistency degree to see how Proposition 3.

ID

• related to inconsistency degrees under i

Theorem 4.

ID

ID

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

Proposition 3. ID(K) \leq IDQ(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 ID4(K) = \frac{1}{2} < \frac{2}{3} = IDQ(K). This shows that ID4(K) can be strictly less than IDQ(K).

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

Theorem 4. ID3(K) = ID LPm(K) = ID4(K) \leq IDQ(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 ID4 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):

• IDi \leq d (resp. IDi < d, IDi \geq d, IDi > d): Given a propositional knowledge base K and a number d \in [0, 1], is IDi(K) \leq d (resp. IDi(K) < d, IDi(K) \geq d, IDi(K) > d)?

• EXACT-IDi: Given a propositional knowledge base K and a number d \in [0, 1], is IDi(K) = d?

• IDi: Given a propositional knowledge base K, what is the value of IDi(K)?

Obviously, we have two trivial instances IDi \leq 1 and IDi \geq 0 with answer “yes” and another two trivial instances IDi < 0 and IDi > 1 with answer “no”.

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

Theorem 5. IDi \leq d and IDi < d (i = 3, 4, LPm, Q) are NP-complete; IDi \geq d and IDi > d (i = 3, 4, LPm, Q) are coNP-complete.

Proof. We prove these results separately as follows:

IDi \leq d is NP-complete:

The membership of IDi \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 |Conflict[I]| \leq d, which can be done in deterministic polynomial time.

IDLPm,\leq d is in NP follows from ID LPm(K) = ID4(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 IDi(K) \leq 0 which is obvious by the definition of inconsistency degree.

IDi < d is NP-complete:

Similarly to the case of IDi \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 IDi(K) < \frac{1}{|Var(K)|}.

This is because, by the definition of IDi(K), the smallest value of IDi(K) for an inconsistent knowledge base is \frac{1}{|Var(K)|}.

IDi \geq d and IDi > d are coNP-complete:

This is because IDi \geq d is the complementary problem of IDi < d and IDi > d is the complementary problem of IDi \leq d.

Theorem 6. EXACT-IDi (i = 3, 4, LPm, Q) is DP-complete.

Proof. To show that it is in DP, we have to exhibit two languages L1 \in NP and L2 \in coNP such that the set of all “yes” instances of EXACT-IDi is L1 \cap L2. This is easy by setting L1 = \{K \mid IDi(K) \leq d\} and L2 = \{K \mid IDi(K) \geq d\}.

To show the completeness, let L = L1 \cap L2 be any language in DP. We have to show that L can be reduced to EXACT-IDi. To this end, recall that IDi \leq is NP-complete and IDi \geq is coNP-complete, that is, there is a reduction R1 from L1 to IDi \leq and a reduction R2 from L2 to IDi \geq. Therefore, the reduction R from L to EXACT-IDi can be defined as follows, for any input x: R(x) = (R1(x), R2(x)). We have that R(x) is a “yes” instance of EXACT-IDi if and only if R1(x) is a “yes” instance of IDi \leq and R2(x) is a “yes” instance of IDi \geq, which is equal to x \in L.

Due to the fact that ID4(K) = ID3(K) = ID LPm,\leq(K), the complexity result of ID4 (Ma et al. 2009) can be extended as the following theorem.

A language L is in the class DP (Papadimitriou 1994) iff there are two languages L1 \in NP and L2 \in coNP such that L = L1 \cap L2.
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 $I \models 4(K)$.

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 $ID_0$.

5.1 Computing Inconsistency Degree under 4-valued Semantics

Given a knowledge base $K = \{\gamma_i | i = 1, \ldots, n\}$ 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):

$$4(\{\gamma_1, \gamma_2, \ldots, \gamma_n\}) = \{4(\gamma_1), 4(\gamma_2), \ldots, 4(\gamma_n)\};$$

$$4(l_1 \lor \ldots \lor l_k) = 4(l_1) \lor \ldots \lor 4(l_k);$$

$$4(p) = +p;$$

$$4(\neg p) = -p .$$

That is, $4(K)$ is a knowledge base over variables $\mathcal{A}^+ = \{+p, -p | p \in Var(K)\}$. 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:

$$p^I = B \text{ iff } +p^I = t \text{ and } -p^I = t;$$

$$p^I = f \text{ iff } +p^I = f \text{ and } -p^I = t;$$

$$p^I = t \text{ iff } +p^I = t \text{ and } -p^I = f;$$

$$p^I = N \text{ iff } +p^I = f \text{ and } -p^I = f.$$

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

Example 5. Let $K = \{-p, p \lor q, -q, r\}$. We have $4(K) = \{-p, +p \lor +q, -q, +r\}$. 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:

$$+p^{I_1} = t, -p^{I_1} = t, +q^{I_1} = f,$n$$

$$-q^{I_1} = t, +r^{I_1} = t, -r^{I_1} = f.$$

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 $\mathcal{A}^+$:

$$ID_4(K, I) = \frac{|b(K, I)|}{|Var(K)|};$$

$$ID_4(K) = \min_{I \models 4(K)} ID_4(K, I) = \frac{\min_{I \models 4(K)} |b(K, I)|}{|Var(K)|} .$$

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

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

$$\min_{I \models 4(K)} \{|p | p \in Var(K), +p^I = t \text{ and } -p^I = t\}|$$

$$= \min_{I \models 4(K)} \{|p | p \in Var(K), (\neg +p \lor \neg -p)^I = f\}|$$

$$= \max_{I \models 4(K)} \{|p | p \in Var(K), (\neg +p \lor \neg -p)^I = t\}|.$$

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:

$$H_4(K) = \{4(\gamma) | \gamma \in K\};$$

$$S_4(K) = \{\neg +p \lor \neg -p | p \in Var(K)\}.$$
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: $P \leftarrow \{\}$
3: $m \leftarrow |\text{Var}(K)|$
4: for all Clause $\gamma \in K$ do
5:    $P$.addHardClause($4(\gamma)$)
6: end for
7: for all Variable $p \in \text{Var}(K)$ do
8:    $P$.addSoftClause($+p \lor -p$)
9: end for
10: $I \leftarrow \text{PartialMaxSATSolver}(P)$
11: $b = |\{p \mid +p = t \land -p = t\}|$
12: 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. \qed

Next example gives a further illustration of Algorithm 1.

**Example 6.** Let $K = \{p \lor q, -p, -q, r\}$. We have $4(K) = \{+p \lor +q, -p, -q, +r\}$. Then, by Definition 3, the hard clause set of $P_2(K)$ is $\{+p \lor +q, -p, -q, +r\}$, and the soft clause set is $P_4(K) = \{\neg p \lor -p, \neg q \lor -q, \neg q \lor -r \lor -r\}$.

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:

$+l = +p \quad \text{if } l = p,
-l = -p \quad \text{if } l = -p.$

**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,

$$Q(\{\gamma_1, \ldots, \gamma_n\}) = \{Q(\gamma_1), \ldots, Q(\gamma_n)\},$$

$$Q(l_1 \lor \ldots \lor l_n) = \bigvee_{i=1}^{n} (\pm i + \neg i) \land \bigwedge_{i=1}^{n} (\pm i \land \neg i).$$

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

$$I \models Q K \iff I \models Q(K).$$

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

Now we can compute $ID_4$ 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 $A$ under QC semantics can be computed by the 2-valued semantics over the variables set $A^+$:

$$ID_4(K, I) = \frac{|\{p \in \text{Var}(K) \mid +p = t \land -p = t\}|}{|\text{Var}(K)|};$$

$$ID_4(K) = \min_{I \models Q(K)} ID_4(K, I).$$

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

$${y_i} := \pm i \land \neg i, i = 1, \ldots, n$$

$${z} := \land_{i=1}^{n} (\pm i \land \neg i).$$

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

$$Q'(\{\gamma_1, \ldots, \gamma_n\}) = \{Q'(\gamma_1), \ldots, Q'(\gamma_n)\}$$

$$Q'(l_1 \lor \ldots \lor l_n) = \bigvee_{i=1}^{n} (\pm y_i \lor z) \land \bigwedge_{i=1}^{n} (\neg y_i \lor +i) \land \bigwedge_{i=1}^{n} (\neg z \lor +i) \land \bigwedge_{i=1}^{n} (\neg z \lor -i).$$
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 \( Q'(-p) \equiv -p \).

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

\[
ID_Q(K) = \min_{I \models Q(K)} |\{ p \in \text{Var}(K) \mid +p^I = t, -p^I = t \}| / |\text{Var}(K)|.
\]

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

\[
y_i' = ( +l_i \land \neg -l_i )^I, \quad i = 1, \ldots, n ;
\]

\[
z' = ( \land_{i=1}^n ( +l_i \land -l_i ) )^I .
\]

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 \text{Var}(K), +p^I = t, -p^I = t \} \models Q(K) \} = \{ p \mid p \in \text{Var}(K), +p^I = t, -p^I = t \} \models Q(K) \} = \{ p \mid p \in \text{Var}(K) \} \} = \{ p \mid p \in \text{Var}(K) \} \} = \{ p \mid p \in \text{Var}(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:

\[
H_Q(K) = \{ Q'(\gamma) \mid \gamma \in \gamma \} ;
\]

\[
S_Q(K) = \{ \neg +p \lor -p \mid p \in \text{Var}(K) \} .
\]

Similar to Theorem 10, we have the following theorem 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 \( b(I, K) = |\{ p \mid +p^I = t \land -p^I = t \}|, m(K) = |\text{Var}(K)|. \) Then \( ID_Q(K) = b(I, K) / m(K) \).

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

\[
Q'(p \lor q) = ( y_p \lor y_q \lor z ) \land ( \neg y_p \lor +p ) \land ( \neg y_p \lor \neg -p ) \land ( \neg y_q \lor +q ) \land ( \neg y_q \lor \neg -q ) \land ( \neg z \lor +p ) \land ( \neg z \lor -p ) \land ( \neg z \lor +q ) \land ( \neg z \lor -q ) .
\]

The soft part of \( P_Q(K) \) is \( \{ \neg +p \lor -p, -q \lor \neg -q, \neg +r \lor -r \} \). \) One solution to \( P_Q(K) \) is \( I_0 = \{ t, -t, +t, -t, +t, -t \} \). \) 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**

1: procedure \( ID_Q(K) \)
2: \( P \leftarrow \{ \} \)
3: \( m \leftarrow |\text{Var}(K)| \)
4: for all \( \text{Clause} \ \gamma = \{ l_1, \ldots, l_n \} \in K \) do
5: \( P.addHardClause(y_1 \lor \ldots \lor y_n \lor z) \)
6: for \( i = 1 \) to \( n \) do
7: \( P.addHardClause(\neg y_i \lor +l_i) \)
8: \( P.addHardClause(\neg y_i \lor -l_i) \)
9: \( P.addHardClause(\neg z \lor +l_i) \)
10: \( P.addHardClause(\neg z \lor -l_i) \)
11: end for
12: end for
13: for all \( p \in \text{Var}(K) \) do
14: \( P.addSoftClause(\neg +p \lor -p) \)
15: end for
16: \( I \leftarrow \text{PartialMaxSATSolver}(P) \)
17: \( b = |\{ p \mid +p^I = t \land -p^I = t \}| \)
18: return \( b/m \)
19: 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, \( \text{Algorithm}2(K) = ID_Q(K) \) where \( \text{Algorithm}2(K) \) 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 (Papitsriswat 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;
- “ID4” and “ID4’”: 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 $K_N = \{p_i \land q_j, \neg p_i \lor \neg q_j \mid 1 \leq i, j \leq N\}$ for $N = 1, 2, 5, 7, 10, 20, 50, 100$. Obviously, $|\text{Var}(K_N)| = 2N$ 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.

Table 1: Comparison of AnyTime and Encoding Algorithm

<table>
<thead>
<tr>
<th>Instance</th>
<th>AnyTime</th>
<th>Encoding Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#V</td>
<td>#C</td>
</tr>
<tr>
<td>001.cnf</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>002.cnf</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>005.cnf</td>
<td>10</td>
<td>35</td>
</tr>
<tr>
<td>007.cnf</td>
<td>14</td>
<td>63</td>
</tr>
<tr>
<td>010.cnf</td>
<td>20</td>
<td>120</td>
</tr>
<tr>
<td>020.cnf</td>
<td>40</td>
<td>440</td>
</tr>
<tr>
<td>050.cnf</td>
<td>100</td>
<td>2600</td>
</tr>
<tr>
<td>100.cnf</td>
<td>200</td>
<td>10200</td>
</tr>
</tbody>
</table>

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 “CC”, 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.

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 ID$_4$. In constrast, the implementation based on Clone performs best for most of the instances for ID$_Q$. 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 $L(\cdot)$ used by Algorithm 1.

6http://www.satlib.org
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 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. Guinli 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


\(^7\)www.maxsat.udl.cat/09/index.php?disp=requirements


