HAL open science

Computing Inconsistency Measure based on Paraconsistent Semantics

Yue Ma, Guilin Qi, Pascal Hitzler

To cite this version:

Yue Ma, Guilin Qi, Pascal Hitzler. Computing Inconsistency Measure based on Paraconsistent Semantics. Journal of Logic and Computation, 2010, pp.1257-1281. hal-00528921

HAL Id: hal-00528921

https://hal.science/hal-00528921v1

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 Measure based on Paraconsistent Semantics

Yue Ma[1], Guilin Qi[2], and Pascal Hitzler[3]

[1]Laboratoire d'Informatique de l'université Paris-Nord (LIPN) - UMR 7030
Université Paris 13 - CNRS, France
[2]School of Computer Science and Engineering, Southeast University, Nanjing, China
[3] Kno.e.sis Center, Wright State University, Dayton, OH, USA
yue.ma@lipn.univ-paris13.fr, gqi@seu.edu.cn, pascal.hitzler@wright.edu

Abstract. Measuring inconsistency in knowledge bases has been recognized as an important problem in several research areas. Many methods have been proposed to solve this problem and a main class of them is based on some kind of paraconsistent semantics. However, existing methods suffer from two limitations: 1) They are mostly restricted to propositional knowledge bases; 2) Very few of them discuss computational aspects of computing inconsistency measures. In this paper, we try to solve these two limitations by exploring algorithms for computing an inconsistency measure of first-order knowledge bases. After introducing a four-valued semantics for first-order logic, we define an inconsistency measure of a first-order knowledge base, which is a sequence of inconsistency degrees. We then propose a precise algorithm to compute our inconsistency measure. We show that this algorithm reduces the computation of the inconsistency measure to classical satisfiability checking. This is done by introducing a new semantics, named $S[n]$-*4* semantics, which can be calculated by invoking a classical SAT solver. Moreover, we show that this auxiliary semantics also gives a direct way to compute *upper and lower bounds* of inconsistency degrees. That is, it can be easily revised to compute approximating inconsistency measures. The approximating inconsistency measures converge to the precise values if enough resources are available. Finally, by some nice properties of the $S[n]$-*4* semantics, we show that some upper and lower bounds can be computed in P-time, which says that the problem of computing these approximating inconsistency measures is tractable.

1 Introduction

Inconsistencies arise naturally when working with logic-based knowledge bases; they can, e.g., come from the merging of several knowledge bases or from ontology learning. However, inconsistencies are obviously undesirable in first-order knowledge bases. To tackle this problem, two fundamentally different approaches can be distinguished. The first is based on the assumption that inconsistencies indicate erroneous data which is to be repaired in order to obtain a consistent knowledge base, e.g., by selecting consistent subsets for the reasoning process [3,

6, 10, 11, 19, 27, 29]. The other approach considers inconsistencies as natural phenomena in realistic data which are to be handled by a logic which can tolerate it [1, 12, 20, 21], such as paraconsistent logic.

Recently, there is an increasing interest in quantifying inconsistency in inconsistent knowledge bases. This is because it is not fine-grained enough to simply say that two inconsistent knowledge bases contain the same amount of inconsistency. Indeed, it has been shown that analyzing inconsistency is helpful to decide how to act on inconsistency [14], i.e. whether to ignore it or to resolve it. Furthermore, measuring inconsistency in a knowledge base can provide some context information which can be used to resolve inconsistencies [13, 15, 16], and proves useful in different scenarios such as Software Engineering [25].

There mainly exist two categories of inconsistency measures. One is defined by the number of formulae which are responsible for an inconsistency [17, 18]. The other considers propositions in the language which are affected by inconsistency [7–9, 13, 14, 26]. The measures belonging to the second class are often based on some paraconsistent semantics because we can still find paraconsistent models for inconsistent knowledge bases. The inconsistency measure considered in this paper belongs to the second class.

Most of the proposed inconsistency measures are restricted to propositional knowledge bases. However, propositional logic has its limitations in expressive power. Furthermore, although there exist many approaches to measuring inconsistency of a knowledge base in a logical framework, very few of them provide algorithms for computing inconsistency degrees. In this paper, we consider inconsistency measures in first-order logic. Our definition of *inconsistency measure, as a sequence of inconsistency degrees*, is similar to the approach given in [8]. The difference is that our approach is based on four-valued semantics and their approach is based on quasi-classical semantics. Compared with [9], which uses four-valued semantics to study inconsistency measures, our focus is on algorithms for computing model-based measures.

The contributions of this paper can be summarized as follows:

- We give a linear reduction from the four-valued first-order semantics to the classical first-order semantics such that the four-valued entailment of first-order logic can be computed by a classical first-order reasoner. This way, our algorithms can benefit from state-of-the-art classical reasoners.
- We present an algorithm to compute inconsistency degrees by invoking a classical SAT solver. To this end, a new semantics for first-order logic, called $S[n]$-*4* semantics, is introduced.
- We further show that the $S[n]$-4 semantics provides a direct way to compute *upper and lower bounds* of inconsistency degrees (which we call *approximating inconsistency degrees*).
- Since an algorithm which is based on invoking a SAT solver is NP-hard in general, we further study some favorable properties of the $S[n]$-4 semantics to enable our algorithm to compute approximating inconsistency degrees in theoretically tractable time.

The rest of this paper is organized as follows. In the next section, we give preliminaries about a four-valued semantics of first-order logic. In Section 3, we give the reduction from four-valued logic to two-valued logic which is important for computation. In Section 4, we give the definition of inconsistency measure of a first-order knowledge base, and then, in Section 5, we give a precise algorithm to compute the inconsistency degrees. In Section 6, we show that our algorithm can be modified to generate approximating inconsistency degrees in theoretically tractable time. Finally, we discuss related work in Section 7 and conclude the paper in Section 8.

This paper is a substantial extension of the paper [22] with some content from [23].

2 Preliminaries: Four-valued Logics

2.1 Belnap's Four-valued Logic and Bilattice

The set of truth values for four-valued semantics [4, 1] contains four elements: *true, false, unknown* (or *undefined*) and *both* (or *overdefined, contradictory*). We use the symbols t, f, N, B, respectively, for these truth values. The truth value B stands for contradictory information, hence enabling four-valued logic to represent inconsistency. The value B thus can be understood to stand for both *true and false*, while N stands for *neither true nor false*, i.e. for the absence of any information about truth or falsity. The four truth values together with two orderings $\preceq_t$ and $\preceq_k$ defined below form a bilattice $\mathsf{FOUR} = (\{t, f, B, N\}, \preceq_t, \preceq_k)$, shown as in Figure 1. We have that $f \preceq_t N \preceq_t t, f \preceq_t B \preceq_t t$, where N and B are incomparable with respect to $\preceq_t$. Furthermore, $N \preceq_k t \preceq_k B, N \preceq_k f \preceq_k B$, where t and f are incomparable with respect to $\preceq_k$.

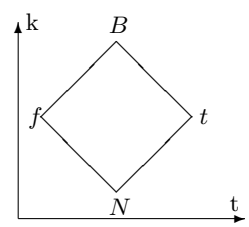

Fig. 1. $\mathcal{F}OUR$

The four-valued semantics of the connectives $\vee, \wedge$ is defined according to the upper and lower bounds with respect to $\preceq_t$: $\vee$ is given by the upper bound, $\wedge$ by the lower bound. The operator $\neg$ is defined as $\neg t = f, \neg f = t, \neg B = B$, and $\neg N = N$. By this definition, the connectives $\vee, \wedge, \neg$ are monotonic with respect to $\preceq_k$ — that is, for $x, y \in \{t, f, B, N\}$ satisfying $x \preceq_k y$, we have $\neg x \preceq_k \neg y$; For $x_1, x_2, y_1, y_2 \in \{t, f, B, N\}$ satisfying $x_1 \preceq_k x_2, y_1 \preceq_k y_2$, we have $(x_1 \circ y_1) \preceq_k (x_2 \circ y_2)$ for $\circ \in \{\wedge, \vee\}$.

2.2 Four-valued Semantics for First-order Logic

A specific first-order language $\mathcal{L}$ is determined by sets of predicate and function symbols (including constants), written $\mathcal{L} = \langle \mathcal{P}, \mathcal{F} \rangle$ for a given set of predicate symbols $\mathcal{P}$ and a set of function symbols $\mathcal{F}$. 0-ary functions are called constants and we use the symbol $\mathcal{C}$ to denote the set of constant symbols of the language under consideration. First-order formulae are built up from predicates, functions, a set of variables $\mathcal{V}$ and the set of logical symbols $\{\neg, \vee, \wedge, \forall, \exists, \rightarrow\}$, where $\alpha \rightarrow \beta$ is the short form of $\neg\alpha \vee \beta$. The language of a formula ϕ (a knowledge base Γ), written $\mathcal{L}(\phi)$ ($\mathcal{L}(\Gamma)$), consists of the predicates and functions used in the formula (knowledge base). We say that two formulae ϕ and ψ (knowledge bases Φ and Ψ) share the same language if and only if $\mathcal{L}(\phi) = \mathcal{L}(\psi)$ ($\mathcal{L}(\Phi) = \mathcal{L}(\Psi)$).

We assume that the reader is familiar with the usual definitions and notations for a language including definitions of free variables, bound variables, and ground formulae. An atom is of the form $P(t_1, ..., t_n)$, where $t_1, ..., t_n$ are terms. A first-order knowledge base considered in this paper is a finite set of first-order formulae without free variables. In this paper, whenever we want to clarify the arity of a function or a predicate, we may state the arity in parentheses following the function or the predicate symbol, e.g. $f(n), P(n)$ mean f, P are an n-ary function and predicate, respectively. We also use t (possibly with subscripts) for terms, Greek lowercase symbols α, ϕ for formulae, and uppercase Γ for a first-order knowledge base. The set of all predicates occurring in Γ is denoted by $\mathcal{P}(\Gamma)$. The cardinality of a set A is denoted by $|A|$.

Formally, a four-valued interpretation $\mathfrak{I}$ of a first-order knowledge base is defined as follows.

Definition 1 *A four-valued interpretation $\mathfrak{I} = (\Delta^{\mathfrak{I}}, \cdot^{\mathfrak{I}})$ contains a non-empty domain $\Delta^{\mathfrak{I}}$ and a mapping $\cdot^{\mathfrak{I}}$ which assigns*

- *to each constant c an element of $\Delta^{\mathfrak{I}}$, written $c^{\mathfrak{I}}$;*
- *to each n-ary function symbol f an n-ary function on $\Delta^{\mathfrak{I}}$, written $f^{\mathfrak{I}}$: $(\Delta^{\mathfrak{I}})^n \mapsto \Delta^{\mathfrak{I}}$, where $(\Delta^{\mathfrak{I}})^n = \overbrace{\Delta^{\mathfrak{I}} \times ... \times \Delta^{\mathfrak{I}}}^{n}$;*
- *to each n-ary predication symbol P a pair of n-ary relations on $\Delta^{\mathfrak{I}}$, written $\langle P_+, P_- \rangle$, where $P_+, P_- \subseteq (\Delta^{\mathfrak{I}})^n$.*

Recall, that a classical first-order interpretation maps each n-ary predicate to an n-ary relation on the domain. A four-valued interpretation assigns a pairwise n-ary relation $\langle P_+, P_- \rangle$ to each n-ary predicate P, where P_+ explicitly denotes the set of n-ary vectors which have the relation P under interpretation $\mathfrak{I}$ and P_- explicitly denotes the set of n-ary vectors which do not have the relation P under interpretation $\mathfrak{I}$. If a four-valued interpretation $\mathfrak{I}$ satisfies $P_+ \cup P_- = \Delta^{\mathfrak{I}}$ and $P_+ \cap P_- = \emptyset$, then it is a classical interpretation.

The definition of a state $\sigma : \mathcal{V} \mapsto \Delta^{\mathfrak{I}}$ remains the same as in classical semantics of first-order logic, which is a mapping assigning to each variable occurring in $\mathcal{V}$ an element of the domain. Given a state σ and an interpretation $\mathfrak{I}$, the

semantics of terms is defined inductively as follows:

$$(a)^{\mathfrak{I},\sigma} = a^{\mathfrak{I}}, \textit{ if } \ a \in \mathcal{C}$$
$$(v)^{\mathfrak{I},\sigma} = v^{\sigma}, \textit{ if } \ v \in \mathcal{V}$$
$$(f(t_1, ..., t_n))^{\mathfrak{I},\sigma} = f^{\mathfrak{I}}(t_1^{\mathfrak{I},\sigma}, ..., t_1^{\mathfrak{I},\sigma}), \textit{ if } \ f \in \mathcal{F}, t_1, ..., t_n \textit{ are terms.}$$

We denote by $\sigma\{x \mapsto d\}$ the state obtained from σ by assigning d to x while leaving other assignments to other variables unchanged.

Given an interpretation $\mathfrak{I}$ and a state σ, the four-valued semantics of an atomic formula can be defined as follows.

Definition 2 *Assume $P(t_1, ..., t_n)$ is an n-ary predicate, where $t_1, ..., t_n$ are terms. $\mathfrak{I}$ is a four-valued interpretation and σ is a state. Then the truth value assignment to atomic predicates and equality is defined as follows:*

$$(P(t_1, ..., t_n))^{\mathfrak{I},\sigma} = t, \ \textit{ if and only if } \ (t_1^{\mathfrak{I},\sigma}, ..., t_n^{\mathfrak{I},\sigma}) \in P_+^{\mathfrak{I}} \textit{ and } (t_1^{\mathfrak{I},\sigma}, ..., t_n^{\mathfrak{I},\sigma}) \notin P_-^{\mathfrak{I}}$$
$$(P(t_1, ..., t_n))^{\mathfrak{I},\sigma} = f, \ \textit{ if and only if } \ (t_1^{\mathfrak{I},\sigma}, ..., t_n^{\mathfrak{I},\sigma}) \notin P_+^{\mathfrak{I}} \textit{ and } (t_1^{\mathfrak{I},\sigma}, ..., t_n^{\mathfrak{I},\sigma}) \in P_-^{\mathfrak{I}}$$
$$(P(t_1, ..., t_n))^{\mathfrak{I},\sigma} = B, \ \textit{ if and only if } \ (t_1^{\mathfrak{I},\sigma}, ..., t_n^{\mathfrak{I},\sigma}) \in P_+^{\mathfrak{I}} \textit{ and } (t_1^{\mathfrak{I},\sigma}, ..., t_n^{\mathfrak{I},\sigma}) \in P_-^{\mathfrak{I}}$$
$$(P(t_1, ..., t_n))^{\mathfrak{I},\sigma} = N, \ \textit{ if and only if } \ (t_1^{\mathfrak{I},\sigma}, ..., t_n^{\mathfrak{I},\sigma}) \notin P_+^{\mathfrak{I}} \textit{ and } (t_1^{\mathfrak{I},\sigma}, ..., t_n^{\mathfrak{I},\sigma}) \notin P_-^{\mathfrak{I}}$$

Based on the semantics of atomic predicates, the semantics of complex formulae can be defined inductively as follows:

Definition 3 *Let φ and ϕ be two first-order formulae, $\gamma(x_1, ..., x_n)$ a formula containing n free variables, $\mathfrak{I}$ a four-valued interpretation and σ be a state. Then,*

$$(\neg\varphi)^{\mathfrak{I},\sigma} = \neg(\varphi)^{\mathfrak{I},\sigma}; (\varphi \wedge \phi)^{\mathfrak{I},\sigma} = \varphi^{\mathfrak{I},\sigma} \wedge \phi^{\mathfrak{I},\sigma}; (\varphi \vee \phi)^{\mathfrak{I},\sigma} = \varphi^{\mathfrak{I},\sigma} \vee \phi^{\mathfrak{I},\sigma}$$
$$(\forall x_1, ..., x_n.\gamma(x_1, ..., x_n))^{\mathfrak{I},\sigma} = \inf_{\sigma'=\sigma\{x_1\mapsto d_1,...,x_n\mapsto d_n\}} (\gamma(d_1, ..., d_n))^{\mathfrak{I},\sigma'}$$
$$(\exists x_1, ..., x_n.\gamma(x_1, ..., x_n))^{\mathfrak{I},\sigma} = \sup_{\sigma'=\sigma\{x_1\mapsto d_1,...,x_n\mapsto d_n\}} (\gamma(d_1, ..., d_n))^{\mathfrak{I},\sigma'},$$

where $\sigma' = \sigma\{x_1 \mapsto d_1, ..., x_n \mapsto d_n\}$ is a substitution state of σ as defined in classical FOL, and the infimum inf *(resp. supermum* sup*) is the greatest lower (least upper) bound of a set according to the ordering $\preceq_t$*[1]*.*

In Section 2.1, it is shown that the connectives $\neg, \wedge, \vee$ are monotonic with respect to the order $\preceq_k$ for the four-valued propositional logic. Next we show that the same property is maintained for four-valued first-order connectives.

Proposition 1 *The first-order connectives $\neg, \wedge, \vee, \forall, \exists$ are monotonic with respect to the ordering $\preceq_k$.*

Proof. The monotonicity of $\neg, \wedge, \vee$ can be proved similarly as in the propositional case. It is also easy to see that $\forall$ and $\exists$ are monotonic with respect to $\preceq_k$ by Definition 3 and the monotonicity of infimum and supermum. □

[1] The infimum and supermum always exist since the set of truth values under the four-valued semantics is bounded w.r.t. the ordering $\preceq_t$.

A four-valued interpretation $\mathfrak{I}$ is a 4-model of a first-order knowledge base Γ if and only if for each formula $\alpha \in \Gamma$, $\alpha^{\mathfrak{I}} \in \{t, B\}$. A knowledge base which has a 4-model is called 4-valued satisfiable. Four-valued entailment for first-order logic can be defined in a standard way as follows.

Definition 4 *Suppose Γ is a first-order knowledge base and α is a first-order formula. Γ 4-valued entails α, written $\Gamma \models_4 \alpha$, if and only if every 4-model of Γ is a 4-model of α.*

Proposition 2 *Given a first-order knowledge base Γ, Γ always has a 4-model of any domain size if UNA (the unique name assumption[2]) is not considered. If UNA is used, Γ always has 4-models whose sizes are equivalent to or larger than the number of constants in Γ.*

Proof. If UNA is not considered, consider a four-valued interpretation $\mathfrak{I}$ which assigns $\langle \Delta^{\mathfrak{I}}, \Delta^{\mathfrak{I}} \rangle$ to each predicate $p \in \mathcal{P}$. Then we can see that for any formula $\phi \in \Gamma$ $\phi^{\mathfrak{I},\sigma} = B$ for any state σ by noting that we don't have boolean constants $\{t, f\}$ in our first-order language. That is, ϕ is four-valued satisfiable under $\mathfrak{I}$, and so is Γ. If UNA is used, we further require that $\mathfrak{I}$ satisfies $a^{\mathfrak{I}} \neq b^{\mathfrak{I}}$ for each $a, b \in \mathcal{C}$. Similarly to above inference, the conclusion holds. □

This property lays the foundation for analyzing the inconsistency degrees of inconsistent knowledge bases defined by four-valued models.

Example 1 *(Canonical example) Consider knowledge base Γ consisting of the following formulae:*

$$\begin{gathered} \mathit{Penguin}(\mathit{tweety}), \\ \mathit{Bird}(fred), \\ \forall x.\mathit{Bird}(x) \rightarrow \mathit{Fly}(x), \\ \forall x.\mathit{Penguin}(x) \rightarrow \mathit{Bird}(x), \\ \forall x.\mathit{Penguin}(x) \rightarrow \neg \mathit{Fly}(x). \end{gathered}$$

Obviously, Γ has no two-valued model. However, it has a 4-model $\mathfrak{I} = (\Delta^{\mathfrak{I}}, \cdot^{\mathfrak{I}})$, where $\Delta^{\mathfrak{I}} = \{a, b\}$ and $\cdot^{\mathfrak{I}}$ is defined as $\mathit{tweety}^{\mathfrak{I}} = a, \mathit{fred}^{\mathfrak{I}} = b, \mathit{Fly}^{\mathfrak{I}}(a) = B, \mathit{Penguin}^{\mathfrak{I}}(a) = \mathit{Bird}^{\mathfrak{I}}(a) = \mathit{Bird}^{\mathfrak{I}}(b) = \mathit{Fly}^{\mathfrak{I}}(b) = t, \mathit{Penguin}^{\mathfrak{I}}(b) = f$.

Proposition 2 guarantees that we can always have finite four-valued models to explicitly display inconsistencies in knowledge bases.

3 Computing Four-valued Entailment of First-order Logic

Our four-valued semantics is an extension of classical semantics. Additionally, 4-valued entailment can be reduced to classical entailment. The reduction in the propositional case is studied in [2]. We extend it to the first-order case.

[2] That is, if c and d are distinct constants, then $c^{\mathfrak{I}} \neq d^{\mathfrak{I}}$ for each interpretation $\mathfrak{I}$.

Theorem 3 *Let Γ be a first-order knowledge base in negation normal form and ϕ be a formula. $\Gamma \models_4 \phi$ if and only if $\Theta(\Gamma) \vdash \Theta(\phi)$, where $\Theta(\phi)$ is a function defined inductively on the structure of a formula as follows and $\Theta(\Gamma) = \{\Theta(\phi) \mid \phi \in \Gamma\}$:*

- *$\Theta(c) = c$, if c is a constant.*
- *$\Theta(P(x_1, ..., x_n)) = P^+(x_1, ..., x_n)$, where P^+ is a new atomic n-ary predicate;*
- *$\Theta(\neg P(x_1, ..., x_n)) = P^-(x_1, ..., x_n)$, where P^- is a new n-ary predicate;*
- *$\Theta(\varphi_1(x_1, ..., x_n) \circ \varphi_2(y_1, ..., y_m)) = \Theta(\varphi_1(x_1, ..., x_n)) \circ \Theta(\varphi_2(y_1, ..., y_m))$, where $\circ \in \{\wedge, \vee\}$;*
- *$\Theta(\varphi_1(x_1, ..., x_n) \rightarrow \varphi_2(y_1, ..., y_m)) = \Theta(\neg\varphi_1(x_1, ..., x_n)) \vee \Theta(\varphi_2(y_1, ..., y_m))$.*
- *$\Theta(Qx.\varphi) = Qx.\Theta(\varphi)$, where $Q \in \{\forall, \exists\}$.*

Before we give the proof of this theorem, we first use the following example to illustrate the transformation defined in it.

Example 2 *(Example 1 continued) Consider Γ used in Example 1. By performing the Θ operator defined in Theorem 3, we have:*

$$\Theta(\Gamma) = \{Penguin^+(tweety), Bird^+(freg), \forall x.Bird^-(x) \vee Fly^+(x),$$
$$\forall x.Penguin^-(x) \vee Bird^+(x), \forall x.Penguin^-(x) \vee Fly^-(x)\}.$$

In order to prove this theorem, we need the following definitions and Lemmas.

Given a first-order language $\mathcal{L} = \langle \mathcal{P}, \mathcal{F} \rangle$, a new first-order language induced by $\mathcal{L}$ is defined as $\mathcal{L}' = \langle \mathcal{P}', \mathcal{F} \rangle$, where $\mathcal{P}' = \{p^+(n), p^-(n) \mid p(n) \in \mathcal{P}\}$. That is, the language $\mathcal{L}'$ differs from $\mathcal{L}$ only in the set of predicates which is formed by replacing each predicate $p(n)$ in $\mathcal{L}$ by two new predicates $p^+(n)$ and $p^-(n)$.

Definition 5 *(Two-interpretation induced by a four-interpretation) Given a four-valued interpretation $\mathfrak{I}_4$ on a language $\mathcal{L} = \langle \mathcal{P}, \mathcal{F} \rangle$ and a two-valued interpretation $\mathfrak{I}_2$ on $\mathcal{L}' = \langle \mathcal{P}', \mathcal{F} \rangle$, we call $\mathfrak{I}_2$ a* two-interpretation induced by $\mathfrak{I}_4$ *if the following conditions are satisfied:*

1. *For each function symbol $f \in \mathcal{F}$, $f^{\mathfrak{I}_2} = f^{\mathfrak{I}_4}$;*
2. *For each predicate symbol $p \in \mathcal{P}, \{p^+, p^-\} \subseteq \mathcal{P}'$, and any state σ,*

$$(p^+(t_1, ..., t_n))^{\mathfrak{I}_2,\sigma} = t \text{ if and only if } (p(t_1, ..., t_n))^{\mathfrak{I}_4,\sigma} = t \text{ or } B;$$
$$(p^-(t_1, ..., t_n))^{\mathfrak{I}_2,\sigma} = t \text{ if and only if } (p(t_1, ..., t_n))^{\mathfrak{I}_4,\sigma} = f \text{ or } B.$$

The underlying intuition of the second condition in the above definition is that $(p^+(t_1, ..., t_n))^{\mathfrak{I}_2,\sigma}$ is assigned to t if and only if $(p(t_1, ..., t_n))^{\mathfrak{I}_4,\sigma}$ has the information of being true; whilst $(p^-(t_1, ..., t_n))^{\mathfrak{I}_2,\sigma}$ is assigned to t if and only if $(p(t_1, ..., t_n))^{\mathfrak{I}_4,\sigma}$ has the information of being false.

Lemma 4 *Given a knowledge base Γ, for any four-valued interpretation $\mathfrak{I}_4$, Γ is satisfied by $\mathfrak{I}_4$ if and only if $\Theta(\Gamma)$ is satisfied by the two-interpretation induced by $\mathfrak{I}_4$ (see Theorem 3 for the definition of $\Theta(\Gamma)$).*

Proof. Let $\mathfrak{I}_2$ be the two-interpretation induced by $\mathfrak{I}_4$. For each formula $\phi \in \Gamma$ without variables, by induction on its structure, we prove that ϕ is satisfied by $\mathfrak{I}_4$ if and only if $\Theta(\phi)$ is satisfied by $\mathfrak{I}_2$ — that is, $\phi^{\mathfrak{I}_4} \in \{t, B\}$ if and only if $(\Theta(\phi))^{\mathfrak{I}_2} = t$.

Indeed, in the basic case of an atomic formula (i.e., $\phi = p(t_1, ..., t_n)$), by the definition of $\mathfrak{I}_2$ in Definition 5, we have $(\Theta(\phi))^{\mathfrak{I}_2} = t$ if and only if $\phi^{\mathfrak{I}_4} \in \{t, B\}$. Inductively, we have the following.

- If $\phi = \neg p(t_1, ..., t_n)$, then $(\Theta(\phi))^{\mathfrak{I}_2} = (p^-(t_1, ..., t_n))^{\mathfrak{I}_2} = t$ if and only if $(p(t_1, ..., t_n))^{\mathfrak{I}_4} \in \{f, B\}$ by Definition 5 if and only if $(\neg p(t_1, ..., t_n))^{\mathfrak{I}_4} \in \{t, B\}$ by Definition 3, that is, $\phi^{\mathfrak{I}_4} \in \{t, B\}$.
- If $\phi = \varphi \wedge \psi$, it is easy to see that the conclusion holds because $(\varphi \wedge \psi)^{\mathfrak{I}_4} \in \{t, B\}$ if and only if $\varphi^{\mathfrak{I}_4} \in \{t, B\}$ and $\psi^{\mathfrak{I}_4} \in \{t, B\}$ by Definition 3, if and only if $(\Theta(\varphi))^{\mathfrak{I}_2} = t$ and $(\Theta(\psi))^{\mathfrak{I}_2} = t$ by the induction hypotheise, that is, $(\Theta(\psi \wedge \varphi))^{\mathfrak{I}_2} = t$.
- If $\phi = \varphi \vee \psi$, it can be shown in a similar way by the semantics of $\vee$.
- If $\phi = \forall x_1, ..., x_n.\psi(x_1, ..., x_n)$, then by Definition 3, we have $\phi^{\mathfrak{I}_4} \in \{t, B\}$ if and only if $\psi(x_1, ..., x_n)^{\mathfrak{I}_4, \sigma} \in \{t, B\}$ for all states σ. Now for the given interpretation $\mathfrak{I}_4$, we add to the language of Γ a collection of constant symbols, one for each element in the domain of $\mathfrak{I}_4$; say that for each d in the domain of $\mathfrak{I}_4$ the constant symbol c_d is fixed. The interpretation $\mathfrak{I}_4$ is extended so that each new constant symbol is assigned to its corresponding element of the domain. Then we have $(\psi(c_{d_1}, ..., c_{d_n}))^{\mathfrak{I}_4} \in \{t, B\}$ for any domain elements $d_1, ..., d_n$. Using the induction hypothesis that the conclusion holds for the formula $\psi(c_{d_1}, ..., c_{d_n})$, we have $(\Theta(\psi)(c_{d_1}, ..., c_{d_n}))^{\mathfrak{I}_2} = t$, that is, $(\forall x_1, ..., x_n.\Theta(\psi)(x_1, ..., x_n))^{\mathfrak{I}_2} = t$ which infers that $\Theta(\phi)^{\mathfrak{I}_2} = t$ (by the definition of $\Theta(\cdot)$ in Theorem 3).
- If $\phi = \exists x_1, ..., x_n \psi(x_1, ..., x_n)$, it can be shown in a way similar to the case of $\phi = \forall x_1, ..., x_n \psi(x_1, ..., x_n)$.

In all, $\phi^{\mathfrak{I}_4} \in \{t, B\}$ if and only if $(\Theta(\phi))^{\mathfrak{I}_2} = t$. □

Definition 6 *(Four-interpretation induced by two-interpretation) Let $\Theta(\mathcal{L}) = \langle \Theta(\mathcal{P}), \mathcal{F} \rangle$. Given a four-valued interpretation $\mathfrak{I}_2$ on a language $\Theta(\mathcal{L})$, we call a two-valued interpretation $\mathfrak{I}_4$ on $\mathcal{L}$ a* four-interpretation induced by $\mathfrak{I}_2$ *if $\mathfrak{I}_4$ satisfies the following conditions:*

1. *For each function symbol $f \in \mathcal{F}$, $f^{\mathfrak{I}_4} = f^{\mathfrak{I}_2}$;*
2. *For each predicate symbol $p \in \mathcal{P}, p^+, p^- \in \Theta(\mathcal{P})$, and any state σ,*

$$
\begin{aligned}
(p(t_1, ..., t_n))^{\mathfrak{I}_4, \sigma} = t \ \ &\textit{iff.}\ \ (p^+(t_1, ..., t_n))^{\mathfrak{I}_2, \sigma} = t \ \textit{and}\ (p^-(t_1, ..., t_n))^{\mathfrak{I}_2, \sigma} = f;\\
(p(t_1, ..., t_n))^{\mathfrak{I}_4, \sigma} = f \ \ &\textit{iff.}\ \ (p^+(t_1, ..., t_n))^{\mathfrak{I}_2, \sigma} = f \ \textit{and}\ (p^-(t_1, ..., t_n))^{\mathfrak{I}_2, \sigma} = t;\\
(p(t_1, ..., t_n))^{\mathfrak{I}_4, \sigma} = B \ \ &\textit{iff.}\ \ (p^+(t_1, ..., t_n))^{\mathfrak{I}_2, \sigma} = t \ \textit{and}\ (p^-(t_1, ..., t_n))^{\mathfrak{I}_2, \sigma} = t;\\
(p(t_1, ..., t_n))^{\mathfrak{I}_4, \sigma} = N \ \ &\textit{iff.}\ \ (p^+(t_1, ..., t_n))^{\mathfrak{I}_2, \sigma} = f \ \textit{and}\ (p^-(t_1, ..., t_n))^{\mathfrak{I}_2, \sigma} = f.
\end{aligned}
$$

Lemma 5 *For any knowledge base Γ and any two-valued interpretation $\mathfrak{I}_2$ for $\Theta(\Gamma)$, Γ is satisfied by the four-interpretation induced by $\mathfrak{I}_2$ if and only if $\Theta(\Gamma)$ is satisfied by $\mathfrak{I}_2$.*

Proof. This lemma can be proved in a similar way as Lemma 4. □

Now we turn to the proof of Theorem 3.

Proof. (⇒) For each two-valued model $\mathfrak{I}_2$ of $\Theta(\Gamma)$, let $\mathfrak{I}_4$ be the 4-model induced by $\mathfrak{I}_2$. By Lemma 5, we know that $\mathfrak{I}_4$ is a model of Γ. Then by the precondition $\Gamma \models_4 \phi$, we know $\mathfrak{I}_4$ is a model of ϕ. By Lemma 5, we know $\mathfrak{I}_2$ is a model of $\Theta(\phi)$.

(⇐) For each 4-model $\mathfrak{I}_4$ of Γ, let $\mathfrak{I}_2$ be the two-valued model induced by $\mathfrak{I}_4$. By Lemma 4, we know that $\mathfrak{I}_2$ is a model of $\Theta(\Gamma)$. Then by the precondition $\Theta(\Gamma) \models_2 \Theta(\phi)$, we know $\mathfrak{I}_2$ is a model of $\Theta(\phi)$. By Lemma 4, we know $\mathfrak{I}_4$ is a model of ϕ. □

An example to illustrate Theorem 3 is given in Section 5 where the computation of four-valued entailment of a first-order knowledge base is required as a midstep of computing inconsistency degrees.

4 Inconsistency Measure by Four-valued Semantics

To measure inconsistency of a first-order knowledge base, we consider only finite knowledge bases and only finite domains in this paper, that is, the knowledge bases only contain finite formulae and the domains of their four-valued interpretations are finite. This is reasonable for practical cases because only finitely many individuals can be represented or would be used [8]. It is also reasonable in theory because we have shown that finite domains can already produce nonempty set of four-valued models for any knowledge base, which can be used to define an inconsistency measure. In contrast, an inconsistency degree based on infinite domains is defined in [9] and is shown to be binary valued $(\infty^{\top}, \infty^{\bot})$, which makes it less sensitive for distinguishing between different inconsistent knowledge bases.

The definition of inconsistency degree used in this paper comes from [8] and is defined as follows.

Definition 7 *Let Γ be a first-order knowledge base and $\mathfrak{I} = (\Delta^{\mathfrak{I}}, \cdot^{\mathfrak{I}})$ be a four-valued model of Γ. The* inconsistency degree *of Γ w.r.t. $\mathfrak{I}$, denoted $Inc_{\mathfrak{I}}(\Gamma)$, is a value in $[0, 1]$ calculated in the following way:*

$$Inc_{\mathfrak{I}}(\Gamma) = \frac{|Conflict(\mathfrak{I}, \Gamma)|}{|Ground(\mathfrak{I}, \Gamma)|}$$

where $Ground(\mathfrak{I}, \Gamma) = \{P(d_1, ..., d_n) \mid d_1, ..., d_n \in \Delta^{\mathfrak{I}}, P(n) \in \mathcal{P}(\Gamma)\}$[3], and $Conflict(\mathfrak{I}, \Gamma) = \{P(d_1, ..., d_n) \mid d_1, ..., d_n \in \Delta^{\mathfrak{I}}, P(n) \in \mathcal{P}(\Gamma), (P(d_1, ..., d_n))^{\mathfrak{I}} = B\}$.

[3] W.l.o.g., we assume that all constant symbols in knowledge bases are elements of the domain.

Intuitively, the inconsistency degree of Γ w.r.t. $\mathfrak{I}$ is the ratio of the number of conflicts divided by the amount of all possibilities. It measures to what extent a given first-order knowledge base Γ contains inconsistencies w.r.t. $\mathfrak{I}$.

Example 3 *(Example 1 continued)*
For the given knowledge base Γ and the four-valued model $\mathfrak{I}$ of Γ, we have

$$Ground(\mathfrak{I}, \Gamma) = \{Bird(a), Penguin(a), Fly(a), Bird(b), Penguin(b), Fly(b)\},$$
$$Conflict(\mathfrak{I}, \Gamma) = \{Fly(a)\}.$$

So $Inc_{\mathfrak{I}}(\Gamma) = \frac{1}{6}$.

Let us consider a different 4-valued model $\mathfrak{I}'$ of Γ where $tweety^{\mathfrak{I}'} = a, fred^{\mathfrak{I}'} = b, Fly^{\mathfrak{I}'}(a) = Penguin^{\mathfrak{I}'}(a) = Bird^{\mathfrak{I}'}(a) = Bird^{\mathfrak{I}'}(b) = Fly^{\mathfrak{I}'}(b) = B$, and where $Penguin^{\mathfrak{I}'}(b) = f$. Obviously, $|Ground(\mathfrak{I}', \Gamma)| = 6$ and $|Conflict(\mathfrak{I}', \Gamma)| = 5$, so $Inc_{\mathfrak{I}'}(\Gamma) = \frac{5}{6}$. Intuitively, $\mathfrak{I}'$ interprets more atoms as conflicting than $\mathfrak{I}$. That is, it somehow overestimates inconsistencies in Γ compared to $\mathfrak{I}$.

By this example, we see that for a first-order knowledge base, its different 4-valued models with the same domain might give rise to different inconsistency degrees. To define an inconsistency measure, the ones which do not over-estimate inconsistency degrees are preferred. To this end, a partial ordering is defined on the set of models as follows.

Definition 8 *(Model ordering w.r.t. inconsistency) Let $\mathfrak{I}_1$ and $\mathfrak{I}_2$ be two four-valued models of a first-order knowledge base Γ such that $|\Delta^{\mathfrak{I}_1}| = |\Delta^{\mathfrak{I}_2}|$. We say that $\mathfrak{I}_1$ is less inconsistent than $\mathfrak{I}_2$, written $\mathfrak{I}_1 \leq_{Incons} \mathfrak{I}_2$, if and only if $Inc_{\mathfrak{I}_1}(\Gamma) \leq Inc_{\mathfrak{I}_2}(\Gamma)$.*

As usual, $\mathfrak{I}_1 <_{Incons} \mathfrak{I}_2$ denotes $\mathfrak{I}_1 \leq_{Incons} \mathfrak{I}_2$ and $\mathfrak{I}_2 \not\leq_{Incons} \mathfrak{I}_1$, and $\mathfrak{I}_1 \equiv_{Incons} \mathfrak{I}_2$ denotes $\mathfrak{I}_1 \leq_{Incons} \mathfrak{I}_2$ and $\mathfrak{I}_2 \leq_{Incons} \mathfrak{I}_1$. $\mathfrak{I}_1 \leq_{Incons} \mathfrak{I}_2$ means that $\mathfrak{I}_1$ is more consistent than $\mathfrak{I}_2$. The models of size n which are minimal w.r.t $\leq_{Incons}$ are called preferred models and they are formally defined as follows.

Definition 9 *(Preferred Models) Let Γ be a first-order knowledge base, $\mathcal{M}_4(\Gamma)$ be the set of 4-models of Γ, and $n (n \geq 1)$ be a given cardinality.* Preferred models of size n w.r.t. $\leq_{Incons}$ *, written $PreferModel_n(\Gamma)$, are defined as follows:*

$$PreferModel_n(\Gamma) = \{\mathfrak{I} \mid |\Delta^{\mathfrak{I}}| = n; \forall \mathfrak{I}' \in \mathcal{M}_4(\Gamma), |\Delta^{\mathfrak{I}'}| = n \text{ implies } \mathfrak{I} \leq_{Incons} \mathfrak{I}'\}.$$

By Proposition 2 and Definition 9, it is not hard to see that given a first-order knowledge base and an integer n, we can always find a preferred model if the unique name assumption is not used. Otherwise, with the unique name assumption, we only can find a preferred model provided that n is not less than the number of constants appearing in the knowledge base.

As a direct consequence of Definitions 8 and 9, the following corollary shows that for any two preferred four-valued models of a first-order knowledge base with the same cardinality, the inconsistency degrees of the knowledge base with respect to them are equal.

Corollary 6 *Let Γ be a first-order knowledge base and $n(\geq 1)$ be any given positive integer. Suppose $\mathfrak{I}_1$ and $\mathfrak{I}_2$ are two four-valued models of Γ such that $|\Delta^{\mathfrak{I}_1}| = |\Delta^{\mathfrak{I}_2}| = n$, and $\{\mathfrak{I}_1, \mathfrak{I}_2\} \subseteq PreferModel_n(\Gamma)$. Then $Inc_{\mathfrak{I}_1}(\Gamma) = Inc_{\mathfrak{I}_2}(\Gamma)$.*

Based on Corollary 6, the definition of inconsistency degree based on preferred models of a first-order knowledge base is well-defined. Following [8], we also use a sequence of inconsistency degrees as the inconsistency measure of a first-order knowledge base because such a sequence reflects the inconsistency degrees of the knowledge base with respect to all finite domains.

Definition 10 *Given a first-order knowledge base Γ and a finite cardinality $n(n \geq 1)$, let $\mathfrak{I}_n$ be an arbitrary model in $PreferModels_n(\Gamma)$. The* inconsistency measure *of Γ, denoted by $IncMea(\Gamma)$, is defined as $\langle r_1, r_2, ..., r_n, ...\rangle$, where $r_n = *$ if $PreferModel_n(\Gamma) = \emptyset$, and $r_n = Inc_{\mathfrak{I}_n}(\Gamma)$ otherwise. We use $*$ as a kind of null value.*

Example 4 *(Example 1 continued) We have $IncMea(\Gamma) = \langle *, \frac{1}{6}, ..., \frac{1}{3n}, ...\rangle$ if UNA is used. If UNA is not used, $IncMea(\Gamma) = \langle \frac{1}{3}, \frac{1}{6}, ..., \frac{1}{3n}, ...\rangle$. This is because the 4-models which only assign Fly(tweety) to B are among the preferred models in the cases of any finite domain.*

In the rest of this section, we discuss some properties of the inconsistency measure as defined above.

Proposition 7 *(Strict Positivity) Given an inconsistent first-order knowledge base Γ, assume $|\mathcal{C}|$ is the number of constants of Γ and $IncMea(\Gamma) = \langle r_1, r_2, ...\rangle$. Then for $i \geq |\mathcal{C}|$, $r_i \neq *$ and $r_i > 0$.*

Proof. Suppose $i \geq |\mathcal{C}|$ and $r_i = 0$. Then there is a preferred model $\mathfrak{I}$ of Γ such that $Conflict(\mathfrak{I}, \Gamma) = \emptyset$. Since Γ is inconsistent and does not have classical models, there is a set Ω of atoms assigned the truth value N under $\mathfrak{I}$. Suppose $\mathfrak{I}'$ is obtained by replacing the assignment of N by t to each atom in Ω. Then by Proposition 1, we know $\mathfrak{I}(\phi) \preceq_k \mathfrak{I}'(\phi)$ for each formula $\phi \in \Gamma$, that is, if $\mathfrak{I}(\phi) \in \{t, B\}$, so is $\mathfrak{I}'(\phi)$. By this fact and the assumption that $\mathfrak{I}$ is a four-valued model of Γ, we know that $\mathfrak{I}'$ is also a four-valued model of Γ. In fact, $\mathfrak{I}'$ is a classical model with no assignment of B or N to atoms. This is a contradiction with the assumption that Γ is inconsistent. □

This proposition shows that for any given first-order knowledge base, its inconsistency measure cannot be a meaningless sequence (i.e., each element is the null value $*$) no matter whether UNA is used or not. Moreover, the non-zero values in the sequence start at least from the position which equals the number of constants in the first-order knowledge base, and remains greater than zero in the latter positions of the sequence.

Suppose $IncMea(\Gamma_1) = \langle r_1^1, r_2^1, ..., r_n^1, ...\rangle$ and $IncMea(\Gamma_2) = \langle r_1^2, r_2^2, ..., r_n^2, ...\rangle$. We say that Γ_1 is more inconsistent than Γ_2, written $IncMea(\Gamma_1) \geq IncMea(\Gamma_2)$, if there is number N, such that for each $n > N$, $r_n^1 \geq r_n^2$. We say that Γ_1 is as inconsistent as Γ_2, written $IncMea(\Gamma_1) = IncMea(\Gamma_2)$, if $IncMea(\Gamma_1) \geq IncMea(\Gamma_2)$ and $IncMea(\Gamma_2) \geq IncMea(\Gamma_1)$.

Theorem 8 *(Syntax Independence) Suppose $\Gamma_1 \models_4 \Gamma_2$ and $\Gamma_1 \models_4 \Gamma_2$, then $IncMea(\Gamma_1) = IncMea(\Gamma_2)$.*

Proof. The proof is obvious because $PreferModel_n(\Gamma_1) = PreferModel_n(\Gamma_2)$ and Γ_1 and Γ_2 share the same language. □

The following example illustrates that the inconsistency measure defined above is independent of the syntax of a knowledge base.

Example 5 *Suppose we have the following two knowledge bases:*

$$\Gamma_1 = \{\forall x.A(x) \wedge B(x), \forall x.\neg A(x) \wedge \neg B(x)\}$$
$$\Gamma_2 = \{\forall x.A(x), \forall x.B(x), \forall x.\neg A(x), \forall x.\neg B(x)\}.$$

By Theorem 8, we have $IncMea(\Gamma_1) = IncMea(\Gamma_2)$. This coincides with the intuition that Γ_1 and Γ_2 contain the same extent of inconsistency.

The following theorem shows that if one inconsistent knowledge base 4-valued logically implies some other one, then the former suffers from more inconsistencies than the later.

Theorem 9 *(Monotonicity w.r.t. Deductive Strength) Assume Γ_1 and Γ_2 are two knowledge bases sharing the same language. If $\Gamma_1 \models_4 \Gamma_2$, then Γ_1 is more inconsistent than Γ, that is, $IncMea(\Gamma_1) \geq IncMea(\Gamma_2)$.*

Proof. By $Model_4(\Gamma)$ we denote the set of four-valued models of Γ. Since $\Gamma_1 \models_4 \Gamma_2$, we have $Model_4(\Gamma_1) \subseteq Model_4(\Gamma_2)$. For each domain of size n, and arbitrary two preferred models $\mathfrak{I}_1 \in PreferModel_n(\Gamma_1)$ and $\mathfrak{I}_2 \in PreferModel_n(\Gamma_2)$, we claim $|Conflict(\mathfrak{I}_2, \Gamma_2)| \leq |Conflict(\mathfrak{I}_1, \Gamma_1)|$. Otherwise, by the fact that $\mathfrak{I}_1 \in Model_4(\Gamma_2)$ we know $\mathfrak{I}_2$ cannot be a preferred model of Γ_2. Moreover, we have $Ground(\mathfrak{I}_1, \Gamma_1) = Ground(\mathfrak{I}_2, \Gamma_2)$ because Γ_1 and Γ_2 share the same language. By Definitions 7 and 10, we have $IncMea(\Gamma_1) \geq IncMea(\Gamma_2)$. □

Since four-valued logic is monotonic, the following is a straightforward corollary of Theorem 9. It shows that each subset of a knowledge base is less inconsistent than the whole knowledge base if we consider the same language.

Corollary 10 *(Monotonicity w.r.t. Containment) Suppose $\Gamma_1 \subseteq \Gamma_2$, and they share the same language, we have $IncMea(\Gamma_1) \leq IncMea(\Gamma_2)$.*

As mentioned in Section 1, a way to resolve inconsistencies in knowledge bases is by pinpointing the problematic formulae and then deleting some necessary ones to obtain a consistent knowledge base. By this theorem, we can see that the process of deleting formulae without changing the vocabulary of knowledge bases is just a way to reduce the inconsistency degree. When the inconsistency degree decreases to zero, the obtained knowledge base becomes consistent.

5 Computing Inconsistency Degrees

A naive way to compute the inconsistency degree is to enumerate models to find a preferred one, and then count the number of atoms which are assigned B by the preferred model. Unfortunately, this naive method does not work because enumeration of all models of a first-order knowledge base is not practicable. Therefore, in this section, we propose an algorithm which can reduce the computation of an inconsistency degree w.r.t. an arbitrary domain size to the classical satisfiability problem, such that existing SAT solvers can be used for computing inconsistency degrees. This is achieved by means of the $S[n]$-4 semantics for first-order logic, which will be discussed in details in Section 5.1. Our $S[n]$-4 semantics is inspired by the S-3/S-1 semantics which are proposed for approximating reasoning in [28].

5.1 $S[n]$-4 Semantics

In this subsection, we define $S[n]$-4 semantics for first-order logic, which will serve as the basis for our algorithm for computing inconsistency degrees in Section 5.2.

Throughout this section, we assume that there is an underlying finite set of predicates $\mathcal{P}$ used for building all formulae and that the domain of size n is $\mathcal{D}_n = \{a_1, ..., a_n\}$. We define the *ground atomic formulae set*, written $Base(\mathcal{P}, \mathcal{D}_n)$, as the set $\{P(a_{i_1}, ..., a_{i_m}) \mid P(m) \in \mathcal{P}, a_{i_1}, ..., a_{i_m} \in \mathcal{D}_n\}$.

Definition 11 *($S[n]$-4 Interpretation) Let $\mathcal{D}_n = \{a_1, ..., a_n\}$ be a domain of size n and S be any given subset of $Base(\mathcal{P}, \mathcal{D}_n)$. A 4-valued interpretation $\mathfrak{I}$ with domain $\mathcal{D}_n$ is called an $S[n]$-4 interpretation if and only if it satisfies the following condition, where B, N, t, f are the truth values of FOUR:*

$$\phi^{\mathfrak{I}} = \begin{cases} B, & \text{if } \phi \in Base(\mathcal{P}, \mathcal{D}_n) \setminus S, \\ N \text{ or } t \text{ or } f, & \text{if } \phi \in S \end{cases}$$

That is, $\mathfrak{I}$ is an $S[n]$-4 interpretation if and only if it is a 4-valued interpretation with domain of size n which assigns the contradictory truth value B to the ground atomic formulae not in S, and non-contradictory truth values $\{N, t, f\}$ to ground atomic formulae in S.

Definition 12 *Let Γ be a first-order knowledge base. An $S[n]$-4 interpretation $\mathfrak{I}$ is an $S[n]$-4 model of Γ if and only if it is a 4-model of Γ. A knowledge base is $S[n]$-4 satisfiable if and only if it has an $S[n]$-4 model.*

Example 6 *Let $\mathcal{P} = \{p(x), q(x, y)\}$, $n = 2$, $\mathcal{D}_2 = \{a_1, a_2\}$. Then*

$$Base(\mathcal{P}, \mathcal{D}_2) = \{p(a_1), p(a_2), q(a_1, a_1), q(a_2, a_2), q(a_1, a_2), q(a_2, a_1)\}.$$

Consider $\Gamma = \{\exists x.(p(x) \wedge \neg p(x)), \forall x \exists y.q(x, y)\}$.

- *Let $S_1 = \{p(a_2), q(a_1, a_1), q(a_2, a_2), q(a_1, a_2), q(a_2, a_1)\}$. Γ is $S_1[2]$-4 satisfiable and has the following $S_1[2]$-4 model $\mathfrak{I}$: $p^{\mathfrak{I}}(a_1) = B$, and $\varphi^{\mathfrak{I}} = t$ for all $\varphi \in S_1$.*

◇ *Let $S_2 = \{p(a_1), p(a_2)\}$. Γ is $S_2[2]$-4 unsatisfiable since all $S_2[2]$-4 interpretations should map neither $p(a_1)$ nor $p(a_2)$ to B, so $\exists x.p(x) \wedge \neg p(x)$ cannot be satisfied.*

Definition 13 *($S[n]$-4 entailment) A formula ϕ is $S[n]$-4 implied by a knowledge base Γ, denoted $\Gamma \models^4_{S[n]} \phi$, if and only if every $S[n]$-4 model of Γ is an $S[n]$-4 model of ϕ.*

The relation between $S[n]$-4 satisfiability and $S[n]$-4 entailment is as follows.

Proposition 11 *Γ is $S[n]$-4 unsatisfiable if and only if $\Gamma \models^4_{S[n]} \overline{f}$, where $\overline{f}$ is a 0-ary predicate which is assigned the truth value f in FOUR under any four-valued interpretation.*[4]

Proof. This is obvious by Definition 12 and Definition 13. □

To benefit from state-of-the-art classical reasoners, in this paper we study a way to compute $S[n]$-4 entailment by invoking a classical reasoner. This contains two steps: the reduction of $S[n]$-4 entailment to the 4-valued entailment of first-order logic, which is given in the following theorem; and the reduction from the 4-valued entailment to the classical entailment of first-order logic, which has been shown in Section 3.

Theorem 12 *Let $\mathcal{D}_n = \{a_1, ..., a_n\}$ be a domain of size $n (n \geq 1)$ and S be any given subset of $Base(\mathcal{P}, \mathcal{D}_n)$, say $S = \{\alpha_1, ..., \alpha_m\}$. Denote $T = Base(\mathcal{P}, \mathcal{D}_n) \setminus S = \{\beta_1, ..., \beta_k\}$. Then the following claim holds:*

$$\Gamma \models^4_{S[n]} \varphi \text{ if and only if } \Gamma \wedge \bigwedge_{1 \leq i \leq k} (\beta_i \wedge \neg\beta_i) \wedge E_n \models_4 \varphi \vee \bigvee_{1 \leq j \leq m} (\alpha_j \wedge \neg\alpha_j), \quad (1)$$

where $E_n = \exists x_1, ..., x_n. \bigwedge_{1 \leq i,j \leq n} (x_i \not\equiv x_j) \wedge \forall y. \bigvee_{1 \leq i \leq n} (y \equiv x_i)$.[5]

Proof. Let $\Gamma' = \Gamma \wedge \bigwedge_{1 \leq i \leq k} (\beta_i \wedge \neg\beta_i) \wedge E_n$ and let $\varphi' = \varphi \vee \bigvee_{1 \leq j \leq m} (\alpha_j \wedge \neg\alpha_j)$.

($\Rightarrow$) For any 4-model $\mathfrak{I}_4$ of Γ', we show that $\mathfrak{I}_4$ satisfies φ'. First, from the assumption that $\mathfrak{I}_4$ satisfies Γ', we know $|\Delta^{\mathfrak{I}_4}| = n$ and $\mathfrak{I}_4(\beta_i) = B$ for $1 \leq i \leq k$. If there is j_0, $1 \leq j_0 \leq m$ such that $\mathfrak{I}_4(\alpha_{j_0}) = B$, then $\mathfrak{I}_4$ is a 4-valued model of φ'. Otherwise, that is, for each $1 \leq j \leq m$, $\mathfrak{I}_4(\alpha_j) \neq B$, then $\mathfrak{I}_4$ is an $S[n]$-4 model of Γ, so $\mathfrak{I}_4$ satisfies φ by hypothesis and therefore satisfies φ'.

($\Leftarrow$) For any $S[n]$-4 model $\mathfrak{I}_S$ of Γ, we show that $\mathfrak{I}_S$ satisfies φ. By definition of $\mathfrak{I}_S$, we have that $|\Delta^{\mathfrak{I}_S}| = n$, $\mathfrak{I}_S(\beta_i) = B$ for $1 \leq i \leq k$, and that $\mathfrak{I}_S(\alpha_j) \neq B$ for $1 \leq j \leq m$. So $\mathfrak{I}_S$ is a 4-model of Γ' but does not satisfy $\bigvee_{1 \leq j \leq m} (\alpha_j \wedge \neg\alpha_j)$. Then $\mathfrak{I}_S$ satisfies φ by hypothesis, that is, $\Gamma \models^4_{S[n]} \varphi$. □

[4] We will see later in Corollary 13 and Theorem 14 that the 0-ary predicate is not necessary when computing inconsistency degrees of first-order knowledge bases. Therefore, it is not contained in the four-valued first-order language in this paper.

[5] The $\equiv$ occurring here is given the classical semantics, that is, $(x \equiv y)^{\mathfrak{I},\sigma} = t$ if and only if $x^\sigma = y^\sigma$ and $(x \equiv y)^{\mathfrak{I},\sigma} = f$ if and only if $x^\sigma \neq y^\sigma$. It is only used for reducing $S[n]$-4 semantics to classical semantics. We can see later in Corollary 13 that we do not require to do four-valued reasoning with $\equiv$, so it is not introduced into the four-valued first-order language used in this paper.

Intuitively, the second entailment relation in Theorem 12 can be explained as follows: for each 4-model $\mathfrak{I}$ of Γ, if $\mathfrak{I}$ satisfies that

1. it has an n-size domain (i.e., E_n is satisfied by $\mathfrak{I}$) and
2. it assigns the truth value B to each element in $Base(\mathcal{P}, \mathcal{D}_n) \setminus S$ (i.e., the conjunction $\bigwedge_{1 \leq i \leq k}(\beta_i \wedge \neg\beta_i)$ is satisfied by $\mathfrak{I}$),

then $\mathfrak{I}$ is either an $S[n]$-4 model of Γ or it assigns B to at least one element in S (i.e., the disjunction $\bigvee_{1 \leq j \leq m}(\alpha_j \wedge \neg\alpha_j)$ is true under $\mathfrak{I}$).

Corollary 13 *Let $S = \{\alpha_1, ..., \alpha_m\}$ and let $T = Base(\mathcal{P}, \mathcal{D}_n) \setminus S = \{\beta_1, ..., \beta_k\}$. Γ is $S[n]$-4 unsatisfiable if and only if*

$$\Theta\left(\Gamma \wedge \bigwedge_{1 \leq i \leq k}(\beta_i \wedge \neg\beta_i)\right) \wedge E_n \vdash \bigvee_{1 \leq j \leq m} \Theta((\alpha_j \wedge \neg\alpha_j)).$$

Proof. Γ is $S[n]$-4 unsatisfiable if and only if $\Gamma \models^4_{S[n]} \overline{f}$ by Proposition 11, if and only if $\Gamma \wedge \bigwedge_{1 \leq i \leq k}(\beta_i \wedge \neg\beta_i) \wedge E_n \models_4 \overline{f} \vee \bigvee_{1 \leq j \leq m}(\alpha_j \wedge \neg\alpha_j)$ by Theorem 12, if and only if $\Gamma \wedge \bigwedge_{1 \leq i \leq k}(\beta_i \wedge \neg\beta_i) \wedge E_n \models_4 \bigvee_{1 \leq j \leq m}(\alpha_j \wedge \neg\alpha_j)$, which infers the conclusion by Theorem 3 and the fact that $\Theta(E_n) = E_n$. □

This result shows that $S[n]$-4 satisfiability checking can be reduced to classical entailment reasoning in first-order logic.

5.2 A Precise Algorithm for Computing Inconsistency Degrees

In this section, we first study how the computation of inconsistency degrees of a knowledge base Γ can be characterized by $S[n]$-4 satisfiability. Then we give an algorithm to compute inconsistency degrees by invoking a classical reasoner.

Without loss of generality, throughout this section, we assume that the n-size ($n \geq 1$) domain of any 4-valued interpretation is $\mathcal{D}_n = \{a_1, ..., a_n\}$. Then we uniformly write $Ground(\mathcal{D}_n, \Gamma)$ instead of $Ground(\mathfrak{I}, \Gamma)$ defined in Definition 7 for any preferred model $\mathfrak{I}$. Moreover, note that $S[n]$-4 semantics is defined with respect to a set of ground atomic formulae $Base(\mathcal{P}, \mathcal{D}_n)$ (Definition 11). In this paper, whenever we talk about using $S[n]$-4 semantics to compute the inconsistency degree r_n of a knowledge base Γ, the underlying finite set of predicates $\mathcal{P}$ is all the predicates occurring in Γ and therefore $Base(\mathcal{P}, \mathcal{D}_n) = Ground(\mathcal{D}_n, \Gamma)$.

Theorem 14 *Let $IncMea(\Gamma) = \langle r_1, ..., r_n, ...\rangle$. If $r_n \neq *$, the equation*

$$r_n = 1 - \frac{C}{|Ground(\mathcal{D}_n, \Gamma)|} \tag{2}$$

holds, where

$$C = \max\{|S| : S \subseteq Ground(\mathcal{D}_n, \Gamma) \text{ satisfying that } \Gamma \text{ is } S[n]\text{-4 satisfiable}\}.$$

Algorithm 1 Computing_Inconsistency_Degree(Γ, n)

Input: An inconsistent first-order knowledge base Γ and a positive integer n
Output: r_n // $IncMea(\Gamma) = \langle r_1, ..., r_n, ... \rangle$

1: $N \leftarrow$ the number of constants in Γ
2: **if** $n < N$ and UNA is used **then**
3: $\quad r_n \leftarrow *$
4: $\quad$ **return** r_n
5: **end if**
6: $\mathcal{D}_n \leftarrow \{a_1, ..., a_n\}$,
7: $a \leftarrow 0, b \leftarrow |Ground(\mathcal{D}_n, \Gamma)| - 1$ // $Ground(\mathcal{D}_n, \Gamma)$ is defined as in Definition 7
8: $l \leftarrow \lfloor \frac{a+b}{2} \rfloor$
9: **while** $a \neq b$ **do**
10: $\quad$ **if** there is an l-sized subset of $Ground(\mathcal{D}_n, \Gamma)$ such that Γ is $S[n]$-4 satisfiable **then**
11: $\quad\quad a \leftarrow l$, $l \leftarrow \lfloor \frac{a+b}{2} \rfloor$
12: $\quad$ **else**
13: $\quad\quad b \leftarrow l - 1$, $l \leftarrow \lfloor \frac{a+b}{2} \rfloor$
14: $\quad$ **end if**
15: **end while**
16: **return** $r_n = 1 - \frac{l}{|Ground(\mathcal{D}_n, \Gamma)|}$

Proof. Let $\mathfrak{I}_n$ be a preferred model and S be the set of atoms none of which is assigned the contradictory value B by $\mathfrak{I}_n$. Then, Γ is $S[n]$-4 satisfiable because $\mathfrak{I}_n$ is already an $S[n]$-4 model of Γ. For any subset $S' \subseteq Ground(\mathcal{D}_n, \Gamma)$ such that $|S'| > |S|$, we claim that Γ is $S'[n]$-4 unsatisfiable. Otherwise suppose $\mathfrak{I}_{S'}$ is an $S'[n]$-4 model of Γ. Obviously, $\mathfrak{I}_{S'} <_{Incons} \mathfrak{I}_n$, since $|S'| > |S|$, we get a contradiction with the definition of $\mathfrak{I}_n$. Thus $C = |Ground(\mathcal{D}_n, \Gamma)| - |Conflict(\mathfrak{I}_n, \Gamma)|$. By Definition 7 and Definition 10, Equation 2 holds. □

Theorem 14 shows that the computation of r_n can be reduced to the problem of computing the maximal cardinality of S such that S is a subset of $Ground(\mathcal{D}_n, \Gamma)$ and Γ is $S[n]$-4 satisfiable. We are now ready to give an algorithm to compute each inconsistency degree r_n of the inconsistency measure for a first-order knowledge base, as shown in Algorithm 1.

Algorithm 1 first judges whether r_n is meaningful (lines 1 to 5). If it is, lines 6 to 8 will give some necessary initializations of the n-size domain $\mathcal{D}_n$ and variables a, b, l used in the following dichotomy search. Then it turns to find, by dichotomy (lines 9 to 15), the maximal size of a subset $S \subseteq \Gamma$ such that Γ is $S[n]$-4 satisfiable. Whenever $a = b$, the value of r_n is calculated by line 16 and the procedure ends. For line 10, the condition of $S[n]$-4 satisfiability can be decided by classical entailment of first-order logic according to Corollary 13, such that each inconsistency degree r_n can be computed by a classical satisfiability reasoner.

We give an example to illustrate Algorithm 1.

Example 7 *(Example 4 continued) Assume that UNA is used and $n \geq 2$. Obviously, r_n will be meaningful. By initialization, we have $\mathcal{D}_n = \{a_i \mid 1 \leq i \leq n\}$*

and $Ground(\mathcal{D}_n, \Gamma) = \{Bird(a_i), Fly(a_i), Penguin(a_i) \mid a_i \in \mathcal{D}_n\}$. *Then the dichotomy search will begin with* $l = \lfloor \frac{3n}{2} \rfloor$. *We can see that there are* l*-sized subsets* S *which can make* Γ $S[n]$*-4 satisfiable. For example, when* $S = \{Bird(a_i) \mid 1 \leq i \leq l\}$ *is selected, an* $S[n]$*-4 model of* Γ *can be the one that assigns all atoms from* S *the truth value* t *and all others the truth value* B. *Actually, by Theorem 12 and the following Example 8, the dichotomy procedure will end with* $l_0 = 3n - 1$ *with the following* S *making* Γ $S[4]$*-4 satisfiable:*

$$S = Ground(\mathcal{D}_n, \Gamma) \setminus \{Fly(a_1)\}.$$

Then $r_n = 1 - \frac{l_0}{3n} = \frac{1}{3n}$ *is returned, which coincides with the result of Example 4.*

Example 8 *(Example 2 continued)*
To check that Γ *is* $S[4]$*-4 satisfiable for* $S = Ground(\mathcal{D}_n, \Gamma) \setminus \{Fly(a_1)\}$, *by Theorem 12 we need to check that* Γ' *entails the formula* φ *under the four-valued semantics, where* Γ' *and* φ *are given as follows:*

$$\begin{aligned}
\Gamma' &= \Gamma \wedge Fly(a_1) \wedge \neg Fly(a_1) \wedge E_n \\
\varphi &= \bigvee_{2 \leq j \leq n} (Fly(a_j) \wedge \neg Fly(a_j)) \vee \\
&\quad \bigvee_{1 \leq j \leq n} ((Bird(a_j) \wedge \neg Bird(a_j)) \vee (Penguin(a_j) \wedge \neg Penguin(a_j))).
\end{aligned}$$

By the definition of $\Theta(\cdot)$, *we easily obtain:*

$$\begin{aligned}
\Theta(\Gamma') &= \Theta(\Gamma) \wedge Fly^+(a_1) \wedge Fly^-(a_1)) \wedge E_n \\
\Theta(\varphi) &= \bigvee_{2 \leq j \leq n} (Fly^+(a_j) \wedge Fly^-(a_j)) \vee \\
&\quad \bigvee_{1 \leq j \leq n} ((Bird(a_j)^+ \wedge Bird^-(a_j)) \vee (Penguin^+(a_j) \wedge Penguin^-(a_j))).
\end{aligned}$$

A classical first-order logic reasoner yields $\Theta(\Gamma') \nvdash \Theta(\varphi)$. *According to Theorem 3, we conclude that* $\Gamma' \not\models_4 \varphi$. *Then from Proposition 11, it concludes that* Γ *is* $S[4]$*-4 satisfiable for* $S = Ground(\mathcal{D}_n, \Gamma) \setminus \{Fly(a_1)\}$.

Theorem 15 (Correctness of Algorithm 1) *Suppose that* Γ *is a first-order knowledge base and* $IncMea(\Gamma) = \langle r_1, r_2, ...r_n, ...\rangle$. *For each* $n \geq 1$, *we have*

$$r_n = Computing_Inconsistency_Degree(\Gamma, n),$$

where $Computing_Inconsistency_Degree(\Gamma, n)$ *denotes the values returned by Algorithm 1.*

Proof. To prove this, by Theorem 14, we only need to show that the search strategy used in Algorithm 1 guarantees that the selected subset S has the maximal cardinality which satisfies the condition in line 10. Indeed, this is true because of monotonicity of $S[n]$-4 satisfiability as shown by Proposition 16 below. Moreover, the search span begins with $[0, |Ground(\mathcal{D}_n, \Gamma)| - 1]$ since the inconsistency of Γ ensures that it is $S[n]$-4 unsatisfiable when $S = Ground(\mathcal{D}_n, \Gamma)$. □

Proposition 16 *(Monotonicity of $S[n]$-4 Satisfiability) For any positive integer n, consider two sets S and S' satisfying $S \subseteq S' \subseteq Base(\mathcal{P}, \mathcal{D}_n)$. If a knowledge base Γ is $S[n]$-4 unsatisfiable, then it is $S'[n]$-4 unsatisfiable.*

Proof. Assume that Γ is $S[n]$-4 unsatisfiable and that there exists an $S'[n]$-4 interpretation $\mathfrak{I}_{S'}$ satisfying Γ. We construct an $S[n]$-4 interpretation $\mathfrak{I}_S$ as

$$\phi^{\mathfrak{I}_S} = \begin{cases} B & \text{if } \phi \in S' \setminus S, \\ \phi^{\mathfrak{I}_{S'}} & \text{otherwise.} \end{cases}$$

Obviously, $\mathfrak{I}_S$ is an $S[n]$-4 model of Γ, which is a contradiction. □

At a first glance, we may think that Algorithm 1 would not terminate on any input first-order knowledge base because of the reduction to the satisfiability decision checking of first-order logic which is semi-decidable. But the next theorem shows that Algorithm 1 actually terminates on any input.

Theorem 17 (Termination of Algorithm 1) *Suppose that Γ is a first-order knowledge base and $IncMea(\Gamma) = \langle r_1, r_2, ...r_n, ...\rangle$. For each $n \geq 1$, the value of r_n can be computed in finite time by Algorithm 1.*

Proof. By Corollary 13, Algorithm 1 only invokes satisfiability decision checking with respect to finite domains $\mathcal{D}_n$. Since there will be only finite Herbrand interpretations on $\mathcal{D}_n$, the satisfiability checking can be done in finite time even though the input is a first-order knowledge base. Moreover, note that no other infinite recursions in Algorithm 1 can happen, so Algorithm 1 terminates in finite time with the correct value returned as stated in Theorem 15. □

6 Approximating Inconsistency Degrees

The algorithm given in the last section provides a general framework for computing accurate inconsistency degrees of first-order knowledge bases with respect to finite domains. In this section, we study how to adjust this framework to produce *approximating inconsistency degrees* which can be computed in theoretically tractable time.

First, we clarify the meaning of approximating inconsistency degrees by formal definitions of *upper* and *lower* bounds.

Definition 14 *(Upper/Lower Bounds) Suppose that Γ is a first-order knowledge base and $IncMea(\Gamma) = \langle r_1, ..., r_n, ...\rangle$. For each r_n, a real x (resp. y) is a* lower (resp. an upper) bound *of the nth element of the inconsistency measure of Γ, if and only if $x \leq r_n$(resp. $y \geq r_n$).*

An upper bound can be computed in the following way.

Theorem 18 (Upper bound) *Given an integer n and $S \subseteq Ground(\mathcal{D}_n, \Gamma)$, if K is $S[n]$-4 satisfiable, then $r_n \leq 1 - |S|/|Ground(\mathcal{D}_n, \Gamma)|$, where r_n is the n-th element of $IncMea(\Gamma)$.*

Proof. If K is $S[n]$-4 satisfiable, K will have an $S[n]$-4 model $\mathfrak{I}$ which assigns the truth value B to and only to every atom $p \notin S$, that is,

$$|\mathit{Conflict}(\mathfrak{I}, K)| = |\mathit{Ground}(\mathcal{D}_n, \Gamma)| - |S|.$$

By the definition of preferred models, $|\mathit{Conflict}(\mathfrak{I}, K)| \geq |\mathit{Conflict}(\mathfrak{I}_{pre}, K)|$ for every preferred model $\mathfrak{I}_{pre}$. Then we get:

$$\begin{aligned} \mathit{IncMea}(K) &= \frac{|\mathit{Conflict}(\mathfrak{I}_{pre}, K)|}{|\mathit{Ground}(\mathcal{D}_n, \Gamma)|} \\ &\leq \frac{|\mathit{Conflict}(\mathfrak{I}, K)|}{|\mathit{Ground}(\mathcal{D}_n, \Gamma)|} = 1 - \frac{|S|}{|\mathit{Ground}(\mathcal{D}_n, \Gamma)|}. \qquad \square \end{aligned}$$

Moreover, by Proposition 16, we obtain a way to compute lower bounds of each element of $\mathit{IncMea}(K)$ as shown in Theorem 19.

Theorem 19 (Lower bound) *For a given w ($1 \leq w \leq |\mathit{Ground}(\mathcal{D}_n, \Gamma)|$), if K is $S[n]$-4 unsatisfiable for each w-size subset S of $\mathit{Ground}(\mathcal{D}_n, \Gamma)$, then we have $\mathit{IncMea}(K) \geq 1 - (w-1)/|\mathit{Ground}(\mathcal{D}_n, \Gamma)|$.*

Proof. Assume that K is $S[n]$-4 unsatisfiable for any w-size subset S for n-sized domain, we claim that K must be $S'[n]$-4 unsatisfiable for any subset S' satisfying $|S'| > w$. Otherwise, suppose there exists an S' such that $|S'| > w$ and K is $S'[n]$-4 satisfiable, then by Proposition 16 there is an $S \subseteq S'$ such that $|S| = w$ and K is $S[n]$-4 unsatisfiable, which yields a contradiction.

The above claim implies that for any preferred model $\mathfrak{I}$ of K, $|\mathit{Conflict}(\mathfrak{I}, K)| \geq |\mathit{Ground}(\mathcal{D}_n, \Gamma)| - (w-1)$, such that

$$\mathit{IncMea}(K) \geq 1 - \frac{w-1}{|\mathit{Ground}(\mathcal{D}_n, \Gamma)|} = 1 - \frac{|S|-1}{|\mathit{Ground}(\mathcal{D}_n, \Gamma)|}.$$

$\square$

So far, we have presented approaches to computing a sequence of inconsistency degrees of a general first-order knowledge base and their upper and lower approximations. The approximations have the property that they will converge to precise values if there is enough computation resource (such that we can check if Γ is $S[n]$-4 satisfiable for each S). If we assume that the input first-order knowledge base is in the form of CNF (conjunctive normal formal), then the computation of $S[n]$-4 satisfiability to compute the (precise or approximating) inconsistency degrees can be achieved by invoking a SAT solver based on the results given in the previous section.

However, calling a SAT solver means invoking an algorithm for an NP-hard task, which makes the overall algorithm intractable w.r.t. the size of the input knowledge base. In the rest of the paper, in order to avoid calls to an intractable SAT solver, we explore a tractable algorithm, which is polynomial w.r.t. the size of the input knowledge base, to compute approximating inconsistency degrees.

Our approach to obtaining the tractable algorithm is based on distinguishing a tractable $S[n]$-4 entailment (proportional to the size of input knowledge bases) by which the computation of approximating inconsistency degrees becomes P-time. Before presenting our approach, we require some preparations.

As shown in the proof of Theorem 17, the computation of (precise or approximating) inconsistency degrees with respect to a finite domain only invokes satisfiability decision checking with respect to a Herbrand domain of the same size. Then by grounding each clause with elements of the Herbrand base, we actually get a propositional knowledge base. That is, the computation of (precise or approximating) inconsistency degrees of a first-order knowledge base is reduced to the satisfiability decision problem in the propositional logic case. For this reason and in order to simplify notation, the following lemmas are given directly in terms of propositional logic. We write the language used by a propositional knowledge base K as $Var(K) = \{p \mid p \text{ is a propositional variable used in } K\}$.

Lemma 20 *Given $Var(K)$, a subset $S = \{s_1, ..., s_k\}$ of $Var(K)$ and its complement $\overline{S} = Var(K) \setminus S$. Suppose φ is a formula such that $Var(\{\varphi\}) \subseteq Var(K)$. Then $K \models^4_S \varphi$ if and only if*

$$K \wedge \bigwedge_{q \in \overline{S}} (q \wedge \neg q) \models_4 \varphi \vee (c_1 \vee ... \vee c_k)$$

holds for any combination $\{c_1, ..., c_k\}$, where each c_i is either s_i or $\neg s_i$ $(1 \leq i \leq k)$.

Proof. Let $K' = K \wedge \bigwedge_{q \in \overline{S}} (q \wedge \neg q)$ and $\varphi' = \varphi \vee \bigvee_{i=1}^{k} (s_i \wedge \neg s_i)$. First we prove that $K \models^4_{S[n]} \varphi$ if and only if $K' \models^4 \varphi'$.

($\Rightarrow$) For any 4-model M_4 of K', $M_4(q) = B$ for $q \in \overline{S}$. If there is j_0, $1 \leq j_0 \leq k$ such that $M_4(s_{j_0}) = B$, then M_4 is a 4-valued model of φ'. Otherwise, if for each $1 \leq j \leq k$, $M_4(p_j) \neq B$, then M_4 is an $S[n]$-4 model of K, so M_4 satisfies φ by the hypothesis and therefore satisfies φ'.

($\Leftarrow$) For any $S[n]$-4 model M_S of K, by the definition of M_S, $M_S(q) = B$ for $q \notin S$, and $M_S(s_j) \neq B$ for $1 \leq j \leq k$. So M_S is a 4-model of K' but does not satisfy $\bigvee_{1 \leq j \leq k} (s_j \wedge \neg s_j)$. Then M_S satisfies φ by hypothesis and $K \models^4_S \varphi$.

Secondly, note that the distributive laws w.r.t $\vee$ and $\wedge$ still hold under the four-valued semantics, so we know

$$\bigvee_{i=1}^{k} (s_i \wedge \neg s_i) \equiv_4 \bigwedge_{\substack{c_i = s_i, \neg s_i \\ 1 \leq i \leq k}} (c_1 \vee ... \vee c_k).$$

Hence, $K \models^4_{S[n]} \varphi$ if and only if $K \wedge \bigwedge_{q \in \overline{S}} (q \wedge \neg q) \models_4 \varphi \vee (c_1 \vee ... \vee c_k)$. $\square$

This lemma shows another way to reduce S-4 entailment to 4-entailment. Note, in particular, that if φ is in CNF, then the right-hand side of the reduced 4-entailment retains CNF form by a little bit of rewriting, as follows: Suppose $\varphi = C_1 \wedge ... \wedge C_n$. Then we have $\varphi \vee (c_1 \vee ... \vee c_k) = (C_1 \vee c_1 \vee ... \vee c_k) \wedge ... \wedge (C_n \vee c_1 \vee ... \vee c_k)$ which is still in CNF and its size is linear in that of $\varphi \vee (c_1 \vee ... \vee c_k)$.

Lemma 21 ([5]) *For any propositional knowledge base K in any form and φ in CNF, there exists an algorithm for deciding if $K \models_4 \varphi$ in $\mathcal{O}(|K| \cdot |\varphi|)$ time.*

By Lemmas 20 and 21, we have the following theorem:

Theorem 22 (Complexity) *Given a propositional knowledge base K, there exists an algorithm for deciding if $K \models_S^4 \varphi$ and deciding if K is S-4 satisfiable in $\mathcal{O}(|K|(|\varphi| + |S|) \cdot 2^{|S|})$ and $\mathcal{O}(|K||S| \cdot 2^{|S|})$ time, respectively.*

Proof. From Lemma 20, we know that $K \models_S^4 \varphi$ can be tested by performing $2^{|S|}$ times the testing if $T \models_4 \gamma$, where T has size proportional to the size of $|K|$ and $|\gamma|$ is linear in $|\varphi| + |S|$. By Lemma 21, we obtain an algorithm with the desired upper bound.

To decide if K is S-4 satisfiable, note that it holds if and only if $K \models_S^4 \overline{f}$ by Proposition 11. By the algorithm used for S-4 entailment given above and its complexity together with the fact $|f| = 1$, the desired complexity bound for S-4 satisfiability is obtained. □

Theorem 22 shows that S-4 entailment and S-4 satisfiability checking can both be decided in polynomial time w.r.t the size of K, exponential w.r.t that of S, though. So they can be justified in P-time if $|S|$ is limited by a logarithmic function of $|K|$.

Theorem 22 and Theorem 18 together show that for a monotonic sequence of sets S_1,...,S_k, where $|S_i| < |S_{i+1}|$ for any $1 \leq i \leq k-1$, if we can show that K is S_i-4 ($i = 1, ..., k$) satisfiable one by one, then we can get a sequence of decreasing upper bounds of the inconsistency degree of K in $\mathcal{O}(|K||S_i| \cdot 2^{|S_i|})$ time. If $|S_i| = \mathcal{O}(\log |K|)$, it is easy to see that the computation of an upper bound is done in polynomial time with respect to the size of K.

Theorem 22 and Theorem 19 together show that for a monotonic sequence of sets $S_1, ..., S_m$ satisfying $|S_i| < |S_{i+1}|$, if we can prove that K is $|S_i|$-4 unsatisfiable[6] for each $i \in [1, m]$, then we can get a series of increasing lower bounds of the inconsistency degree of K in $\mathcal{O}(\binom{|Var(K)|}{|S_i|}|K||S_i| \cdot 2^{|S_i|})$ time at most. This is because for each S_i, it needs maximal $\binom{|Var(K)|}{|S_i|}$ times testing of S-4 unsatisfiability. If and only if $|S_i|$ is limited by a constant, we have that each lower bound can be obtained in polynomial time according to Proposition 23 by replacing k in the proposition with $|S_i|$ and replacing n with $|Var(K)|$.

Proposition 23 *Let $f(n) = \mathcal{O}(2^k \binom{n}{k} nk)$ where $0 \leq k \leq n$. There exists $p \in \mathbb{N}$ such that $f(n) = \mathcal{O}(n^p)$ if and only if k is limited by a constant.*

Because of the length of the proof of this proposition, it is given in the appendix.

[6] For the sake of simplicity, we say that K is l-4 satisfiable for $l \in \mathbb{N}$, if there is a subset $S \subseteq Var(K)$ such that K is S-4 satisfiable. We say that K is l-4 unsatisfiable if K is not l-4 satisfiable.

7 Related Work

Using paraconsistent models for measuring inconsistency has been studied in the literature [7–9, 13, 14, 26]. These approaches can basically be divided into multi-valued model based approaches and quasi-classical model based approaches. In this paper, we mainly studied the four-valued model based approach for measuring inconsistency degrees of arbitrary first-order logic knowledge bases. The reason why we use four-valued models is that the four-valued semantics for the whole first-order language can be implemented by a linear reduction to the classical semantics as shown in Theorem 3. While for the quasi-classical logic, this is only achieved when restricted to knowledge bases in CNF because the reduction of the QC semantics for the whole language is still an open problem [24]. This property of the four-valued semantics is useful for the algorithmic aspect of computing inconsistency degrees.

Restricted to multi-valued semantics, there are still some alternatives ways to define inconsistency measures for first-order logic. They differ in how they deal with different domain sizes of the paraconsistent models when defining inconsistency degrees, which is a core design decision required due to the presence of variables in first-order logic. Our definition follows the approach in [8] which defines the inconsistency measure as a sequence of inconsistency degrees. Each inconsistency degree is defined with respect to a specific finite domain. In [9], for a set of domains, three sorts are distinguished to define an inconsistency measure, named the bounded frame, the unbounded frame, and the infinite frame. While the inconsistency measure defined w.r.t bounded frame is the simplest case, the one defined w.r.t. unbounded frame is more related to our definition. Suppose that our inconsistency measure is $\langle r_1, r_2, ..., r_n, ...\rangle$. In fact, the inconsistency measure defined in [9] is the limit of the sequence of our inconsistency degrees (i.e. $\lim_{i\to\infty} r_i$). One of the significant results obtained in [9] is that the limit exists and can be expressed in the form of a *special function*. However, how to compute that function is not given for a general first-order knowledge base in [9]. Our result shows a way to compute each r_n which provides a way to approximately compute the inconsistency measure $\lim_{i\to\infty} r_i$. The inconsistency measure defined w.r.t. an infinite frame allows to consider any infinite domain. However, the values obtained in this case are binary ($\{\infty^{\top}, \infty^{\bot}\}$) which is not fine-grained enough to distinguish between different inconsistent knowledge bases.

Compared to the existing work which mainly studies inconsistency measures for propositional knowledge bases, the work [7–9] and ours are about first-order knowledge bases. To the best of our knowledge, this paper is the first work that explores the computation of both precise and approximating inconsistency degrees.

8 Discussions and Future Work

In this paper, we studied the computational aspects of calculating the sequence of inconsistency degrees of a first-order knowledge base. By the proposed $S[n]$-4

semantics, we explored an algorithm to compute precise inconsistency degrees by invoking a SAT solver, and then a way to compute approximating inconsistency degrees. In particular, we studied a theoretically tractable approach to computing approximating inconsistency degrees.

The computation of the inconsistency degree sequence $\langle r_1, ..., r_n, ...\rangle$ of a first-order knowledge base Γ can be achieved using Algorithm 1. However, a practical problem is that the infinitary definition of $IncMea(\Gamma)$ makes it impossible to obtain the exact value of $IncMea(\Gamma)$ in finite time. We can however set a termination condition in order to guarantee that an answer will be obtained. Assume time (resource) is used up, a possible way is to use the already obtained partial sequence $\langle r_1, ..., r_n\rangle$ as an approximating value of $IncMea(\Gamma)$. This point causes the consideration of defining inconsistency degree in other ways, which is among our future work.

Our approach can be extended to compute other notions of inconsistency degrees, such as the one based on the quasi-classical (QC for short) semantics [13], as follows. Define an $S[n]$-QC semantics similar to Definitions 11, 12, and 13 by replacing the corresponding four-valued semantics by QC semantics. Then check that the computation of inconsistency degrees can be reduced to $S[n]$-QC satisfiability (similar to Theorem 14). Next, make sure that $S[n]$-QC satisfiability can be simulated by QC entailment (similar to Theorem 12) and then be reduced to classical entailment, which holds by [24]. This way, we can compute the inconsistency degree defined by the QC semantics by invoking an SAT solver.

In the future, we will also work on developing a framework to combine different kinds of inconsistency handling approaches based on the computation of inconsistency degrees.

9 Acknowledgement

This work was 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. Pascal Hitzler is partially supported by the Wright State University Research Council. We are very grateful to the anonymous reviewers of this paper and Jian Wang for their valuable comments and technical suggestions.

References

1. O. Arieli and A. Avron. The value of the four values. *Artificial Intelligence*, 102 (1):97–141, 1998.
2. O. Arieli and M. Denecker. Reducing preferential paraconsistent reasoning to classical entailment. *Journal of Logic and Computation*, 13(4):557–580, 2003.
3. C. Baral, S. Kraus, J. Minker, and V. S. Subrahmanian. Combining knowledge bases consisting of first-order analysis. *Computational Intelligence*, 8:45–71, 1992.
4. N. D. Belnap. A useful four-valued logic. In G.Epstein and J.M.Dunn, editors, *Modern uses of multiple-valued logics*, pages 7–73. Reidel Publishing Company, 1977.

5. M. Cadoli and M. Schaerf. On the complexity of entailment in propositional multivalued logics. *Annals of Mathematics and Artificial Intelligence*, 18(1):29–50, 1996.
6. G. Friedrich and K. M. Shchekotykhin. A general diagnosis method for ontologies. In Y. Gil, E. Motta, V. R. Benjamins, and M. A. Musen, editors, *Proceedings of the 4th International Conference on Semantic Web (ISWC'05)*, pages 232–246. Springer, 2005.
7. J. Grant. Classifications for inconsistent theories. *Notre Dame Journal of Formal Logic*, 19(3):435–444, 1978.
8. J. Grant and A. Hunter. Measuring inconsistency in knowledgebases. *Journal of Intelligent Information Systems*, 27(2):159–184, 2006.
9. J. Grant and A. Hunter. Analysing inconsistent first-order knowledgebass. *Artificial Intelligence*, 172:1064–1093, 2008.
10. A. Y. Halevy. Answering queries using views: A survey. *VLDB Journal*, 10(4): 270–294, 2001.
11. Z. Huang, F. van Harmelen, and A. ten Teije. Reasoning with inconsistent ontologies. In L. P. Kaelbling and A. Saffiotti, editors, *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI'05)*, pages 454–459. Professional Book Center, 2005.
12. A. Hunter. Reasoning with contradictory information using quasi-classical logic. *Journal of Logic and Computation*, 10(5):677–703, 2000.
13. A. Hunter. Measuring inconsistency in knowledge via quasi-classical models. In G. Gottlob and T. Walsh, editors, *Proceedings of the 18th International Joint Conference on Artificial Intellignce (IJCAI'03)*, pages 68–73. Morgan Kaufmann, 2003.
14. A. Hunter. How to act on inconsistent news: Ignore, resolve, or reject. *Data & Knowledge Engineering*, 57(3):221–239, 2006.
15. A. Hunter and S. Konieczny. Approaches to measuring inconsistent information. In L. E. Bertossi, A. Hunter, and T. Schaub, editors, *Inconsistency Tolerance*, volume 3300 of *Lecture Notes in Computer Science*, pages 191–236. Springer, 2005. ISBN 3-540-24260-0.
16. A. Hunter and S. Konieczny. Shapley inconsistency values. In P. Doherty, J. Mylopoulos, and C. A. Welty, editors, *Proceedings of the 10th International Conference on Principles of Knowledge Representation and Reasoning (KR'06)*, pages 249–259. AAAI Press, 2006.
17. A. Hunter and S. Konieczny. Measuring inconsistency through minimal inconsistent sets. In G. Brewka and J. Lang, editors, *Proceedings of the 11th International Conference on Principles of Knowledge Representation and Reasoning (KR'08)*, pages 358–366. AAAI Press, 2008.
18. K. Knight. Measuring inconsistency. *Journal of Philosophical Logic*, 31:77–98, 2001.
19. S. Konieczny and R. P. Pérez. On the logic of merging. In A. G. Cohn, L. K. Schubert, and S. C. Shapiro, editors, *Proceedings of the 6th International Conference on Principles of Knowledge Representation and Reasoning (KR'98)*, pages 488–498. Morgan Kaufmann, 1998.
20. Y. Ma and P. Hitzler. Paraconsistent reasoning for OWL2. In A. Polleres and T. Swift, editors, *Proceedings of the 3rd International ConferenceWeb on Reasoning and Rule Systems, (RR'09)*, pages 197–211. Springer, 2009.
21. Y. Ma, P. Hitzler, and Z. Lin. Algorithms for paraconsistent reasoning with OWL. In E. Franconi, M. Kifer, and W. May, editors, *Proceedings of the 4th European Semantic Web Conference (ESWC'07)*, pages 399–413. Springer, 2007.

22. Y. Ma, G. Qi, P. Hitzler, and Z. Lin. An algorithm for computing inconsistency measurement by paraconsistent semantics. In K. Mellouli, editor, *Proceedings of the 9th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU'07)*, pages 91–102. Springer, 2007.
23. Y. Ma, G. Qi, G. Xiao, P. Hitzler, and Z. Lin. An anytime algorithm for computing inconsistency measurement. In D. Karagiannis and Z. Jin, editors, *Proceedings of the 3rd International Conference on Knowledge Science, Engineering and Management (KSEM'09)*, volume 5914 of *Lecture Notes in Computer Science*, pages 29–40. Springer, 2009.
24. P. Marquis and N. Porquet. Computational aspects of quasi-classical entailment. *Journal of Applied Non-Classical Logics*, 11(3-4):295–312, 2001.
25. K. Mu, Z. Jin, R. Lu, and W. Liu. Measuring inconsistency in requirements specifications. In L. Godo, editor, *Proceedings of the 8th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty, (ECSQARU'05)*, pages 440–451. Springer, 2005.
26. C. A. Oller. Measuring coherence using LP-models. *Journal of Applied Logic*, 2 (4):451–455, 2004.
27. B. Parsia, E. Sirin, and A. Kalyanpur. Debugging OWL ontologies. In A. Ellis and T. Hagino, editors, *Proceedings of the 14th International Conference on World Wide Web (WWW'05)*, pages 633–640. ACM, 2005.
28. M. Schaerf and M. Cadoli. Tractable reasoning via approximation. *Artificial Intelligence*, 74(2):249–310, 1995.
29. S. Schlobach. Diagnosing terminologies. In D. Fox and C. P. Gomes, editors, *Proceedings of the 20th Twentieth National Conference on Artificial Intelligence (AAAI'05)*, pages 670–675. AAAI Press/The MIT Press, 2005.

Appendix. Proof of Proposition 23

Proof. Sufficiency is obvious. We prove necessity which includes two steps:

1. We prove that by the condition of the proposition, $\lim_{n\to\infty}\frac{k(n)}{n}$ must exist and be zero (i.e. $\lim_{n\to\infty}\frac{k(n)}{n}=0$).
2. Based on $\lim_{n\to\infty}\frac{k(n)}{n}=0$, we prove that $\lim_{n\to\infty}k(n)\neq\infty$ which is the conclusion.

Both of the these two claims are proven by contradiction. To this end, we need to rewrite $A(n)$ by Stirling's formula[7]:

$$n!=\sqrt{2\pi n}\left(\frac{n}{e}\right)^n e^{\frac{\theta}{12n}},\ 0<\theta<1.$$

We have

$$A(n)=\binom{n}{k(n)}2^{k(n)}=\frac{e^{\frac{\theta_1}{12n}-\frac{\theta_2}{12(n-k(n))}-\frac{\theta_3}{12k(n)}}}{(n-k(n))^{n-k(n)}k(n)^{k(n)}}\sqrt{\frac{n}{(n-k(n))k(n)}}\frac{2^{k(n)}n^n}{\sqrt{2\pi}},$$

[7] See e.g. V.A. Zorich, *Mathematical Analysis*, Springer, 2004, Section 17.3.5

where $0 < \theta_i < 1$ for $i = 1, 2, 3$. Denote $B(n) = \frac{2^{k(n)} n^n}{(n-k(n))^{n-k(n)} k(n)^{k(n)}}$. Obviously, we have

$$C_1 \frac{1}{n} B(n) \leq A(n) \leq C_2 n B(n), \tag{3}$$

where C_1, C_2 are two positive constants. We consider the limit

$$\limsup_{n\to\infty} \frac{k(n)}{n}.$$

Now we prove the first claim, that is, by the condition that $A(n)$ is the same order with some polynomial about n (i.e., $A(n) = O(n)$), $\lim_{n\to\infty} \frac{k(n)}{n}$ must exist and be zero. Otherwise, suppose the subsequence $\{n_i\}_{i=1}^{\infty}$ of $\mathbb{N}$ satisfies

$$\lim_{i\to\infty} \frac{k(n_i)}{n_i} = \limsup_{n\to\infty} \frac{k(n)}{n} = \beta > 0.$$

Obviously, $0 < \beta \leq 1$. By this assumption, for any $p \in \mathbb{N}$, we have

$$\lim_{i\to\infty} (\ln 2 \cdot k(n_i) - p \ln n_i) = \lim_{i\to\infty} [\ln(n_i) \cdot (\ln 2 \cdot \frac{k(n_i)}{\ln n_i} - p)] = \infty,$$

which implies $\lim_{i\to\infty} \frac{B(n_i)}{n_i^p} = \infty$. By Inequation 3, we have $\lim_{i\to\infty} \frac{A(n_i)}{n_i^p} = \infty$ which contradicts the assumption that $A(n) = O(n)$. We have proved this claim.

Secondly, we prove that $\lim_{n\to\infty} k(n) \neq \infty$. Otherwise, assume $\lim_{n\to\infty} k(n) = \infty$. Since $A(n) = O(n)$ and by Inequation 3, there exists $p \in \mathbb{N}$ and a positive constant C_3 such that

$$0 \leq \liminf_{n\to\infty} \frac{B(n)}{n^p} \leq \limsup_{n\to\infty} \frac{B(n)}{n^p} \leq C_3.$$

Now we select a subsequence $\{n_i\}_{i=1}^{\infty}$ of $\mathbb{N}$ that satisfies

$$\lim_{i\to\infty} \frac{B(n_i)}{n_i^p} = \limsup_{n\to\infty} \frac{B(n)}{n^p} \leq C_3,$$

if and only if

$$\lim_{i\to\infty} (\ln(B(n_i)) - p \ln(n_i)) \leq \ln(C_3),$$

if and only if

$$\begin{aligned} \lim_{i\to\infty} \{k(n_i)(\ln(2n_i) - \ln(k(n_i))) + \frac{n_i - k(n_i)}{n_i} \cdot k(n_i) \cdot \ln[(1 - \frac{k(n_i)}{n_i})^{-\frac{n_i}{k(n_i)}}] \\ - p \ln(n_i)\} \leq \ln(C_3). \end{aligned} \tag{4}$$

By dividing $k(n_i)$ on both sides of the inequation above and by the fact that $\lim_{n\to\infty} \frac{k(n)}{n} = 0$ and the assumption $\lim_{n\to\infty} k(n) = \infty$, we have

$$\lim_{i\to\infty} (\frac{p}{k(n_i)} \ln(n_i) + \ln(k(n_i)) - \ln(n_i)) \geq 1 + \ln 2,$$

that is

$$\lim_{i\to\infty} \ln(\frac{n^{\frac{p}{k(n_i)}} k(n_i)}{n_i}) \geq \ln 2e. \tag{5}$$

As $\lim_{n\to\infty} \frac{k(n)}{n} = 0$, by equation 5 we have $\lim_{i\to\infty} n_i^{\frac{p}{k(n_i)}} = \infty$. So we have $\lim_{i\to\infty} \frac{k(n_i)}{p \ln n_i} = 0$. This implies that for any $\alpha > 0$, we have

$$\lim_{i\to\infty} \frac{k(n_i)}{n_i^{\alpha}} = 0. \tag{6}$$

If we calculate Inequation 4 again carefully, we obtain

$$\lim_{i\to\infty} \{k(n_i)(\ln \frac{2n_i}{k(n_i)} + \frac{n_i - k(n_i)}{n_i} \ln[(1 - \frac{k(n_i)}{n_i})^{-\frac{n_i}{k(n_i)}}]) - p\ln(n_i)\} \leq \ln(C_3). \tag{7}$$

Let $\frac{k(n_i)}{n_i} = x$ and $f(x) \triangleq \frac{x-1}{x} \ln(1-x) = \frac{n_i - k(n_i)}{n_i} \ln[(1 - \frac{k(n_i)}{n_i})^{-\frac{n_i}{k(n_i)}}]$. Obviously, x is very small when i is big enough. By Taylor's formula with the Peano form of the remainder $\ln(1-x) = -x - \frac{1}{2}x^2 + O(x^3)$, we have $f(x) = 1 - \frac{1}{2}x + O(x^2)$. So we obtain

$$\frac{n_i - k(n_i)}{n_i} \ln[(1 - \frac{k(n_i)}{n_i})^{-\frac{n_i}{k(n_i)}}] - 1 = (-\frac{1}{2} + o(1)) \cdot \frac{k(n_i)}{n_i}. \tag{8}$$

Then by Inequation 7 and Equation 8 we have

$$\lim_{i\to\infty} \{k(n_i)(\ln \frac{2n_i}{k(n_i)} + 1 + (-\frac{1}{2} + o(1)) \cdot \frac{k(n_i)}{n_i}) - p\ln(n_i)\} \leq \ln(C_3).$$

By Equation 6, let $\alpha = \frac{1}{2}$, and we have

$$\lim_{i\to\infty} \{k(n_i) \ln \frac{2en_i}{k(n_i)} - p\ln(n_i) - \ln(C_3)\} \leq 0.$$

That is,

$$\lim_{i\to\infty} \{\frac{(2e)^{k(n_i)}}{C_3 n_i^p} \cdot (\frac{n_i}{k_i})^{k(n_i)}\} \leq 1,$$

which further requires

$$\lim_{i\to\infty} \{\frac{(2e)^{k(n_i)}}{C_3} \cdot (\frac{n_i^{\frac{k(n_i)-p}{k(n_i)}}}{k(n_i)})^{k(n_i)}\} \leq 1.$$

But this is impossible by Equation 6 and our assumption that $\lim_{n\to\infty} k(n) = \infty$. Therefore, we have $\lim_{n\to\infty} k(n) < \infty$, which is the conclusion we wanted to prove. □