

Measuring Strong Inconsistency

Markus Ulbricht

Department of Computer Science
Leipzig University
Germany

Matthias Thimm

Institute for Web Science and Technologies
University of Koblenz-Landau
Germany

Gerhard Brewka

Department of Computer Science
Leipzig University
Germany

Abstract

We address the issue of quantitatively assessing the severity of inconsistencies in nonmonotonic frameworks. While measuring inconsistency in classical logics has been investigated for some time now, taking the nonmonotonicity into account poses new challenges. In order to tackle them, we focus on the structure of minimal strongly \mathcal{K} -inconsistent subsets of a knowledge base \mathcal{K} —a generalization of minimal inconsistency to arbitrary, possibly nonmonotonic, frameworks. We propose measures based on this notion and investigate their behavior in a nonmonotonic setting by revisiting existing rationality postulates, analyzing the compliance of the proposed measures with these postulates, and by investigating their computational complexity.

1 Introduction

In the literature on inconsistency measurement—see e. g. (Hunter and Konieczny 2004; Grant and Hunter 2006; Thimm 2016)—inconsistency measures are functions that aim at assessing the severity of the inconsistency in knowledge bases formalized in propositional logic. The basic intuition behind an inconsistency measure \mathcal{I} is that the larger the inconsistency in \mathcal{K} the larger the value $\mathcal{I}(\mathcal{K})$.

A simple but popular approach to measure inconsistency is to take the number of minimal inconsistent subsets (Hunter and Konieczny 2008), i. e., to define $\mathcal{I}_{\text{MI}}(\mathcal{K}) = |I_{\text{min}}(\mathcal{K})|$, where $I_{\text{min}}(\mathcal{K})$ is the set of all minimal inconsistent subsets of a knowledge base \mathcal{K} . This measure already complies with many basic ideas of inconsistency measurement, in particular $\mathcal{I}_{\text{MI}}(\mathcal{K}) = 0$ iff \mathcal{K} is consistent. By also taking the size and the relationships of minimal inconsistent subsets into account, a wide variety of different inconsistency measures can be defined on top of that idea, see e. g. (Hunter and Konieczny 2008; Jabbour et al. 2016; Jabbour and Sais 2016).

Measuring inconsistency in nonmonotonic logics has only recently gained some attention (Ulbricht, Thimm, and Brewka 2016; Brewka, Thimm, and Ulbricht 2017) and a thorough study is still needed. In this setting, a measure such as \mathcal{I}_{MI} is not applicable as a consistent nonmonotonic knowledge base \mathcal{K} may contain minimal inconsistent subsets. In (Brewka, Thimm, and Ulbricht 2017), a refined notion of

inconsistent subsets of a knowledge base \mathcal{K} of a possibly nonmonotonic framework has been introduced, called *strong \mathcal{K} -inconsistency*. The notion of strong inconsistency generalizes classical inconsistency in a well-behaved manner as it preserves many structural properties as e. g. the hitting set duality with maximal consistent sets (Reiter 1987). Moreover, this notion allows us to generalize existing inconsistency measures based on minimal inconsistent sets to arbitrary logics, which is the topic of the present paper.

Research in inconsistency measurement is driven by *rationality postulates*, i. e., desirable properties that should hold for concrete approaches. There is a growing number of rationality postulates for inconsistency measurement but not every postulate is generally accepted, see (Besnard 2014) for a discussion on this topic. The issue of measuring inconsistency in nonmonotonic frameworks requires some reconsideration compared to the classical setting. This becomes apparent when considering the *monotony* postulate which is usually satisfied by classical inconsistency measures and demands $\mathcal{I}(\mathcal{K}) \leq \mathcal{I}(\mathcal{K}')$ whenever $\mathcal{K} \subseteq \mathcal{K}'$ holds, i. e., the severity of inconsistency cannot be decreased by adding new information. However, in nonmonotonic frameworks, adding information may *resolve* conflicts. It is thus possible that \mathcal{K} is inconsistent, while \mathcal{K}' is not, so we would expect $\mathcal{I}(\mathcal{K}') < \mathcal{I}(\mathcal{K})$ for any reasonable measure \mathcal{I} .

The main contributions of this paper can be summarized as follows. As the basis of our investigation, we consider generalized versions of three measures based on minimal inconsistent sets (Section 2). In order to assess their behaviour, we develop rationality postulates based on previous ones from the literature. Some of the postulates still make sense for a general, possibly nonmonotonic framework, but most of them require refinements (Section 3). We analyze the measures with respect to the postulates in Section 4 and assess their computational complexity by considering natural decision and function problems following (Thimm and Wallner 2016) in Section 5. We apply our ideas to the problem of measuring inconsistency wrt. given contexts, i. e., of relating values of inconsistency of subsets of a knowledge base to inconsistency of the whole knowledge base (Section 6). Section 8 concludes.¹

¹An extended version of this paper with proofs can be found at http://mthimm.de/misc/utb_aaai18_ext.pdf

2 Measures for Strong Inconsistency

Since we are going to consider arbitrary (nonmonotonic) frameworks, we start by giving an abstract definition of a logic. We do so as in (Brewka, Thimm, and Ulbricht 2017).

Definition 2.1. A logic L is a tuple $L = (\text{WF}, \text{BS}, \text{INC}, \text{ACC})$ where WF is a set of well-formed formulas, BS is a set of belief sets (which itself can be defined arbitrarily), $\text{INC} \subseteq \text{BS}$ is an upward closed² set of inconsistent belief sets, and $\text{ACC} : 2^{\text{WF}} \rightarrow 2^{\text{BS}}$ assigns a collection of belief sets to each subset of WF (belief sets are representations of the knowledge which is implicit in a set of formulas; ACC thus defines the semantics of the logic by assigning beliefs sets to sets of formulas). A *knowledge base* \mathcal{K} of L is a finite subset of WF . A knowledge base \mathcal{K} is called *inconsistent* iff $\text{ACC}(\mathcal{K}) \subseteq \text{INC}$. Let $I(\mathcal{K})$ and $I_{\min}(\mathcal{K})$ denote the inconsistent and \subseteq -minimal inconsistent subsets of \mathcal{K} , respectively.

Note that a knowledge base \mathcal{K} can be inconsistent because it has no belief sets, and consistent even if some (but not all) of its belief sets are in INC . Observe that this definition is general enough to capture established monotonic and nonmonotonic frameworks like propositional logic, logic programming under answer set semantics (Gelfond and Lifschitz 1988), abstract argumentation frameworks (Dung 1995) and so on (Brewka, Thimm, and Ulbricht 2017). We show how to cast logic programs under the answer set semantics (Gelfond and Lifschitz 1991; Gelfond and Leone 2002; Brewka, Eiter, and Truszczyński 2011) into our general definition of a logic.

Example 2.2. Let A be a set of propositional atoms and let $\text{Lit}(A) = A \cup \{\neg a \mid a \in A\}$ be the set of corresponding literals. A disjunctive rule r is a rule of the form

$$l_0 \text{ or } \dots \text{ or } l_k \leftarrow l_{k+1}, \dots, l_m, \text{ not } l_{m+1}, \dots, \text{ not } l_n. \quad (1)$$

with $l_0, \dots, l_n \in \text{Lit}(A)$. Let WF_A^{ASP} be the set of all such rules.

A set $P \subseteq \text{WF}_A^{\text{ASP}}$ is also called a logic program for short. Furthermore, a belief set is any set of literals, i.e., $\text{BS}_A^{\text{ASP}} = 2^{\text{Lit}(A)}$. For a program $P \subseteq \text{WF}_A^{\text{ASP}}$ without default negation ($m = n$) an answer set is a minimal set of literals closed under all rules, where a set is closed under a rule of form (1) iff at least one of the head literals l_0, \dots, l_k is in the set whenever the body literals l_{k+1}, \dots, l_m are. For a program P with default negation, a set M of literals is an answer set iff M is an answer set of the reduced program P^M obtained from P by (i) deleting rules with *not* l_j in the body for some $l_j \in M$, and (ii) deleting default negated literals from the remaining rules.

Now let $\text{ACC}_A^{\text{ASP}}(P) = \{M \mid M \text{ is an answer set of } P\}$ for all $P \subseteq \text{WF}_A^{\text{ASP}}$. Finally, inconsistent belief sets are those sets containing complementary literals. This gives us the logic $L_A^{\text{ASP}} = (\text{WF}_A^{\text{ASP}}, \text{BS}_A^{\text{ASP}}, \text{INC}_A^{\text{ASP}}, \text{ACC}_A^{\text{ASP}})$.

Let $\text{WF}_A^{\text{ASP}*} \subseteq \text{WF}_A^{\text{ASP}}$ be the set of all rules of the form (1) with $k = 0$. Then the logic corresponding to disjunction-

free logic programs under answer set semantics is $L_A^{\text{ASP}*} = (\text{WF}_A^{\text{ASP}*}, \text{BS}_A^{\text{ASP}}, \text{INC}_A^{\text{ASP}}, \text{ACC}_A^{\text{ASP}})$.

Definition 2.3. A logic $L = (\text{WF}, \text{BS}, \text{INC}, \text{ACC})$ is monotonic whenever $\mathcal{K} \subseteq \mathcal{K}'$ implies: if $B' \in \text{ACC}(\mathcal{K}')$ then $B \subseteq B'$ for some $B \in \text{ACC}(\mathcal{K})$.

Note that in a monotonic logic conflicts cannot be resolved, i.e., if $\mathcal{K} \subseteq \mathcal{K}'$ holds for two knowledge bases \mathcal{K} and \mathcal{K}' where \mathcal{K} is inconsistent, then \mathcal{K}' is inconsistent as well. We can now define *strong \mathcal{K} -inconsistency* (Brewka, Thimm, and Ulbricht 2017) for our general setting, which is the central notion of this paper.

Definition 2.4. For $H, \mathcal{K} \subseteq \text{WF}$ with $H \subseteq \mathcal{K}$, H is called *strongly \mathcal{K} -inconsistent* if $H \subseteq H' \subseteq \mathcal{K}$ implies H' is inconsistent. The set H is called *strongly inconsistent* if it is strongly WF -inconsistent.

We let $SI(\mathcal{K})$ and $SI_{\min}(\mathcal{K})$ denote the set of all strongly \mathcal{K} -inconsistent and all \subseteq -minimal strongly \mathcal{K} -inconsistent subsets of \mathcal{K} , respectively.

In (Brewka, Thimm, and Ulbricht 2017) it has been shown that the notion of strong inconsistency faithfully generalizes classical inconsistency to arbitrary logics. In particular, the notions coincide for monotonic logics and the existence of a strongly \mathcal{K} -inconsistent subset of \mathcal{K} is a necessary and sufficient condition for inconsistency of \mathcal{K} itself. Moreover, removing from \mathcal{K} any hitting set³ of $SI_{\min}(\mathcal{K})$ yields a maximal consistent subset of \mathcal{K} , which is also known as the *hitting set duality* in classical logics, cf. (Reiter 1987). We refer to (Brewka, Thimm, and Ulbricht 2017) for a more thorough discussion of strong inconsistency.

We now introduce the inconsistency measures we are going to consider throughout this paper. Assume an arbitrary but fixed logic L . In classical inconsistency measurement, minimal inconsistent subsets of a knowledge base play an important role as they can be seen as the “atomic conflicts” within \mathcal{K} . A rather simple but still popular approach to measure inconsistency is thus taking the value $|SI_{\min}(\mathcal{K})|$. The notion of strong \mathcal{K} -inconsistency facilitates the following generalization of this measure to nonmonotonic logics. Let $\mathbb{R}_{\geq 0}^{\infty}$ be the set of non-negative real values including ∞ .

Definition 2.5. Define $\mathcal{I}_{\text{MSI}} : 2^{\text{WF}} \rightarrow \mathbb{R}_{\geq 0}^{\infty}$ via $\mathcal{I}_{\text{MSI}}(\mathcal{K}) = |SI_{\min}(\mathcal{K})|$.

It has been noted that larger minimal inconsistent sets should be considered “less inconsistent” than smaller ones (see e.g. the lottery paradox). Making use of strong \mathcal{K} -inconsistency, we obtain the following measure based on one from (Hunter and Konieczny 2008):

Definition 2.6. Define $\mathcal{I}_{\text{MSI}^c} : 2^{\text{WF}} \rightarrow \mathbb{R}_{\geq 0}^{\infty}$ via $\mathcal{I}_{\text{MSI}^c}(\mathcal{K}) = \sum_{H \in SI_{\min}(\mathcal{K})} \frac{1}{|H|}$.

Instead of counting the number of sets in $SI_{\min}(\mathcal{K})$, one could also take the amount of *formulas* into account. Based on a measure in (Grant and Hunter 2011), we consider the following, quite simple approach:

³A set H is called a *hitting set* of a set of sets $S = \{S_1, \dots, S_n\}$ iff $H \cap S_i \neq \emptyset$ for $i = 1, \dots, n$.

² S upward closed means $B \in S$ and $B \subseteq B'$ implies $B' \in S$.

Definition 2.7. Let $\mathcal{I}_p : 2^{\text{WF}} \rightarrow \mathbb{R}_{\geq 0}^{\infty}$ be the measure $\mathcal{I}_p(\mathcal{K}) = |\bigcup_{H \in SI_{\min}(\mathcal{K})} H|$.

Note that there are further variants in the classical setting for taking minimal inconsistent sets into account for the task of measuring inconsistency, see e. g. (Hunter and Konieczny 2008; Jabbour et al. 2016; Jabbour and Sais 2016). An investigation of generalizations of those is left for future work.

3 Rationality Postulates for General Logics

We are going to revisit rationality postulates for inconsistency measures from the literature and phrase them within the context of an arbitrary, possibly nonmonotonic, logic. We will start by considering the four postulates which a *basic inconsistency measure* should have due to (Hunter and Konieczny 2010). We will then continue our investigation with a collection of other postulates that can be lifted to our general setting.

If not stated otherwise, we assume an arbitrary but fixed logic $L = (\text{WF}, \text{BS}, \text{INC}, \text{ACC})$ (with \emptyset being a consistent knowledge base) and an inconsistency measure $\mathcal{I} : 2^{\text{WF}} \rightarrow \mathbb{R}_{\geq 0}^{\infty}$ for the remainder of this section.

Basic Postulates

The most basic (and undisputed) property that an inconsistency measure should have is the ability to distinguish between consistency and inconsistency, i. e., $\mathcal{I}(\mathcal{K}) = 0$ if and only if \mathcal{K} is consistent. There is no doubt that this makes sense in nonmonotonic frameworks as well.

Consistency For any knowledge base $\mathcal{K} \subseteq \text{WF}$, $\mathcal{I}(\mathcal{K}) = 0$ if and only if \mathcal{K} is consistent.

Another fairly accepted postulate is *monotony*. It formalises the intuition that adding information to a knowledge base should increase the inconsistency value since a knowledge base that contains more information than another one, is also more likely to contain more conflicts.

Monotony If \mathcal{K} and \mathcal{K}' are knowledge bases, then $\mathcal{I}(\mathcal{K}) \leq \mathcal{I}(\mathcal{K} \cup \mathcal{K}')$.

In nonmonotonic frameworks, *monotony* does not make sense because additional information might restore consistency. A simple special case where this does *not* happen is when the knowledge base \mathcal{K} is itself *strongly* inconsistent rather than just inconsistent, motivating a postulate like “ $\mathcal{I}(\mathcal{K}) \leq \mathcal{I}(\mathcal{K} \cup \mathcal{K}')$ if \mathcal{K} is strongly inconsistent”. This would formalize the intuition that adding information cannot resolve *strong* inconsistency. However, it might still be possible that some conflicts in \mathcal{K} are resolved, while others are not. In the most extreme case, all but one conflict in \mathcal{K} are resolved, but \mathcal{K}' contains new ones. In this case, the inconsistency in $\mathcal{K} \cup \mathcal{K}'$ stems more from \mathcal{K}' rather than \mathcal{K} , rendering the comparison between $\mathcal{I}(\mathcal{K})$ and $\mathcal{I}(\mathcal{K} \cup \mathcal{K}')$ quite meaningless. We should thus formalize the concept of monotony more precisely:

Definition 3.1. Let \mathcal{K} and \mathcal{K}' be knowledge bases. We say that \mathcal{K}' *preserves conflicts* of \mathcal{K} if for any $H \in SI(\mathcal{K})$, it also holds that $H \in SI(\mathcal{K} \cup \mathcal{K}')$.

Proposition 3.2. If \mathcal{K} and \mathcal{K}' are knowledge bases and \mathcal{K}' preserves conflicts of \mathcal{K} , then $SI_{\min}(\mathcal{K}) \subseteq SI_{\min}(\mathcal{K} \cup \mathcal{K}')$.

Strong Monotony If \mathcal{K}' preserves conflicts of \mathcal{K} , then $\mathcal{I}(\mathcal{K}) \leq \mathcal{I}(\mathcal{K} \cup \mathcal{K}')$.

We turn to *free formula independence* (Hunter and Konieczny 2010). It states that $\mathcal{I}(\mathcal{K}) = \mathcal{I}(\mathcal{K} \cup \{\alpha\})$ should hold whenever $\alpha \in \mathcal{K}$ is free, i. e., α does not occur in any minimal inconsistent set. A simple adjustment makes use of the observation that we should take strong \mathcal{K} -inconsistency into account: Since a free formula does not introduce inconsistency in the classical case, it makes sense to require that it does not “introduce strong inconsistency” in nonmonotonic logics.

Definition 3.3. Let \mathcal{K} be a knowledge base. A formula $\alpha \in \mathcal{K}$ is called *free with respect to strong \mathcal{K} -inconsistency* (SI-free for short) if $\alpha \notin \bigcup_{H \in SI_{\min}(\mathcal{K})} H$. Let $Free_{SI}(\mathcal{K})$ be the set of all SI-free formulas of \mathcal{K} .

Due to nonmonotonicity, an SI-free formula can still resolve conflicts. Since it cannot introduce them, one might expect the following:

SI-free If $\alpha \in Free_{SI}(\mathcal{K})$, then $\mathcal{I}(\mathcal{K}) \leq \mathcal{I}(\mathcal{K} \setminus \{\alpha\})$.

Note that this postulate is similar to *free formula dilution* (Mu, Liu, and Jin 2011). However, even this property turns out to be quite strong. Not even \mathcal{I}_{MSI} —a measure based on minimal strong \mathcal{K} -inconsistent subsets—satisfies it in general (cf. Section 4). The reason for this is that the set $Free_{SI}(\mathcal{K})$ may affect the structure of $SI_{\min}(\mathcal{K})$. Thus, measures based on $SI_{\min}(\mathcal{K})$ will not tend to satisfy *SI-free* even though the intuition is that formulas in $Free_{SI}(\mathcal{K})$ do not participate in conflicts. This motivates consideration of a stronger notion, ensuring that α can neither introduce nor *resolve* inconsistency:

Definition 3.4. (Brewka, Thimm, and Ulbricht 2017) Let \mathcal{K} be a knowledge base. A formula $\alpha \in \mathcal{K}$ is called *neutral* if any subset $H \subseteq \mathcal{K}$ is consistent if and only if $H \cup \{\alpha\}$ is. We denote the set of all neutral formulas of \mathcal{K} by $Ntr(\mathcal{K})$.

Independence $\mathcal{I}(\mathcal{K}) = \mathcal{I}(\mathcal{K} \setminus \{\alpha\})$ if $\alpha \in Ntr(\mathcal{K})$.

In the propositional setting, the postulate *dominance* (Hunter and Konieczny 2010) requires that α being consistent and $\alpha \models \beta$ implies $\mathcal{I}(\mathcal{K} \cup \{\alpha\}) \geq \mathcal{I}(\mathcal{K} \cup \{\beta\})$. The intuitive meaning is that α carries more information and is thus more likely to be involved in conflicts. Of the postulates we considered so far, it is probably the most disputed one (see e. g. (Besnard 2014)). We are convinced that this postulate is of rather limited use in a nonmonotonic setting. Since adding information to a knowledge base may *resolve* conflicts, there is simply no reason why α , which carries more information than β , should be considered more problematic in general. Even though there are conceivable adjustments that would make *dominance* work for nonmonotonic frameworks, we are not going to discuss them here.

This concludes our discussion on the four postulates for *basic inconsistency measures* (Hunter and Konieczny 2010).

Extended Postulates

We continue our investigation with a generalization of *monotony*, namely *super-additivity* (Thimm 2009). It states that $\mathcal{I}(\mathcal{K}) + \mathcal{I}(\mathcal{K}') \leq \mathcal{I}(\mathcal{K} \cup \mathcal{K}')$ should hold whenever $\mathcal{K} \cap \mathcal{K}' = \emptyset$. As above, we need to take into account that adding information might resolve conflicts in nonmonotonic frameworks. Therefore, we add the additional condition of conflict preservation to our version of *super-additivity*.

Strong Super-Additivity If \mathcal{K}' and \mathcal{K} preserve each other's conflicts and $\mathcal{K} \cap \mathcal{K}' = \emptyset$, then $\mathcal{I}(\mathcal{K}) + \mathcal{I}(\mathcal{K}') \leq \mathcal{I}(\mathcal{K} \cup \mathcal{K}')$.

Many concrete approaches to inconsistency measurement depend on the syntax of a knowledge base. The most common example is the difference between the conjunction $\{a \wedge b\}$ and two formulas $\{a, b\}$. The postulate *adjunction invariance* (Besnard 2014) formalizes the idea that there should be no difference, i. e., $\mathcal{I}(\mathcal{K} \cup \{a \wedge b\}) = \mathcal{I}(\mathcal{K} \cup \{a, b\})$. In the classical setting, there are a couple of more postulates considering situations where (parts of) semantically equivalent knowledge bases are compared, cf. (Thimm 2017).

In nonmonotonic frameworks, a notion of equivalence of the form “ \mathcal{K} has the same models as \mathcal{K}' ” is too weak as conclusions can be withdrawn due to nonmonotonicity. In the context of logic programming, for example, this observation has led to the consideration of so called *strong equivalence* (Lifschitz, Pearce, and Valverde 2001). It is straightforward to generalize this notion to arbitrary nonmonotonic frameworks (Brewka, Thimm, and Ulbricht 2017).

Definition 3.5. Let $L = (\text{WF}, \text{BS}, \text{INC}, \text{ACC})$ be a logic. The knowledge bases \mathcal{K} and \mathcal{K}' are *strongly equivalent*, denoted by $\mathcal{K} \equiv_s \mathcal{K}'$, iff $\text{ACC}(\mathcal{K} \cup H) = \text{ACC}(\mathcal{K}' \cup H)$ for each $H \subseteq \text{WF}$.

Proposition 3.6. (Brewka, Thimm, and Ulbricht 2017) Let \mathcal{K} , \mathcal{K}' and H be knowledge bases. If \mathcal{K} and \mathcal{K}' are strongly equivalent, then \mathcal{K} is strongly $\mathcal{K} \cup H$ -inconsistent iff \mathcal{K}' is strongly $\mathcal{K}' \cup H$ -inconsistent.

Strong equivalence plays a similar role in nonmonotonic frameworks as (normal) equivalence in monotonic frameworks. In particular, it allows for *modularisation* of knowledge bases. If a subset S of a knowledge base \mathcal{K} is strongly equivalent to a set S' then S can be replaced in \mathcal{K} by S' without changing the inferences one can draw from \mathcal{K} . This also means that S and S' should be interchangeably when it comes to the inconsistency they contribute to \mathcal{K} . By generalising this idea to the whole knowledge base, we obtain the desirable property that strongly equivalent knowledge bases should have the same degree of inconsistency.

Strong Equivalence If $\mathcal{K} \equiv_s \mathcal{K}'$ then $\mathcal{I}(\mathcal{K}) = \mathcal{I}(\mathcal{K}')$.

Example 3.7. Consider the logic programs P and P' given as follows.

$$\begin{array}{lll} P : & a. & b. & c \leftarrow a, b, \text{not } c. \\ P' : & a. & b \leftarrow a. & c \leftarrow a, b, \text{not } c. \end{array}$$

Both programs are inconsistent due to not having an answer set. However, the observation $P \equiv_s P'$ is still meaningful as both programs can be rendered consistent by adding rules

which facilitate derivation of c . Moreover, in both cases inconsistency stems from the derivation of a and b (in both cases without making use of defaults) combined with the rule “ $c \leftarrow a, b, \text{not } c$.”. It is thus reasonable to expect $\mathcal{I}(P) = \mathcal{I}(P')$.

However, whether or not this is meaningful depends on the framework under consideration. Sometimes, it might not be helpful to *distinguish* between inconsistent knowledge bases. For example, in classical logics we have $\mathcal{K} \equiv_s \mathcal{K}'$ for any two inconsistent knowledge bases \mathcal{K} and \mathcal{K}' , thus rendering the above postulates too strong.

In order to obtain a more fine grained notion of equivalence, we utilize a technique from (Thimm 2013a) for the postulate “irrelevance of syntax”. For our setting we define:

Definition 3.8. Let \mathcal{K} and \mathcal{K}' be two knowledge bases. We call \mathcal{K} and \mathcal{K}' *formula-wise strongly equivalent*, denoted by $\mathcal{K} \equiv_\alpha \mathcal{K}'$, if there is a bijection $\rho : \mathcal{K} \rightarrow \mathcal{K}'$ such that $\{\alpha\} \equiv_s \{\rho(\alpha)\}$ holds for all $\alpha \in \mathcal{K}$.

We obtain the following generalization of *irrelevance of syntax* (Thimm 2013a) (FW=formula-wise):

FW-Strong Equivalence If $\mathcal{K} \equiv_\alpha \mathcal{K}'$ then $\mathcal{I}(\mathcal{K}) = \mathcal{I}(\mathcal{K}')$.

The final postulate we consider in this subsection is *separability* (Hunter and Konieczny 2010) which has a straightforward representation in our general context.

Separability If $SI_{\min}(\mathcal{K} \cup \mathcal{K}') = SI_{\min}(\mathcal{K}) \cup SI_{\min}(\mathcal{K}')$ and $SI_{\min}(\mathcal{K}) \cap SI_{\min}(\mathcal{K}') = \emptyset$ then $\mathcal{I}(\mathcal{K} \cup \mathcal{K}') = \mathcal{I}(\mathcal{K}) + \mathcal{I}(\mathcal{K}')$.

In other words, if the conflicts of two knowledge bases \mathcal{K} and \mathcal{K}' are completely independent, the inconsistency value of their union should be decomposed as the sum of the individual values.

4 Analysis

Now we investigate the compliance of the considered measures with the rationality postulates above. For postulates that are not satisfied by a particular measure in general, we give counterexamples within the logic L_A^{ASP} .

Regarding *consistency*, note that $SI_{\min}(\mathcal{K}) = \emptyset$ if and only if \mathcal{K} is consistent. Hence, the postulate is satisfied by all measures. In general, we obtain the following result on the compliance of our measures with the rationality postulates, see also Table 1.

Proposition 4.1. \mathcal{I}_{MSI} , $\mathcal{I}_{\text{MSI}^c}$ and \mathcal{I}_p satisfy consistency, strong monotony, independence and strong super-additivity, and FW-strong equivalence. \mathcal{I}_{MSI} and $\mathcal{I}_{\text{MSI}^c}$ also satisfy separability.

As already mentioned in Section 3, the postulate *SI-free* is not satisfied by any of the measures.

Example 4.2. Consider the program P given as follows:

$$P : a \leftarrow \text{not } a, b. \quad a \leftarrow \text{not } c, \text{not } d. \quad b. \quad c. \quad d.$$

The rule $r = a \leftarrow \text{not } c, \text{not } d$. is in $\text{Free}_{SI}(P)$: the rule “ $a \leftarrow \text{not } a, b$.” combined with the fact “ b .” is responsible for P being inconsistent and r cannot restore consistency as long as “ c .” or “ d .” are present. Hence, $SI_{\min}(P)$ consists of $\{a \leftarrow \text{not } a, b., b., c.\}$ and $\{a \leftarrow \text{not } a, b., b., d.\}$, i. e.,

	\mathcal{I}_{MSI}	$\mathcal{I}_{\text{MSIC}}$	\mathcal{I}_{p}
Consistency	✓	✓	✓
Strong Monotony	✓	✓	✓
SI-free	✗	✗	✗
Independence	✓	✓	✓
Strong Super-Additivity	✓	✓	✓
Strong Equivalence	✗	✗	✗
FW-Strong Equivalence	✓	✓	✓
Separability	✓	✓	✗

Table 1: Compliance of measures with rationality postulates

$\mathcal{I}_{\text{MSI}}(P) = 2$, $\mathcal{I}_{\text{MSIC}}(P) = \frac{2}{3}$ and $\mathcal{I}_{\text{p}}(P) = 4$. However $SI_{\min}(P \setminus \{r\}) = \{a \leftarrow \text{not } a, b., b.\}$, i. e., $\mathcal{I}_{\text{MSI}}(P \setminus \{r\}) = 1$, $\mathcal{I}_{\text{MSIC}}(P \setminus \{r\}) = \frac{1}{2}$ and $\mathcal{I}_{\text{p}}(P \setminus \{r\}) = 2$.

Example 4.3. Consider the programs P and P' given via

$$P : a. \quad \neg a. \quad P' : a. \quad \neg a. \quad a \leftarrow \neg a. \quad \neg a \leftarrow a.$$

It is easy to see that $P \equiv_s P'$ as the inconsistency in both programs cannot be repaired in any extension of them. However, we have that $\mathcal{I}(P_1) \neq \mathcal{I}(P_2)$ for all $\mathcal{I} \in \{\mathcal{I}_{\text{MSI}}, \mathcal{I}_{\text{MSIC}}, \mathcal{I}_{\text{p}}\}$ thus showing that *strong equivalence* is violated by all three measures.

For a counterexample of *separability* wrt. \mathcal{I}_{p} see (Thimm 2017) (already in the classical case). Observe that for those postulates that are generalizations of classical ones—i. e., *consistency*, *strong monotony*, *independence*, *strong super-additivity*, and *separability*—the compliance of our three measures generalizes their compliance with the corresponding postulates in the classical case, cf. (Thimm 2017).

5 Computational Complexity

We now address the computational complexity of the measures we considered so far. Following (Thimm and Wallner 2016), we consider the three decision problems $\text{EXACT}_{\mathcal{I}}$, $\text{UPPER}_{\mathcal{I}}$, $\text{LOWER}_{\mathcal{I}}$, and the natural function problem $\text{VALUE}_{\mathcal{I}}$. Let $L = (\text{WF}, \text{ACC}, \text{BS}, \text{INC})$ be a logic.

$\text{EXACT}_{\mathcal{I}}^L$	Input: $\mathcal{K} \subseteq \text{WF}$, $x \in [0, \infty]$ Output: TRUE iff $\mathcal{I}(\mathcal{K}) = x$
$\text{UPPER}_{\mathcal{I}}^L$	Input: $\mathcal{K} \subseteq \text{WF}$, $x \in [0, \infty]$ Output: TRUE iff $\mathcal{I}(\mathcal{K}) \leq x$
$\text{LOWER}_{\mathcal{I}}^L$	Input: $\mathcal{K} \subseteq \text{WF}$, $x \in (0, \infty]$ Output: TRUE iff $\mathcal{I}(\mathcal{K}) \geq x$
$\text{VALUE}_{\mathcal{I}}^L$	Input: $\mathcal{K} \subseteq \text{WF}$ Output: The value of $\mathcal{I}(\mathcal{K})$

We assume the reader to be familiar with the polynomial hierarchy, i. e., the classes Σ_m^p and Π_m^p for $m \geq 0$. We also consider the class D_m^p that consists of all languages that are the intersection of a language in Σ_m^p and a language in Π_m^p . Finally, we make use of the counting polynomial hierarchy (Wagner 1986), which provides classes for decision problems involving subproblems such as counting strongly minimal inconsistent sets (represented by a prefixing “C” before the standard complexity class names).

First, we consider an arbitrary, possibly nonmonotonic logic $L = (\text{WF}, \text{ACC}, \text{BS}, \text{INC})$. Since hardness results cannot be expected in general (these depend on the concrete logic), we will only give membership statements here. Our results will depend on the complexity of the satisfiability check of L .

SAT_L **Input:** $\mathcal{K} \subseteq \text{WF}$
 Output: TRUE iff \mathcal{K} is consistent

Depending on SAT_L , we obtain the following membership results for \mathcal{I}_{MSI} and \mathcal{I}_{p} :

Theorem 5.1. *Let $m \geq 1$. If the problem SAT_L is in*

- Σ_m^p , then $\text{UPPER}_{\mathcal{I}_{\text{MSI}}}^L$ and $\text{LOWER}_{\mathcal{I}_{\text{MSI}}}^L$ are in $\text{C}\Sigma_m^p$,
- Π_m^p , then $\text{UPPER}_{\mathcal{I}_{\text{MSI}}}^L$ and $\text{LOWER}_{\mathcal{I}_{\text{MSI}}}^L$ are in $\text{C}\Sigma_{m+1}^p$,
- Π_m^p and L is monotonic, then $\text{UPPER}_{\mathcal{I}_{\text{MSI}}}^L$ and $\text{LOWER}_{\mathcal{I}_{\text{MSI}}}^L$ are in $\text{C}\Sigma_m^p$.

Note that no counting class occurs in the following theorem as the decision problems for \mathcal{I}_{p} are much easier.

Theorem 5.2. *Let $m \geq 1$. If the problem SAT_L is in*

- Σ_m^p , then $\text{LOWER}_{\mathcal{I}_{\text{p}}}^L$ is in Σ_{m+1}^p ,
- Π_m^p , then $\text{LOWER}_{\mathcal{I}_{\text{p}}}^L$ is in Σ_{m+2}^p ,
- Π_m^p and L is monotonic, then $\text{LOWER}_{\mathcal{I}_{\text{p}}}^L$ is in Σ_{m+1}^p .

Using the latter theorem and techniques from (Thimm and Wallner 2016), Lemmas 2 and 3, we infer:

Corollary 5.3. *Let $m \geq 1$. If the problem SAT_L is in*

- Σ_m^p , then $\text{UPPER}_{\mathcal{I}_{\text{p}}}^L$ is in Π_{m+1}^p , $\text{EXACT}_{\mathcal{I}_{\text{p}}}^L$ is in D_{m+1}^p and $\text{VALUE}_{\mathcal{I}_{\text{p}}}^L$ is in $\text{FP}_{m+1}^{\Sigma_m^p}[\log n]$,
- Π_m^p , then $\text{UPPER}_{\mathcal{I}_{\text{p}}}^L$ is in Π_{m+2}^p , $\text{EXACT}_{\mathcal{I}_{\text{p}}}^L$ is in D_{m+2}^p and $\text{VALUE}_{\mathcal{I}_{\text{p}}}^L$ is in $\text{FP}_{m+2}^{\Sigma_m^p}[\log n]$,
- Π_m^p and L is monotonic, then $\text{UPPER}_{\mathcal{I}_{\text{p}}}^L$ is in Π_{m+1}^p , $\text{EXACT}_{\mathcal{I}_{\text{p}}}^L$ is in D_{m+1}^p and $\text{VALUE}_{\mathcal{I}_{\text{p}}}^L$ is in $\text{FP}_{m+1}^{\Sigma_m^p}[\log n]$.

Furthermore, Table 2 summarises some additional results pertaining to hardness wrt. the concrete logics $L_A^{\text{ASP}^*}$ and L_A^{ASP} (disjunction-free and disjunctive logic programs under the answer set semantics, respectively). As with the other results, proofs can be found in the online appendix. Recall that deciding whether a program P within the framework $L_A^{\text{ASP}^*}$ is consistent is NP-complete in general, while the decision problem for programs P in L_A^{ASP} is Σ_2^p -complete (Eiter and Gottlob 1995). The results show that, for $L_A^{\text{ASP}^*}$, the computational complexity of the problems we consider is similar to the results for the propositional case given in (Thimm and Wallner 2016). This seems natural as the satisfiability check for propositional logic is NP-complete as well. As expected, considering L_A^{ASP} involves going up one level within the corresponding hierarchy.

Finally, we have the following results pertaining to $\text{VALUE}_{\mathcal{I}_{\text{MSI}}}$, cf. (Valiant 1979) for the definition of the used counting complexity classes.

Theorem 5.4. *The problem $\text{VALUE}_{\mathcal{I}_{\text{MSI}}}^{L_A^{\text{ASP}^*}}$ is #coNP-complete under subtractive reductions. The problem $\text{VALUE}_{\mathcal{I}_{\text{MSI}}}^{L_A^{\text{ASP}}}$ is # Π_2^p -complete under subtractive reductions.*

		$\overline{\mathcal{I}}_{\text{MSI}}$	$\overline{\mathcal{I}}_{\text{MSIC}}$	$\overline{\mathcal{I}}_{\text{p}}$
	UPPER $_{\mathcal{I}}^{L_A^{\text{ASP}^*}}$	CNP-c	CNP-h	$\Sigma_2^p\text{-c}$
	LOWER $_{\mathcal{I}}^{L_A^{\text{ASP}^*}}$	CNP-c	CNP-h	$\Pi_2^p\text{-c}$
	EXACT $_{\mathcal{I}}^{L_A^{\text{ASP}^*}}$	C=NP-h	C=NP-h	$D_2^p\text{-c}$
	UPPER $_{\mathcal{I}}^{L_A^{\text{ASP}}}$	$C\Sigma_2^p\text{-c}$	$C\Sigma_2^p\text{-h}$	$\Sigma_3^p\text{-c}$
	LOWER $_{\mathcal{I}}^{L_A^{\text{ASP}}}$	$C\Sigma_2^p\text{-c}$	$C\Sigma_2^p\text{-h}$	$\Pi_3^p\text{-c}$
	EXACT $_{\mathcal{I}}^{L_A^{\text{ASP}}}$	$C=\Sigma_2^p\text{-h}$	$C=\Sigma_2^p\text{-h}$	$D_3^p\text{-c}$

Table 2: Hardness results for $L_A^{\text{ASP}^*}$ and L_A^{ASP}

6 Strong Inconsistency and Context

As already taken into account by Definition 2.4, the central issue regarding inconsistency in nonmonotonic logics is that conflicts may be resolved by adding information. This raises the question how, given an inconsistent knowledge base \mathcal{K} , the degree of inconsistency of a subset $H \subseteq \mathcal{K}$ should be assessed. We give a motivating example within the logic L_A^{ASP} .

Example 6.1. Consider the program P given as follows:

$$P : a \text{ or } b. \quad d \leftarrow \text{not } a, \text{ not } d. \quad d \leftarrow \text{not } b, \text{ not } d.$$

Inconsistency of P stems from the two rules “ $d \leftarrow \text{not } a, \text{ not } d.$ ” and “ $d \leftarrow \text{not } b, \text{ not } d.$ ”. As answer sets are required to be minimal models, an answer set can only contain *either* a or b , but not both. Yet, the subset $H = \{\leftarrow \text{not } a., \leftarrow \text{not } b.\}$ obviously consists of *two* conflicts and this intuition is confirmed by the observation that $\overline{\mathcal{I}}_{\text{MSI}}(H) = 2$ holds.

The preceding example makes use of the observation that a set $H \in SI_{\text{min}}(\mathcal{K})$ may itself contain more than one inconsistent subset. In fact, as the hitting set duality utilizing strong \mathcal{K} -inconsistency (Brewka, Thimm, and Ulbricht 2017) suggests, inconsistency of a subset $H \subseteq \mathcal{K}$ is only meaningful *within the context* of the knowledge base \mathcal{K} . Postulates describing the behaviour of subsets of knowledge bases as well as single formulas should respect this observation. This motivates the following notion.

Definition 6.2. Let $\mathcal{I} : 2^{\text{WF}} \rightarrow \mathbb{R}_{\geq 0}^{\infty}$ be an inconsistency measure, \mathcal{K} a knowledge base and $H \subseteq \mathcal{K}$. We call

$$\text{Co}_{\mathcal{K}, \mathcal{I}}(H) := \min_{H \subseteq H' \subseteq \mathcal{K}} \mathcal{I}(H') \quad (2)$$

the value of $\mathcal{I}(H)$ *with respect to the context* \mathcal{K} .

Taking the minimum in (2) has the same motivation as considering all supersets of H in Definition 2.4: We are only interested in (the severity of) conflicts that cannot be resolved within \mathcal{K} . For example, if we are given $H \subseteq \mathcal{K}$ and a measure \mathcal{I} satisfying *consistency*, then $\text{Co}_{\mathcal{K}, \mathcal{I}}(H) = 0$ iff H is *not* strongly \mathcal{K} -inconsistent. Moreover, as most classical inconsistency measures for propositional knowledge bases satisfy *monotony* (cf. (Thimm 2017)), $\text{Co}_{\mathcal{K}, \mathcal{I}}(H) = \mathcal{I}(H)$ oftentimes holds in the classical case, anyway.

Hence, when assessing the severity of inconsistency of *subsets* of a knowledge base \mathcal{K} , utilizing $\text{Co}_{\mathcal{K}, \mathcal{I}}(H)$ appears to be appropriate. Moreover, many of the postulates given above are preserved by $\text{Co}_{\mathcal{K}, \mathcal{I}}(H)$:

Proposition 6.3. Let \mathcal{K} be a knowledge base and $H, H' \subseteq \mathcal{K}$. It holds that

1. $\text{Co}_{\mathcal{K}, \mathcal{I}}(H) \leq \text{Co}_{\mathcal{K}, \mathcal{I}}(H \cup H')$,
2. if \mathcal{I} satisfies consistency, then $\text{Co}_{\mathcal{K}, \mathcal{I}}(H) = 0$ if and only if $H \notin SI(\mathcal{K})$,
3. if \mathcal{I} satisfies independence and $\alpha \in \text{Ntr}(\mathcal{K})$, then $\text{Co}_{\mathcal{K}, \mathcal{I}}(H) = \text{Co}_{\mathcal{K}, \mathcal{I}}(H \setminus \{\alpha\})$,
4. if \mathcal{I} satisfies strong equivalence and $H \equiv_s H'$ for $H, H' \subseteq \mathcal{K}$, then $\text{Co}_{\mathcal{K}, \mathcal{I}}(H) = \text{Co}_{\mathcal{K}, \mathcal{I}}(H')$.

Note that $\text{Co}_{\mathcal{K}, \mathcal{I}}(H) \leq \text{Co}_{\mathcal{K}, \mathcal{I}}(H \cup H')$ holds without the notion of *preserving conflicts*. Since we calculate the minimum in (2), nonmonotonicity is already taken into account.

Now consider the following postulate adapted from (Hunter and Konieczny 2010):

MSI-Normalization If $H \in SI_{\text{min}}(\mathcal{K})$, then $\mathcal{I}(H) = 1$.

MSI-Normalization reflects the intuition that a subset $H \in SI_{\text{min}}(\mathcal{K})$ should correspond to an “atomic” conflict within \mathcal{K} . However, this is not done by $\text{Co}_{\mathcal{K}, \mathcal{I}}(H)$, even for the special case $\mathcal{I} = \overline{\mathcal{I}}_{\text{MSI}}$.

Example 6.4. Let $\mathcal{K} = \{\alpha_1, \dots, \alpha_4\}$ be a knowledge base such that only $\{\alpha_1, \alpha_3\}$ and $\{\alpha_2, \alpha_4\}$ are consistent subsets. Let $H = \{\alpha_1, \alpha_2\}$. One can verify that $\text{Co}_{\mathcal{K}, \overline{\mathcal{I}}_{\text{MSI}}}(H) = 2$.

The reason is, roughly speaking, that a set $H \in SI_{\text{min}}(\mathcal{K})$ does not represent “one conflict”, but rather “one conflict that cannot be resolved”. Thus, to capture atomic conflicts in a nonmonotonic logic, we need to utilize a notion of “resolvable conflicts”. This induces the following novel measure.

Definition 6.5. Let \mathcal{K} be a knowledge base. Let $H \subseteq \mathcal{K}$ and consider $\{H_1, \dots, H_n\} = SI_{\text{min}}(H)$, i. e., the minimal conflicts of H . Let $\overline{H} = H \setminus SI_{\text{min}}(H)$. Define $\mathcal{I}_{\text{Res}(\mathcal{K})}(H)$ as the number of conflicts within H that *cannot* be resolved within \mathcal{K} , i. e.,

$$\mathcal{I}_{\text{Res}(\mathcal{K})}(H) = \min_{I \subseteq \{1, \dots, n\}} \{n - |I| \mid \bigcup_{i \in I} H_i \cup \overline{H} \notin SI(\mathcal{K})\}.$$

Example 6.6. Consider P from above again:

$$P : a \text{ or } b. \quad \leftarrow \text{not } a. \quad \leftarrow \text{not } b.$$

Again, let $H = P \setminus \{a \text{ or } b.\}$. We have $SI_{\text{min}}(H) = \{\{\leftarrow \text{not } a.\}, \{\leftarrow \text{not } b.\}\}$ and $\overline{H} = \emptyset$. Let $H_1 = \{\leftarrow \text{not } a.\}$ and $H_2 = \{\leftarrow \text{not } b.\}$. Then,

$$H_i \notin SI(P), i = 1, 2, \quad H_1 \cup H_2 \in SI(P).$$

So, “ a or $b.$ ” can “resolve one conflict”. As desired, $\mathcal{I}_{\text{Res}(P)}(H) = 1$. The next example shall illustrate why we need \overline{H} in the definition of $\mathcal{I}_{\text{Res}(P)}(H)$. Consider

$$P' : \neg a. \quad \leftarrow \text{not } a. \quad a.$$

with H' being the subset $H' = \{\neg a., \leftarrow \text{not } a.\}$. We have $SI_{\text{min}}(H') = \{\{\leftarrow \text{not } a.\}\}$ and hence, within H' the fact “ $\neg a.$ ” is not yet identified as potential conflict. However, as satisfaction of the constraint “ $\leftarrow \text{not } a.$ ” requires “ $a.$ ”, the negated fact “ $\neg a.$ ” should be taken into account. Thus, even though the constraint—being the only rule in $SI_{\text{min}}(H')$ —can be resolved, $\mathcal{I}_{\text{Res}(P')}(H') = 1$. This is as desired, because H' carries two problems of which only one can be resolved; not both simultaneously. Moreover, $H' \in SI(P')$ suggests that a measure should not assign 0 to it.

Since conflicts cannot be resolved by adding information in monotonic logics, $\mathcal{I}_{\text{Res}(\mathcal{K})}(H)$ and $\mathcal{I}_{\text{MI}}(H)$ should coincide to capture our intuition of minimal inconsistent sets as “atomic conflicts”. Indeed:

Proposition 6.7. *If L is a monotonic logic and \mathcal{K} a knowledge base with $H \subseteq \mathcal{K}$, then $\mathcal{I}_{\text{Res}(\mathcal{K})}(H) = \mathcal{I}_{\text{MI}}(H)$.*

Moreover, it complies with some of the ideas above, e. g.:

Proposition 6.8. *Let \mathcal{K} be a knowledge base and $H, H' \subseteq \mathcal{K}$. It holds that*

- $\mathcal{I}_{\text{Res}(\mathcal{K})}(H) = 0$ if and only if $H \notin \text{SI}(\mathcal{K})$,
- $\mathcal{I}_{\text{Res}(\mathcal{K})}(H) \leq \mathcal{I}_{\text{Res}(\mathcal{K})}(H \cup H')$,
- $\mathcal{I}_{\text{Res}(\mathcal{K})}(H) = \mathcal{I}_{\text{Res}(\mathcal{K})}(H \setminus \{\alpha\})$ if $\alpha \in \text{Ntr}(\mathcal{K})$.

In particular, $\mathcal{I}_{\text{Res}(\mathcal{K})}(H)$ captures the intuition that a set $H \in \text{SI}_{\text{min}}(\mathcal{K})$ represents exactly one conflict.

Proposition 6.9. *Let \mathcal{K} be a knowledge base and $H \in \text{SI}_{\text{min}}(\mathcal{K})$. Then $\mathcal{I}_{\text{Res}(\mathcal{K})}(H) = 1$.*

7 Related Work

Inconsistency measurement in non-classical frameworks has been addressed in some limited fashion before (Thimm 2013b; Potyka 2014; De Bona and Finger 2015; Condotta, Raddaoui, and Salhi 2016; Ulbricht, Thimm, and Brewka 2016; Amgoud and Ben-Naim 2017). The latter paper studies disagreement in argumentation graphs, a notion slightly different from inconsistency. It will nevertheless be interesting to see whether postulates for disagreement are applicable to inconsistency as well. But the closest work to this one is (Ulbricht, Thimm, and Brewka 2016) where the special case of ASP is considered rather than a general, nonmonotonic logic. We briefly discuss some of the postulates mentioned. Many of them are concerned about situations where some kind of *monotony* should hold. However, all of them do have in common that conflicts are preserved as in *strong monotony* from above. In fact, they turn out to be special cases. Before we state them, we need the notion of a splitting set of a program P .

Definition 7.1. Let P be a logic program, i. e., a set of rules of the form (1). A set U of literals is called a *splitting set* for P , if $\{l_0, \dots, l_k\} \cap U \neq \emptyset$ implies $\{l_0, \dots, l_n\} \subseteq U$ for any rule $r \in P$. For a splitting set U , let $\text{bot}_U(P)$ be the set of all rules $r \in P$ with $\{l_0, \dots, l_n\} \subseteq U$.

Now we consider the following postulates from (Ulbricht, Thimm, and Brewka 2016).

CLP-Monotonicity If P does not contain default negation “not”, then $\mathcal{I}(P) \leq \mathcal{I}(P \cup P')$ for any program P' .

Split-Monotonicity If U is a splitting set for P , then $\mathcal{I}(\text{bot}_U(P)) \leq \mathcal{I}(P)$.

Con-Monotonicity If r is a constraint⁴, then $\mathcal{I}(P) \leq \mathcal{I}(P \cup \{r\})$ for any $P \in \mathcal{P}$.

All of them describe situations where the additional rules preserves conflicts of the program on the left hand side.

⁴A rule “ $a \leftarrow l_{k+1}, \dots, l_m, \text{not } l_{m+1}, \dots, \text{not } l_n, \text{not } a.$ ” where a does not occur elsewhere in P is called *constraint*.

Proposition 7.2. *If a measure \mathcal{I} satisfies strong monotony, it satisfies CLP-Monotony, Split-Monotony and Con-Monotony as well.*

Some postulates in the literature explicitly make use of the language resp. atoms occurring in a formula or knowledge base. It is thus hard to phrase them for a general logic L . However, it is done for answer set programming. Recall *SI-free* and *independence* from above. Both are generalizations of *free formula independence*. A weaker version is *safe rule independence* (Hunter and Konieczny 2010). It states that a consistent formula α should not change the inconsistency value of a knowledge base \mathcal{K} if no atom in \mathcal{K} occurs in α . Within answer set programming, we can define the notion of a safe rule more liberally as only the head atoms of a rule are to be taken into account (Ulbricht, Thimm, and Brewka 2016).

Definition 7.3. Let P be a disjunctive logic program. A rule r such that $\{r\}$ is consistent is called *safe* with respect to P if the atoms occurring in the head of r do not occur in $P \setminus \{r\}$.

This induces the following postulate:

Safe-rule independence If P is a logic program and r^* safe with respect to P , then $\mathcal{I}(P) = \mathcal{I}(P \cup \{r^*\})$.

More generally, the inconsistency values of two independent programs should add up.

Language Separability If P and P' are programs that do not share any atoms, then $\mathcal{I}(P \cup P') = \mathcal{I}(P) + \mathcal{I}(P')$.

Note that this postulate is similar to *separability* from above, without explicitly making use of $\text{SI}_{\text{min}}(P)$ and $\text{SI}_{\text{min}}(P')$.

8 Conclusions

We made first steps towards measuring inconsistency in a general, possibly nonmonotonic, framework by revisiting rationality postulates for propositional logic and adjusting them for our setting. Utilizing those postulates, we examined the behaviour of measures based on minimal strong \mathcal{K} -inconsistent subsets, a generalization of minimal inconsistent subsets to nonmonotonic frameworks. We also analyzed their computational complexity. Finally, we re-examined the notion of strong inconsistency in order to define an inconsistency measure on subsets of a context. This measure can therefore be used to analyse the distribution of inconsistency within a knowledge base, similar as the Shapley measure (Hunter and Konieczny 2010) in the classical case.

In the literature, additional measures based on minimal inconsistent sets of a knowledge base \mathcal{K} have been proposed, e. g., (Jabbour et al. 2016; Jabbour and Sais 2016). It is a natural idea to investigate their behavior in nonmonotonic frameworks, making use of strong \mathcal{K} -inconsistency. Moreover, focusing on particular frameworks similar in spirit to (Ulbricht, Thimm, and Brewka 2016) may lead to the development of more significant rationality postulates, as they can be tailored for the framework.

Acknowledgements

This work was partially funded by DFG (Research Training Group 1763; project BR 1817/7-2).

References

- Amgoud, L., and Ben-Naim, J. 2017. Measuring disagreements in argumentation graphs. In *Proceedings of the 11th International Conference on Scalable Uncertainty Management (SUM'17)*.
- Besnard, P. 2014. Revisiting postulates for inconsistency measures. In *Proceedings of the 14th European Conference on Logics in Artificial Intelligence (JELIA'14)*, 383–396.
- Brewka, G.; Eiter, T.; and Truszczynski, M. 2011. Answer set programming at a glance. *Commun. ACM* 54(12):92–103.
- Brewka, G.; Thimm, M.; and Ulbricht, M. 2017. Strong inconsistency in nonmonotonic reasoning. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, 901–907.
- Condotta, J.-F.; Raddaoui, B.; and Salhi, Y. 2016. Quantifying conflicts for spatial and temporal information. In *Proceedings of the 15th International Conference on Principles of Knowledge Representation and Reasoning (KR'16)*.
- De Bona, G., and Finger, M. 2015. Measuring inconsistency in probabilistic logic: Rationality postulates and dutch book interpretation. *Artificial Intelligence* 227:140–164.
- Dung, P. M. 1995. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intell.* 77(2):321–358.
- Eiter, T., and Gottlob, G. 1995. On the computational cost of disjunctive logic programming: Propositional case. *Annals of Mathematics and Artificial Intelligence* 15(3-4):289–323.
- Gelfond, M., and Leone, N. 2002. Logic programming and knowledge representation – the A-Prolog perspective. *Artificial Intelligence* 138(1–2):3–38.
- Gelfond, M., and Lifschitz, V. 1988. The stable model semantics for logic programming. In *ICLP/SLP*, volume 88, 1070–1080.
- Gelfond, M., and Lifschitz, V. 1991. Classical negation in logic programs and disjunctive databases. *New Generation Comput.* 9(3/4):365–386.
- Grant, J., and Hunter, A. 2006. Measuring Inconsistency in Knowledgebases. *Journal of Intelligent Information Systems* 27:159–184.
- Grant, J., and Hunter, A. 2011. Measuring consistency gain and information loss in stepwise inconsistency resolution. *Symbolic and Quantitative Approaches to Reasoning with Uncertainty* 362–373.
- Hunter, A., and Konieczny, S. 2004. Approaches to Measuring Inconsistent Information. In *Inconsistency Tolerance*, volume 3300 of *Lecture Notes in Computer Science*. Springer International Publishing. 189–234.
- Hunter, A., and Konieczny, S. 2008. Measuring inconsistency through minimal inconsistent sets. In *Proceedings of the Eleventh International Conference on Principles of Knowledge Representation and Reasoning (KR'2008)*, 358–366. AAAI Press.
- Hunter, A., and Konieczny, S. 2010. On the measure of conflicts: Shapley inconsistency values. *Artificial Intelligence* 174(14):1007–1026.
- Jabbour, S., and Sais, L. 2016. Exploiting MUS Structure to Measure Inconsistency of Knowledge Bases. In *Proceedings of the 22nd European Conference on Artificial Intelligence (ECAI'16)*.
- Jabbour, S.; Ma, Y.; Raddaoui, B.; Sais, L.; and Salhi, Y. 2016. A MIS Partition Based Framework for Measuring Inconsistency. In *Proceedings of the 15th International Conference on Principles of Knowledge Representation and Reasoning (KR'16)*.
- Lifschitz, V.; Pearce, D.; and Valverde, A. 2001. Strongly equivalent logic programs. *ACM Transactions on Computational Logic* 2(4):526–541.
- Mu, K.; Liu, W.; and Jin, Z. 2011. A general framework for measuring inconsistency through minimal inconsistent sets. *Knowledge and Information Systems* 27(1):85–114.
- Potyka, N. 2014. Linear Programs for Measuring Inconsistency in Probabilistic Logics. In *Proceedings of the 14th International Conference on Principles of Knowledge Representation and Reasoning (KR'14)*, 568–577.
- Reiter, R. 1987. A theory of diagnosis from first principles. *Artif. Intell.* 32(1):57–95.
- Thimm, M., and Wallner, J. P. 2016. Some complexity results on inconsistency measurement. In *Proceedings of the 15th International Conference on Principles of Knowledge Representation and Reasoning (KR'16)*, 114–124.
- Thimm, M. 2009. Measuring inconsistency in probabilistic knowledge bases. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, 530–537. AUAI Press.
- Thimm, M. 2013a. Inconsistency measures for probabilistic logics. *Artificial Intelligence* 197:1–24.
- Thimm, M. 2013b. Inconsistency Measures for Probabilistic Logics. *Artificial Intelligence* 197:1–24.
- Thimm, M. 2016. On the expressivity of inconsistency measures. *Artificial Intelligence* 234:120–151.
- Thimm, M. 2017. On the compliance of rationality postulates for inconsistency measures: A more or less complete picture. *KI-Künstliche Intelligenz* 31(1):31–39.
- Ulbricht, M.; Thimm, M.; and Brewka, G. 2016. Measuring Inconsistency in Answer Set Programs. In *Proceedings of the 15th European Conference on Logics in Artificial Intelligence (JELIA'16)*.
- Valiant, L. G. 1979. The complexity of computing the permanent. *Theoretical computer science* 8(2):189–201.
- Wagner, K. W. 1986. The complexity of combinatorial problems with succinct input representation. *Acta informatica* 23(3):325–356.