Handling and Measuring Inconsistency in Non-monotonic Logics

Markus Ulbricht[1] and Matthias Thimm[2] and Gerhard Brewka[1]

[1] *Department of Computer Science, Leipzig University, Germany*
[2] *Institute for Web Science and Technologies, University of Koblenz-Landau, Germany*

Abstract

We address the issue of quantitatively assessing the severity of inconsistencies in non-monotonic frameworks. While measuring inconsistency in classical logics has been investigated for some time now, taking the non-monotonicity 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 sound generalization of minimal inconsistent subsets to arbitrary, possibly non-monotonic, frameworks which induces a generalization of Reiter's famous hitting set duality between minimal inconsistent and maximal consistent subsets of a knowledge base. We propose measures based on this notion and investigate their behavior in a non-monotonic setting by revisiting existing rationality postulates, analyzing the compliance of the proposed measures with these postulates, and by investigating their computational complexity. Motivated by the observation that a knowledge base of a non-monotonic logic can also be repaired by *adding* formulas –whereas Reiter's duality is only concerned about *removing*–, we also investigate situations where we are given potential additional assumptions to repair a knowledge base. For this, we characterize the minimal modifications to a knowledge base in terms of a hitting set duality.

Keywords: non-monotonic reasoning, inconsistency handling, inconsistency measurement

1. Introduction

In applications such as decision-support systems, a knowledge base is usually compiled by merging the formalised knowledge of many different experts.

It is unavoidable that different experts contradict each other and that the merged knowledge base would become inconsistent. The field of *Knowledge Representation and Reasoning* (KR) is the subfield of *Artificial Intelligence* (AI) that deals with the issues of logical formalisations of information and the modelling of rational reasoning behavior, in particular in light of inconsistent or uncertain information. One paradigm to deal with inconsistent information is to abandon classical inference and define new ways of reasoning. Some examples of such formalisms are, e. g., paraconsistent logics (Béziau et al., 2007), default logic (Reiter, 1980), answer set programming (Gelfond and Leone, 2002), and, more recently, computational models of argumentation (Atkinson et al., 2017). Moreover, the fields of belief revision (Hansson, 2001) and belief merging (Cholvy and Hunter, 1997; Konieczny and Pérez, 1998) deal with the particular case of inconsistencies in dynamic settings.

In the literature on inconsistency measurement, inconsistency measures are functions that aim at assessing the severity of the inconsistency in knowledge bases formalized in propositional logic (Hunter and Konieczny, 2004; Grant and Hunter, 2006; Thimm, 2018). The basic intuition behind an inconsistency measure nc 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 $\mathcal{I}_{\mathsf{MI}}(\mathcal{K}) = |I_{min}(\mathcal{K})|$, where $I_{min}(\mathcal{K})$ 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 $\mathcal{I}_{\mathsf{MI}}(\mathcal{K}) = 0$ 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 (Hunter and Konieczny, 2008; Jabbour et al., 2016; Jabbour and Sais, 2016).

Measuring inconsistency in non-monotonic logics has only recently gained some attention (Ulbricht et al., 2016; Brewka et al., 2019) and a thorough study is still needed. In this setting, a measure such as $\mathcal{I}_{\mathsf{MI}}$ is not applicable as a consistent non-monotonic knowledge base $\mathcal{K}$ may contain minimal inconsistent subsets. Recently, a refined notion of inconsistent subsets of a knowledge base $\mathcal{K}$ of a possibly non-monotonic framework has been introduced, called *strong $\mathcal{K}$-inconsistency* (Brewka et al., 2017). 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 (Besnard, 2014, 2017). The issue of measuring inconsistency in non-monotonic frameworks requires some reconsideration compared to the classical setting. This becomes apparent when considering the *monotonicity* 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 non-monotonic 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 $\mathcal{I}$ in this case.

In this paper, we provide a general account of measuring inconsistency in logics that are not necessarily monotonic. We do so by relying on a very general notion of a logic and we will phrase all our contributions in such a general manner. More concretely, the main contributions of this paper can be summarized as follows:

1. As the basis of our investigation, we consider generalized versions of three measures based on minimal inconsistent sets (Section 3).

2. In order to assess the behavior of these measures, we develop rationality postulates based on previous ones from the literature; some of the postulates still make sense for a general, possibly non-monotonic logic, but most of them require refinements (Section 4).

3. We analyze the measures with respect to the postulates (Section 5).

4. We assess the computational complexity of the measures by considering natural decision and function problems (Section 6).

5. We extend the hitting set duality from previous work (Brewka et al., 2019) to situations where knowledge bases can be repaired by adding information; moreover the observation that conflicts may be resolved due to additional formulas gives rise to the question of how to assess inconsistencies of a knowledge base within the context of a larger one; we analyze this setting in depth (Section 7).

We introduce preliminaries in Section 2, discuss related work in Section 8, and conclude in Section 9.

The paper combines and extends results from previous works (Ulbricht et al., 2018; Ulbricht, 2019). In particular, a) the discussion on rationality postulates is greatly extended, b) all full proofs are given, c) the discussion on measuring inconsistent subsets in Section 7 is novel, and d) we give more examples throughout the paper.

2. Background

In this paper, we will make use of a very general notion of a logic, but we will provide examples in some concrete instantiations. For that we provide the necessary background information on propositional logic (Section 2.1), answer set programming (Section 2.2), and abstract argumentation (Section 2.3). In Section 2.4 we give the general definition of a logic encompassing the previously mentioned instantiations. In Section 2.5 and 2.6 we discuss two important aspects of logics, which are the center of our work, namely *inconsistency* and *monotonicity*, respectively.

2.1. Propositional Logic

We define propositional logic as usual, so let us briefly recall the standard definitions. Let A be a (possibly infinite) set of propositional atoms, i. e., a propositional signature. Any atom $a \in A$ is a *well-formed formula* wrt. A. If ϕ and ψ are well-formed formulas wrt. A, then $\neg\phi$, $\phi \wedge \psi$, and $\phi \vee \psi$ are also well-formed formulas wrt. A (we also assume that the usual abbreviations $\rightarrow, \leftrightarrow$ are defined accordingly). A *literal* is either an atom a or its negation $\neg a$. Let $Lit(A)$ be the set of all literals over A. A formula ϕ is in *conjunctive normal form* (CNF) if it is of the form $\phi = C_1 \wedge \ldots \wedge C_r$ where each C_k is a *clause*, i. e., C_k is of the form $C_k = a_{k,1} \vee \ldots \vee a_{k,n(k)}$ for literals $a_{k,j}$ (for $1 \leq k \leq r$ and $1 \leq j \leq n(k)$). If each C_k contains at most 3 literals, then ϕ is in 3-CNF. We abuse notation and identify a formula ϕ of this form with the set $\{C_1, \ldots, C_r\}$ of clauses. Similarly, a formula ϕ is in *disjunctive normal form* (DNF) if $\phi = C_1 \vee \ldots \vee C_r$ where $C_k = a_{k,1} \wedge \ldots \wedge a_{k,n(k)}$ (for $1 \leq k \leq r$ and $1 \leq j \leq n(k)$). If each C_k contains at most 3 literals, then ϕ is in 3-DNF.

If $\omega : A \rightarrow \{0, 1\}$ is an assignment, then ω is extended to formulas in the usual way:

- $\omega(\neg a) = 1 - a$,

- $\omega(\phi \wedge \psi) = \min\{\omega(\phi), \omega(\psi)\}$ and
- $\omega(\phi \vee \psi) = \max\{\omega(\phi), \omega(\psi)\}$,

If $\omega(\phi) = 1$, then we say ω *satisfies* ϕ. A *propositional knowledge base* $\mathcal{K}$ is a finite set of propositional formulas. As usual, ω satisfies $\mathcal{K}$ iff ω satisfies ϕ for all $\phi \in \mathcal{K}$. We say $\mathcal{K}$ *entails* a formula ϕ, denoted by $\mathcal{K} \vDash \phi$, iff each assignment ω satisfying $\mathcal{K}$ also satisfies ϕ.

We call a knowledge base $\mathcal{K}$ *consistent* iff there is an assignment ω satisfying $\mathcal{K}$, otherwise it is called *inconsistent*.

Example 2.1. Consider the propositional knowledge base $\mathcal{K} = \{a,\ a \rightarrow b,\ \neg b,\ c,\ \neg c\}$. Obviously, no assignment satisfies $\mathcal{K}$. Hence, $\mathcal{K}$ is inconsistent.

2.2. Answer Set Programming

Answer set programming (ASP) is a problem solving paradigm (Brewka et al., 2011). It is based on logic programs under the answer set semantics (Gelfond and Leone, 2002; Gelfond and Lifschitz, 1991), a popular non-monotonic formalism for knowledge representation and reasoning which consists of rules possibly containing default-negated literals. Inconsistencies occur in ASP for two reasons (Schulz et al., 2015). First, the rules allow the derivation of two complementary literals l and $\neg l$ –also called *incoherence* (Madrid and Ojeda-Aciego, 2010)– thus producing inconsistencies similar to propositional logic. Second, due to the use of default negation it may happen that some literal assumed to be false is again derived (called *instability*).

Let us consider logic programs with disjunction in the head of rules and two kinds of negation, namely strong negation "¬" and default negation "not", under the answer set semantics (Gelfond and Leone, 2002; Gelfond and Lifschitz, 1991). Such programs are also called extended disjunctive databases (Gelfond and Lifschitz, 1991) or simply logic programs or A-Prolog programs (Gelfond and Leone, 2002).

Assume we are given a (possibly infinite) set A of atoms. Then, a disjunctive logic program P (over A) is a finite set of rules r of the form

$$l_0 \vee ... \vee l_k \leftarrow l_{k+1}, \ldots, l_m, \text{not } l_{m+1}, \ldots, \text{not } l_n. \tag{1}$$

where $l_0, \ldots, l_n$ are literals over A and $0 \leq k \leq m \leq n$. If $k = 0$ holds for each rule $r \in P$, then we call P a *normal logic program*. When there is no risk of confusion, we will simply speak of *logic programs* instead of disjunctive logic programs resp. normal logic programs.

For a rule r of the form (1) let $head(r) = \{l_0, \ldots, l_k\}$, $pos(r) = \{l_{k+1}, \ldots, l_m\}$, and $neg(r) = \{l_{m+1}, \ldots, l_n\}$. If $m = n = k$, then r is written "$head(r)$." instead of "$head(r) \leftarrow .$" and if in addition $k = 0$ holds, then the rule is called a *fact*.

Now we are ready to define answer sets of a given program.

Definition 2.2. Let P be a logic program over A such that $neg(r) = \emptyset$ holds for each rule $r \in P$. Then, a set M of literals is a *model* of P if for all $r \in P$ the following is true: If $pos(r) \subseteq M$, then $head(r) \cap M \neq \emptyset$. If M is a model of P containing two complementary literals, then M is extended to $M = Lit(A)$. A model M is *minimal* if for all proper subsets M' of M, M' is not a model of P. A minimal model of P is called an *answer set* of P.

Example 2.3. Consider the program P:

$$P: \qquad a \vee b.$$

The program has two answer sets $\{a\}$ and $\{b\}$, as well as the model $\{a, b\}$. The latter is no answer set.

Example 2.4. The program P

$$P: \qquad a \vee b. \qquad a \leftarrow b. \qquad c. \qquad \neg c.$$

possesses the answer set $Lit(A)$.

We extend the definition of an answer set now to arbitrary logic programs. For that assume we are given a logic program P and a set M of literals. We call

$$P^M = \{head(r) \leftarrow pos(r) \mid head(r) \leftarrow pos(r), neg(r) \in P, neg(r) \cap M = \emptyset\}$$

the *reduct* of P wrt. M. Observe that P^M itself is a logic program and $neg(r) = \emptyset$ holds for each $r \in P^M$. Now we define:

Definition 2.5. Let P be a logic program over A. A set M of literals is an *answer set* of P iff M is an answer set of P^M.

Example 2.6. Let P be the program

$$P: \qquad a \leftarrow \text{not } a.$$

Let us consider $M_1 = \emptyset$ and $M_2 = \{a\}$. We find

$$P^{M_1} = \{a.\} \qquad\qquad P^{M_2} = \{\}.$$

In particular, both M_1 and M_2 are *not* answer sets of P, because M_1 is not a model of P^{M_1} and M_2 –although being a model of P^{M_2}– is not minimal.

Example 2.7. Now consider the following program P, which will be one of our running examples:

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

The program has no answer set. To see this, consider the three candidates $\{a\}$, $\{b\}$ and $\{a, b\}$ with

$$\begin{array}{lllll} P^{\{a\}}: & a \vee b. & a \leftarrow b. & c. & \neg c. \\ P^{\{b\}}: & a \vee b. & a \leftarrow b. & & \\ P^{\{a,b\}}: & a \vee b. & a \leftarrow b. & & \end{array}$$

We see that $\{a\}$ is not a model of $P^{\{a\}}$, $\{b\}$ is not a model of $P^{\{b\}}$ and $\{a, b\}$ is a model of $P^{\{a,b\}}$, but not minimal.

Note that so far, we defined what an answer set is, no matter whether it is consistent or not. Recall that if a model contains two complementary literals, it is extended to $Lit(A)$. Clearly, this should not be considered a *consistent* answer set. Moreover, in order for a program to be consistent, it should possess consistent answer sets. Hence, we define:

Definition 2.8. Let P be a logic program over A. An answer set M is *consistent* if it does not contain two complementary literals. The program P is consistent if it possesses at least one consistent answer set.

We thus see that the program from Example 2.7 is inconsistent.

Finally, we call a rule of the form

$$r: \quad a \leftarrow l_1, \ldots, l_m, \text{not } l_{m+1}, \ldots, \text{not } l_n, \text{not } a. \tag{2}$$

where a is an atom that does not occur elsewhere in a given program P a *constraint*. The intuitive meaning is that no answer set of P is allowed to contain all literals $l_1, \ldots, l_m$ and none of the literals $l_{m+1}, \ldots, l_n$. We use the established shorthand

$$\leftarrow l_1, \ldots, l_m, \text{not } l_{m+1}, \ldots, \text{not } l_n.$$

for constraints of the form (2).

In the original formulation (Dung, 1995), an abstract argumentation framework (AF) F is a directed graph $F = (A, R)$ where nodes in A represent arguments and the relation R models "attacks", i. e., for $a, b \in A$, if $(a, b) \in R$ then a is a counterargument for b and we say a attacks b. Abstract argumentation frameworks consider the problem of argumentation only at this abstract level and do neither consider the inner structure of arguments nor how the attack relation is derived. Semantics are given to an abstract argumentation framework $F = (A, R)$ by identifying sets $E \subseteq A$ of arguments (called extensions) that can be "jointly accepted". The literature offers various approaches on how to define "jointly accepted", see (Baroni et al., 2018) for an overview.

Throughout this paper, we will focus on so-called *stable semantics* (Dung, 1995). This is an intuitive semantics that is easy to understand and thus an appropriate tool to illustrate our results with examples from abstract argumentation.

Definition 2.9. Let $F = (A, R)$ be an AF. A set $E \subseteq A$ is called *stable extension* if

- $a, b \in E$ implies $(a, b) \notin R$,
- $c \in A \setminus E$ implies there is an $a \in E$ with $(a, c) \in R$.

We denote the set of stable extensions of an AF F by $\mathit{stable}(F)$.

The first item ensures that E is *conflict free*, i. e., there are no two "accepted" arguments that attack each other. The second item is what characterizes stable semantics: each argument which is not included in E shall be attacked by E. This is a rather decisive requirement, partitioning the arguments in "accepted" and "rejected" ones.

Example 2.10. Consider the AF $F = (A, R)$ where

$$A = \{a, b, c, d\} \qquad R = \{(a, b),\ (b, a),\ (c, b),\ (c, c),\ (d, c)\}.$$

The AF is depicted in Figure 1. The stable extensions of F are

$$\mathit{stable}(F) = \{\{a, d\},\ \{b, d\}\}$$

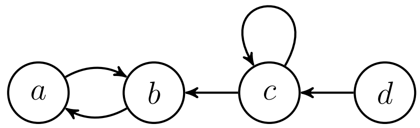


Figure 1: The argumentation framework F from Example 2.10

In comparison to other semantics, stable semantics possess a rare property, namely that an AF might have no extension at all. This is, for example the case for the following simplified version of the previous AF.

Example 2.11. Consider the AF $F = (A, R)$ (see Figure 2) with

$$A = \{a, b, c\} \qquad\qquad R = \{(a, b),\ (b, c),\ (c, c)\}:$$

The argument c attacks itself, so $c \notin E$ if E is conflict free. However, we see

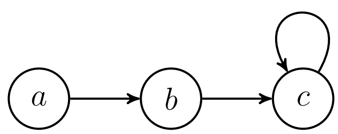


Figure 2: The argumentation framework F from Example 2.11

that in order to attack c, the argument b must be included in our extension E, but then, a can neither be included in E nor attacked.

This motivates our definition of inconsistency of an AF: similar to ASP where we call a program inconsistent whenever there is no (consistent) answer set, we will call an AF inconsistent whenever there is no stable extension.

Definition 2.12. Let F be an AF. If $stable(F) = \emptyset$, then we call F *inconsistent* wrt. stable semantics. If there is no risk of confusion, we will call F simply *inconsistent*.

2.4. Logics - A General Approach

Most of the main results in this work are independent of the actual logic, i. e., they hold for propositional logic, ASP, AFs and many other frameworks. It is thus natural to phrase those results for an arbitrary but fixed logic L.

To achieve this, we require a general definition of a logic, covering a wide range of frameworks as special cases.

We follow notation of previous work (Brewka et al., 2019) for a general logical framework. In a nutshell, a logic L consists of syntax and semantics of formulas. To model the syntax properly, we stipulate a set WF of so-called well-formed formulas. Any knowledge base $\mathcal{K}$ is a (finite) subset of WF. To model the semantics, we let BS be a set of so-called belief sets. Intuitively, given a knowledge base $\mathcal{K}$, the set of all formulas that can be inferred from $\mathcal{K}$ is $B \subseteq \mathsf{BS}$. To formalize this, a mapping ACC assigns the set B of corresponding belief sets to each knowledge base $\mathcal{K}$. For example, if our knowledge base is a logic program P, then we want to assign all answer sets of P to it. Hence, BS should contain all potential answer sets of P and we expect $\mathsf{ACC}(P) = \{M \mid M \text{ is an answer set of } P\}$. Finally, some belief sets are considered inconsistent. We call the set of all inconsistent belief sets INC. The inconsistent belief sets are supposed to model conflicting conclusions. We thus expect them to be upward-closed in BS, i. e., if $B, C \in \mathsf{BS}$ with $B \subseteq C$ and B is in INC, then $C \in \mathsf{INC}$ as well.[1]

Hence, our definition of a logic is as follows.

Definition 2.13. A logic L is a tuple

$$L = (\mathsf{WF}, \mathsf{BS}, \mathsf{INC}, \mathsf{ACC})$$

where WF is a set (of well-formed formulas), BS is a set (of belief sets), $\mathsf{INC} \subseteq \mathsf{BS}$ is upward closed wrt. BS and $\mathsf{ACC} : 2^{\mathsf{WF}} \to 2^{\mathsf{BS}}$ assigns a collection of belief sets to each subset of WF. A *knowledge base* $\mathcal{K}$ of L is a finite subset of WF.

In order to familiarize us with this abstract definition of a logic, let us illustrate how to model propositional logic, ASP and AFs under stable semantics as a logic according to Definition 2.13.

Example 2.14 (Propositional logic)**.** Let A be a set of propositional atoms. We define a logic

$$L_A^{\mathrm{P}} = \left(\mathsf{WF}_A^{\mathrm{P}}, \mathsf{BS}_A^{\mathrm{P}}, \mathsf{INC}_A^{\mathrm{P}}, \mathsf{ACC}_A^{\mathrm{P}}\right).$$

[1]Do not confuse upward-closure of a belief set with monotonicity of the logic itself; we assume this property also holds for non-monotonic logics.

We let $\mathsf{WF}_{\mathrm{P}}^{A}$ be the well-formed formulas over A (see the inductive definition in Section 2.1) and $\mathsf{BS}_{\mathrm{P}}^{A}$ the deductively closed sets of formulas, i. e.,

$$\mathsf{BS}_{A}^{\mathrm{P}} = \left\{\mathcal{K} \subseteq \mathsf{WF}_{A}^{\mathrm{P}} \mid \mathcal{K} = \{\phi \mid \mathcal{K} \vDash \phi\}\right\}.$$

The set $\mathsf{INC}_{A}^{\mathrm{P}}$ is supposed to contain the inconsistent belief sets. Since $\mathsf{INC}_{A}^{\mathrm{P}} \subseteq \mathsf{BS}_{\mathrm{P}}^{A}$, any set in $\mathsf{INC}_{A}^{\mathrm{P}}$ needs to be deductively closed as well. As anything can be derived from an inconsistent knowledge base, we define $\mathsf{INC}_{A}^{\mathrm{P}} = \left\{\mathsf{WF}_{A}^{\mathrm{P}}\right\}$. Finally, the mapping $\mathsf{ACC}_{A}^{\mathrm{P}}$ assigns to each $\mathcal{K} \subseteq \mathsf{WF}_{A}^{\mathrm{P}}$ the singleton set containing its set of theorems, i. e.,

$$\mathsf{ACC}_{A}^{\mathrm{P}}(\mathcal{K}) = \{\{\phi \mid \mathcal{K} \vDash \phi\}\}.$$

During the remainder of this work, we omit the superscript A whenever there is no risk of confusion.

Example 2.15 (Disjunctive logic programs)**.** Let A be a set of propositional atoms. Logic programs under answer set semantics over A can be modeled as logic

$$L_{A}^{\mathrm{ASP}} = \left(\mathsf{WF}_{A}^{\mathrm{ASP}}, \mathsf{BS}_{A}^{\mathrm{ASP}}, \mathsf{INC}_{A}^{\mathrm{ASP}}, \mathsf{ACC}_{A}^{\mathrm{ASP}}\right).$$

Here, $\mathsf{WF}_{A}^{\mathrm{ASP}}$ is the set of all rules of the form (1) over A (see Section 2.2). Moreover, $\mathsf{BS}_{A}^{\mathrm{ASP}}$ consists of the sets of literals over A, i. e.,

$$\mathsf{BS}_{A}^{\mathrm{ASP}} = 2^{A}$$

and $\mathsf{INC}_{A}^{\mathrm{ASP}} = \{Lit(A)\}$. The mapping $\mathsf{ACC}_{A}^{\mathrm{ASP}}$ assigns to a logic program $P \subseteq \mathsf{WF}_{A}^{\mathrm{ASP}}$ the set of all answer sets of P, i. e.,

$$\mathsf{ACC}_{A}^{\mathrm{ASP}}(P) = \left\{M \in 2^{A} \mid M \text{ is an answer set of } P\right\}.$$

As before we omit the superscript A whenever there is no risk of confusion.

It is a quite simple, yet pleasing observation that moving to a subclass of a certain logic just requires restricting WF.

Example 2.16 (Normal logic programs)**.** If we let $\mathsf{WF}_{A}^{\mathrm{ASP}^*} \subseteq \mathsf{WF}_{A}^{\mathrm{ASP}}$ be the set of all rules of the form (1) with $k = 0$, then

$$L_{A}^{\mathrm{ASP}^*} = \left(\mathsf{WF}_{A}^{\mathrm{ASP}^*}, \mathsf{BS}_{A}^{\mathrm{ASP}}, \mathsf{INC}_{A}^{\mathrm{ASP}}, \mathsf{ACC}_{A}^{\mathrm{ASP}}\right)$$

is the logic corresponding to normal logic programs under answer set semantics.

Example 2.17 (Abstract argumentation frameworks)**.** Representing an AF $F = (A, R)$ as a logic according to Definition 2.13 requires some caution since a knowledge base is supposed to be a subset of WF, but an AF is a tuple. In order to obtain a simple and intuitive representation of AFs as a set, let us assume a finite set A of arguments is given. Now, each well-formed formula corresponds to one attack within the AF.

More precisely, we define a logic

$$L_A^{\mathrm{AAF}} = \left(\mathsf{WF}_A^{\mathrm{AAF}}, \mathsf{BS}_A^{\mathrm{AAF}}, \mathsf{INC}_A^{\mathrm{AAF}}, \mathsf{ACC}_A^{\mathrm{AAF}}\right).$$

The set $\mathsf{WF}_A^{\mathrm{AAF}}$ is the set of all possible attacks, i. e., $\mathsf{WF}_A^{\mathrm{AAF}} = (A \times A)$. Note that by this treatment, we do not cover abstract argumentation frameworks with isolated arguments, i. e., arguments that are neither attacked nor attacking another argument. However, as those are always included in every stable extension anyway, we do not consider them for reasons of simplicity.

Belief sets are arbitrary sets of arguments, i. e., $\mathsf{BS}_A^{\mathrm{AAF}} = 2^A$. We consider no notion of an inconsistent set of arguments (recall that an AF F is inconsistent if $\mathit{stable}(F) = \emptyset$). We thus let $\mathsf{INC}_A^{\mathrm{AAF}} = \emptyset$. Hence, to represent an AF $F = (A, R)$, we fix the set A of arguments and let R be our knowledge base. Now, the AF under consideration is F, but the knowledge base is the set R. So we have

$$\mathsf{ACC}_A^{\mathrm{AAF}}(R) = \mathit{stable}(F), \qquad \text{where } F = (A, R).$$

We will thus sometimes abuse terminology and speak of the AF R instead of $F = (A, R)$ when A is given. As usual, we omit the superscript A whenever it is implicitly clear.

The reader may verify that a wide spectrum of other logics can be modeled as well, e. g., first-order logic, modal logic, probabilistic and fuzzy logics. This also includes examples where INC has more than one element, and examples where $\mathsf{ACC}(\mathcal{K})$ has elements in INC, but is not contained in INC. Consider disjunctive rules with classical negation (no default negation) and define the belief sets as minimal sets of literals closed under the rules, where a set of literals S is closed under a disjunctive rule r if S contains at least one of the disjuncts in the head of r whenever all body literals of r are in S; inconsistent belief sets are those containing a complementary pair of literals. Consider the rules $\{a \vee b, c \leftarrow a, \neg c \leftarrow a\}$. We obtain two belief sets, namely $\{b\}$ and $\{a, c, \neg c\}$, the first consistent, the latter inconsistent. We do not see any reason to exclude formalisms of this kind from our definition of logics.

2.5. Inconsistency

Consider a logic $L = (\mathsf{WF}, \mathsf{BS}, \mathsf{INC}, \mathsf{ACC})$. Until now, the meaning of the set INC—the inconsistent belief sets—is only intuitively clear. The definition of inconsistency is quite natural: A knowledge base should possess at least one consistent belief set in order to be consistent, otherwise we call it inconsistent. For example, a logic program should have at least one consistent answer set, as already noticed in Definition 2.8. A knowledge base $\mathcal{K}$ is thus inconsistent if all belief sets $\mathsf{ACC}(\mathcal{K})$ are.

Definition 2.18. A knowledge base $\mathcal{K}$ is called *inconsistent* iff $\mathsf{ACC}(\mathcal{K}) \subseteq \mathsf{INC}$. Let $I(\mathcal{K})$ denote the collection of all inconsistent subsets of $\mathcal{K}$, that is

$$I(\mathcal{K}) = \{M \subseteq \mathcal{K} \mid \mathsf{ACC}(M) \subseteq \mathsf{INC}\}$$

Let $I_{min}(\mathcal{K})$ be the set of all minimal (wrt. set inclusion) inconsistent subsets of $\mathcal{K}$, i. e., $I_{min}(\mathcal{K}) = \{M \in I(\mathcal{K}) \mid \text{there is no } M' \in I(\mathcal{K}) \text{ with } M' \subsetneq M\}$.

To be precise, inconsistency is a property a knowledge base has with respect to a given logic L. We should thus write $I_{min}(\mathcal{K})^L$ instead of $I_{min}(\mathcal{K})$. However, in most cases the underlying logic will be clear, so we may omit the superscript without risking confusion.

Now let us discuss the above definition of inconsistency. For propositional logic, inconsistency means that every formula can be inferred from a knowledge base.

Example 2.19 (Inconsistency in propositional logic)**.** Consider the propositional knowledge base $\mathcal{K} = \{a.,\ a \to b.,\ \neg b.,\ c.,\ \neg c.\}$ from above. Recall that $\mathcal{K}$ entails a contradiction, so by definition of propositional logic we have $\mathsf{ACC}^{\mathrm{P}}(\mathcal{K}) = \{\{a, \neg a, b, \dots\}\} = \left\{\mathsf{WF}^{\mathrm{P}}\right\}$. In particular, $\mathsf{ACC}^{\mathrm{P}}(\mathcal{K}) \subseteq \mathsf{INC}$. Thus, $\mathcal{K}$ is as expected inconsistent according to Definition 2.18.

Observe that our definition of inconsistency also captures cases where a given knowledge base $\mathcal{K}$ has no belief set at all. Formally, if $\mathsf{ACC}(\mathcal{K}) = \emptyset$, then $\mathsf{ACC}(\mathcal{K}) \subseteq \mathsf{INC}$ holds trivially. At a first glance, this may look like an overlooked technical detail. It is however intended and of importance for many non-monotonic frameworks, including ASP and AFs. The following example illustrates such a case.

Example 2.20 (Inconsistency in ASP)**.** Consider again the logic program P given as

$$
\begin{array}{lcc}
P: & a \vee b. & a \leftarrow b. \\
 & c \leftarrow \text{not } b. & \neg c \leftarrow \text{not } b.
\end{array}
$$

As pointed out in Example 2.7, P has no answer set. Therefore,

$$\mathsf{ACC}^{\mathrm{ASP}}(P) = \emptyset \subseteq \mathsf{INC}^{\mathrm{ASP}}$$

and hence P is considered inconsistent as well.

In fact, this is a quite common reason for a logic program to be inconsistent. In the original formulation, a logic program only contains atoms of the form "a" and no literals of the form "$\neg a$", so there is no notion of inconsistent answer sets. Hence, a program of this kind can only be inconsistent when possessing no answer set. Inconsistency in AFs is similar. Due to our definition, a given AF F is considered inconsistent whenever $\mathit{stable}(F) = \emptyset$.

Example 2.21 (Inconsistency in AFs)**.** Let us consider again the AF $F = (A, R)$ depicted in Figure 2 above. Recall that we assume A to be implicit, so our knowledge base is $R = \{(a, b),\ (b, c),\ (c, c)\}$. Since F has no stable extension we obtain

$$\mathsf{ACC}^{\mathrm{AAF}}(R) = \emptyset \subseteq \mathsf{INC}^{\mathrm{AAF}}.$$

The framework is thus considered inconsistent, as expected.

Having established a formal meaning of inconsistency, our definition of consistency is straightforward.

Definition 2.22. A knowledge base $\mathcal{K}$ is *consistent* if $\mathsf{ACC}(\mathcal{K}) \not\subseteq \mathsf{INC}$, i. e., it is not inconsistent. We let $C(\mathcal{K})$ and $C_{max}(\mathcal{K})$ denote the set of all consistent and maximal (wrt. set inclusion) consistent subsets of $\mathcal{K}$, respectively.

2.6. Monotonicity

As already mentioned, the central issue of this work is to investigate the behavior of inconsistency in non-monotonic logics. We thus require a formal definition of a monotonic logic in our setting. The intuitive understanding of monotonicity is that a conclusion which is inferred from a knowledge base $\mathcal{K}$

is never withdrawn due to additional information. We want to formalize this idea for our general logic while taking the usual reasoning modes, i. e., skeptical and credulous reasoning, into account. Our definition generalizes the one of Brewka and Eiter (2007). Whereas the latter requires monotonic logics to associate unique belief sets to knowledge bases, our definition shows that a reasonable notion of monotonicity can be defined for logics with multiple belief sets.

Definition 2.23. A logic $L = (\mathsf{WF}, \mathsf{BS}, \mathsf{INC}, \mathsf{ACC})$ is *skeptically monotonic* or simply *monotonic* whenever $\mathcal{K} \subseteq \mathcal{K}' \subseteq \mathsf{WF}$ implies:

- if $B' \in \mathsf{ACC}(\mathcal{K}')$ then $B \subseteq B'$ for some $B \in \mathsf{ACC}(\mathcal{K})$.

The name "skeptically" monotonic is motivated by the observation that in a skeptically monotonic logic, skeptical reasoning based on the intersection of belief sets is monotonic. More precisely, we have:

Proposition 2.24. *Let L be a skeptically monotonic logic. If $\mathcal{K}$ and $\mathcal{K}'$ are consistent knowledge bases and $\mathcal{K} \subseteq \mathcal{K}'$, then*

$$\bigcap_{B \in \mathsf{ACC}(\mathcal{K})} B \subseteq \bigcap_{B' \in \mathsf{ACC}(\mathcal{K}')} B'.$$

Proof. Let $p \in \bigcap_{B \in \mathsf{ACC}(\mathcal{K})} B$, i. e., $p \in B$ for each $B \in \mathsf{ACC}(\mathcal{K})$. Now consider an arbitrary B' with $B' \in \mathsf{ACC}(\mathcal{K}')$. Due to skeptical monotonicity, there is a $B \in \mathsf{ACC}(\mathcal{K})$ with $B \subseteq B'$. We thus have $p \in B \subseteq B'$, so $p \in B'$. Since B' was an arbitrary set in $\mathsf{ACC}(\mathcal{K}')$, $p \in \bigcap_{B' \in \mathsf{ACC}(\mathcal{K}')} B'$. $\square$

In this sense, the natural counterpart would be a notion of a "credulously" monotonic logic. This is indeed possible by requiring $B \subseteq B'$ for some $B' \in \mathsf{ACC}(\mathcal{K}')$ if $B \in \mathsf{ACC}(\mathcal{K})$. However, within the scope of this work the crucial monotonicity notion is the one given in Definition 2.23. The reason is that it ensures that conflicts within a knowledge base cannot be resolved by adding new information (see Lemma 2.29 below).

When there is no risk of confusion, we will call a knowledge base monotonic whenever its associated logic is. This is a slight abuse of terminology since monotonicity is a property of a logic, not of a knowledge base. However, leaving the actual logic implicit does no harm in many cases, and we prefer the simpler terminology.

Example 2.25 (Monotonicity in propositional logic)**.** Consider the propositional knowledge bases $\mathcal{K} = \{a \wedge b\}$ and $\mathcal{K}' = \{a \wedge b, b \to c.\}$. Observe that both $\mathsf{ACC}^{\mathrm{P}}(\mathcal{K})$ as well as $\mathsf{ACC}^{\mathrm{P}}(\mathcal{K}')$ are singletons. More precisely,

$$\mathsf{ACC}^{\mathrm{P}}(\mathcal{K}) = \{\{a, b, a \wedge b, a \vee b, a \vee c, \ldots\}\},$$
$$\mathsf{ACC}^{\mathrm{P}}(\mathcal{K}') = \{\{a, b, c, c, a \wedge b, a \wedge c, b \wedge c, \ldots\}\}.$$

So set $B = \{a, b, a \wedge b, a \vee b, a \vee c, \ldots\}$ and $B' = \{a, b, c, c, a \wedge b, a \wedge c, b \wedge c, \ldots\}$. Since $B \subseteq B'$ we see monotonicity according to Definition 2.23. As this is the case for any two propositional knowledge bases $\mathcal{K} \subseteq \mathcal{K}'$, we see that this logic is skeptically monotonic.

The following example illustrates that ASP is non-monotonic. The intuitive reason is as follows: Given two logic programs $P \subseteq P'$, it might happen that P' possesses a novel answer set in the sense that it is not a superset of an answer set of P. Within ASP, this is a common feature. It is thus sufficient to consider a rather simple example.

Example 2.26 (Monotonicity in ASP)**.** Consider $P \subseteq P'$ given as follows:

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

We have

$$\mathsf{ACC}^{\mathrm{ASP}}(P) = \{\{a\}\}, \qquad \mathsf{ACC}^{\mathrm{ASP}}(P') = \{\{a\}, \{b\}\}.$$

If we set $B' = \{b\} \in \mathsf{ACC}^{\mathrm{ASP}}(P')$, then there is no $B \in \mathsf{ACC}^{\mathrm{ASP}}(P)$ with $B \subseteq B'$. ASP is thus not skeptically monotonic.

It is also a straightforward observation that AFs are non-monotonic as well.

Example 2.27 (Monotonicity in AFs)**.** Recall the AF from Figure 2 which we represent as knowledge base $R = \{(a, b), (b, c), (c, c)\}$. Consider $R' = R \cup \{(a, c)\}$ which yields an AF with one additional attack. We see $\mathsf{ACC}^{\mathrm{AAF}}(R) = \{\}$ and $\mathsf{ACC}^{\mathrm{AAF}}(R') = \{\{a\}\}$. If we set $B' = \{a\}$, then there is no $B \in \mathsf{ACC}^{\mathrm{AAF}}(R)$ satisfying $B \subseteq B'$ which is trivial since $\mathsf{ACC}^{\mathrm{AAF}}(R)$ is empty.

Example 2.28. We already mentioned that skeptical monotonicity implies monotonicity of sceptical reasoning (see Proposition 2.24) above. To see this observation at work, let us consider the following example: Let us denote by $\mathsf{WF}^{\mathrm{ASP-not}}$ rules of the form (1) with $m = n$, i. e., rules r such that $neg(r) = \emptyset$. The induced logic

$$L^{\mathrm{ASP-not}} = \left(\mathsf{WF}^{\mathrm{ASP-not}}, \mathsf{BS}^{\mathrm{ASP}}, \mathsf{INC}^{\mathrm{ASP}}, \mathsf{ACC}^{\mathrm{ASP}}\right)$$

is skeptically monotonic. Now consider for example the programs P and P' with $P \subseteq P'$

$$\begin{array}{llllll} P: & a \vee b. & \qquad P': & a \vee b. & \\ & c \leftarrow a. & & c \leftarrow a. & \quad d \leftarrow a. \\ & c \leftarrow b. & & c \leftarrow b. & \quad e \leftarrow b. \end{array}$$

By $\mathsf{ACC}^{\mathrm{ASP}}(P) = \{\{a, c\}, \{b, c\}\}$ we see that c is skeptically accepted in P. By skeptical monotonicity of this logic, given $M' \in \mathsf{ACC}^{\mathrm{ASP}}(P')$, there is $M \in \mathsf{ACC}^{\mathrm{ASP}}(P)$ with $M \subseteq M'$. Indeed, $\mathsf{ACC}^{\mathrm{ASP}}(P')$ contains $\{a, c, d\}$ and $\{b, c, e\}$ which are supersets of $\{a, c\}$ and $\{b, c\}$, respectively. It can hence immediately be seen that c is also skeptically accepted in P'.

Lemma 2.29 (Brewka et al., 2019)**.** *Let $L = (\mathit{WF}, \mathit{BS}, \mathit{INC}, \mathit{ACC})$ be monotonic and $\mathcal{K} \subseteq \mathcal{K}'$. If $\mathcal{K}$ is inconsistent, then so is $\mathcal{K}'$.*

Throughout this work, our results regarding monotonic logics depend especially on the above Lemma 2.29 which states that inconsistency survives moving to supersets. Regarding inconsistency in non-monotonic logics, the loss of this property is the central issue we need to handle.

3. Measures for Strong Inconsistency

We can now define *strong inconsistency* (Brewka et al., 2019) for our general setting, which is the central notion used for the measures developed in this paper.

Definition 3.1. For $\mathcal{H}, \mathcal{K} \subseteq \mathsf{WF}$ with $\mathcal{H} \subseteq \mathcal{K}$, $\mathcal{H}$ is called *strongly $\mathcal{K}$-inconsistent* if $\mathcal{H} \subseteq \mathcal{H}' \subseteq \mathcal{K}$ implies $\mathcal{H}'$ is inconsistent. The set $\mathcal{H}$ is simply called *strongly inconsistent* if there is no risk of confusion. Denote by $SI(\mathcal{K})$ and $SI_{min}(\mathcal{K})$ the set of all strongly inconsistent and all minimal (wrt. set inclusion) strongly inconsistent subsets of $\mathcal{K}$, respectively.

In other words, a subset of a knowledge base $\mathcal{K}$ is strongly inconsistent if all its supersets within $\mathcal{K}$ are inconsistent as well. Intuitively, one can think of a conflict that cannot be resolved by formulas in $\mathcal{K}$ itself.

Example 3.2. Consider again $\mathcal{K} = \{a, a \to b, \neg b, c, \neg c\}$. Recall that $\{c, \neg c\}$ is an inconsistent subset of $\mathcal{K}$. Of course, any set $\mathcal{H}'$ with $\{c, \neg c\} \subseteq \mathcal{H}' \subseteq \mathcal{K}$ is inconsistent as well. Hence, $\{c, \neg c\}$ is strongly $\mathcal{K}$-inconsistent.

More generally, Lemma 2.29 ensures that classical and strong inconsistency coincide whenever our logic is monotonic (Brewka et al., 2019). Let us thus consider examples involving non-monotonic logics to see the definition at work.

Example 3.3. Consider again the logic program P from above.

$$\begin{array}{lrr} P: & a \vee b. & a \leftarrow b. \\ & c \leftarrow \text{not } b. & \neg c \leftarrow \text{not } b. \end{array}$$

Recall that $H = \{c \leftarrow \text{not } b., \neg c \leftarrow \text{not } b.\}$ is an inconsistent subset of P. There is a consistent program H' with $H \subseteq H' \subseteq P$, namely

$$H': \qquad a \vee b. \qquad c \leftarrow \text{not } b. \qquad \neg c \leftarrow \text{not } b.$$

Hence, H is not strongly inconsistent. However,

$$H'': \qquad a \leftarrow b. \qquad c \leftarrow \text{not } b. \qquad \neg c \leftarrow \text{not } b.$$

is strongly inconsistent; even minimal. In particular, there is no other minimal strongly inconsistent set, therefore

$$SI_{min}(P) = \{\{a \leftarrow b., \ c \leftarrow \text{not } b., \ \neg c \leftarrow \text{not } b.\}\}.$$

Example 3.4. Recall the AF depicted in Figure 2 which corresponds to the knowledge base $R = \{(a, b), (b, c), (c, c)\}$. We already found the inconsistent subset $\{(c, c)\}$. However, the framework over $A = \{a, b, c\}$ with attacks $\{(b, c), (c, c)\}$ has a stable extension, namely $\{b\}$. Hence, the set $\{(c, c)\}$ of attacks is not strongly inconsistent. The reader may verify that

$$SI_{min}(R) = \{\{(a, b), \ (c, c)\}\}.$$

It has been shown that the notion of strong inconsistency faithfully generalizes classical inconsistency to arbitrary logics (Brewka et al., 2019). In particular, the notions coincide for monotonic logics and the existence of a strongly inconsistent subset of $\mathcal{K}$ is a necessary and sufficient condition for inconsistency of $\mathcal{K}$ itself. Moreover, removing from $\mathcal{K}$ any minimal hitting set[2] 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 (Reiter, 1987). We refer to our previous work (Brewka et al., 2017) for a more thorough discussion of strong inconsistency.

We now introduce the inconsistency measures we are going to consider throughout most of this paper. Assume an arbitrary but fixed logic L. In classical inconsistency measurement, minimal inconsistent subsets of a knowledge base play an important role since 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 $|I_{min}(\mathcal{K})|$. The notion of strong inconsistency facilitates the following generalization of this measure to arbitrary logics.

Definition 3.5. Define $\mathcal{I}_{\mathsf{MSI}} : 2^{\mathsf{WF}} \to \mathbb{R}^{\infty}_{\geq 0}$ via $\mathcal{I}_{\mathsf{MSI}}(\mathcal{K}) = |SI_{min}(\mathcal{K})|$.

We want to emphasize that the co-domain of $\mathcal{I}_{\mathsf{MSI}}$ (and the other measures we introduce as well) is explicitly restricted to non-negative numbers. The reason is that inconsistency measures aim at generalizing the binary view of "consistency" vs. "inconsistency". For an inconsistency measure $\mathcal{I}$, the intuitive meaning of $\mathcal{I}(\mathcal{K}) = 0$ is that $\mathcal{K}$ is consistent and $\mathcal{I}(\mathcal{K}') \leq \mathcal{I}(\mathcal{K}'')$ means $\mathcal{K}''$ is "at least as inconsistent as" $\mathcal{K}'$. Since an inconsistency measure does not distinguish between consistent knowledge bases, there is no need for negative values.

One drawback of the approach from the previous definition is that the size of a set $\mathcal{H} \in SI_{min}(\mathcal{K})$ is not taken into account. Usually, a minimal inconsistent subset is considered more severe the smaller it is, i. e., the fewer formulas are required in order to yield a contradiction. A famous example to illustrate this is the so-called *lottery paradox* Knight (2001).

Example 3.6. Assume there is a lottery with n tickets. Consider atoms $t_1, \ldots, t_n$ with the intuitive meaning that t_i is true iff the i-th ticket wins. Assume the lottery is fair and one ticket wins, i. e., $t_1 \vee \ldots \vee t_n$. However,

[2] A set $\mathcal{S}$ is called a *hitting set* of a set of sets $\mathcal{M} = \{M_1, \ldots, M_n\}$ iff $\mathcal{S} \cap M_i \neq \emptyset$ for $i = 1, \ldots, n$. It is minimal of no proper subset of it is a hitting set.

considering an individual ticket t_i it appears reasonable to assume that it loses, so we have $\neg t_1, \ldots, \neg t_n$. We thus obtain the inconsistent knowledge base $\mathcal{K}_n = \{t_1 \vee \ldots \vee t_n, \neg t_1, \ldots, \neg t_n\}$. Now consider a lottery where n is quite small, e.g., $n = 1$. In this case, the assumption that t_1 loses is not as reasonable anymore given that at least one ticket wins. However, the bigger n is, the more reasonable this assumption becomes, e.g., $n = 10^6$ yields a negligible chance for each t_i to win. Hence, even though both $\mathcal{K}_1$ and $\mathcal{K}_{10^6}$ are inconsistent, the latter appears quite reasonable while the former is hardly reasonable.

So commonly, the bigger a minimal inconsistent set is, the less severe the conflict is viewed. This is obviously ignored by $\mathcal{I}_{\mathsf{MSI}}$. For example, $\mathcal{I}_{\mathsf{MSI}}(\mathcal{K}_n) = 1$ for any n for the knowledge base $\mathcal{K}_n$ from the lottery paradox. The measure $\mathcal{I}_{\mathsf{MI}^c}$ has been proposed to take this into account (Hunter and Konieczny, 2008). Making use of strong inconsistency, we obtain the following measure:

Definition 3.7. Define $\mathcal{I}_{\mathsf{MSI}^\mathsf{c}} : 2^{\mathsf{WF}} \to \mathbb{R}^{\infty}_{\geq 0}$ via $\mathcal{I}_{\mathsf{MSI}^\mathsf{c}}(\mathcal{K}) = \sum_{\mathcal{H} \in SI_{min}(\mathcal{K})} \frac{1}{|\mathcal{H}|}$.

Instead of counting the number of sets in $SI_{min}(\mathcal{K})$, one could also consider the number of *formulas* which are considered problematic. Based on an existing measure (Grant and Hunter, 2011), we have the following, quite simple approach:

Definition 3.8. Define $\mathcal{I}_{\mathsf{p}} : 2^{\mathsf{WF}} \to \mathbb{R}^{\infty}_{\geq 0}$ via $\mathcal{I}_{\mathsf{p}}(\mathcal{K}) = |\bigcup_{\mathcal{H} \in SI_{min}(\mathcal{K})} \mathcal{H}|$.

Note that there are further measures based on minimal inconsistent sets (Hunter and Konieczny, 2008; Jabbour and Sais, 2016; Jabbour et al., 2016). An investigation of generalizations of those is left for future work.

Example 3.9. Consider our running examples from the previous section, i. e., the propositional knowledge base $\mathcal{K} = \{a,\ a \to b,\ \neg b,\ c,\ \neg c\}$ with

$$SI_{min}(\mathcal{K}) = \{\{a,\ a \to b,\ \neg b\},\ \{c,\ \neg c\}\},$$

the logic program

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

with

$$SI_{min}(P) = \{\{c \leftarrow \text{not } b.,\ \neg c \leftarrow \text{not } b.,\ a \leftarrow b.\}\}$$

and the argumentation framework over $A = \{a, b, c\}$ represented by the knowledge base $R = \{(a, b), (b, c), (c, c)\}$ with

$$SI_{min}(R) = \{\{(a, b), (c, c)\}\}.$$

The inconsistency measures from above assign the following values:

$$\begin{array}{lll}
\mathcal{I}_{\mathsf{MSI}}(\mathcal{K}) = 2 & \mathcal{I}_{\mathsf{MSI^c}}(\mathcal{K}) = 5/6 & \mathcal{I}_{\mathsf{p}}(\mathcal{K}) = 5 \\
\mathcal{I}_{\mathsf{MSI}}(P) = 1 & \mathcal{I}_{\mathsf{MSI^c}}(P) = 1/3 & \mathcal{I}_{\mathsf{p}}(P) = 3 \\
\mathcal{I}_{\mathsf{MSI}}(R) = 1 & \mathcal{I}_{\mathsf{MSI^c}}(R) = 1/2 & \mathcal{I}_{\mathsf{p}}(R) = 2
\end{array}$$

We observe that $\mathcal{I}_{\mathsf{MSI^c}}$ and $\mathcal{I}_{\mathsf{p}}$ differ for P and R even though both possess one minimal strongly inconsistent subset.

Whenever we are given an inconsistency measure $\mathcal{I}$, we oftentimes call the value $\mathcal{I}(\mathcal{K})$ the *inconsistency degree* of a knowledge base $\mathcal{K}$.

4. 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 non-monotonic, logic. We will start by considering the four postulates that a *basic inconsistency measure* should have (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 $L = (\mathsf{WF}, \mathsf{BS}, \mathsf{INC}, \mathsf{ACC})$ and an inconsistency measure $\mathcal{I} : 2^{\mathsf{WF}} \rightarrow \mathbb{R}^{\infty}_{\geq 0}$ for the remainder of this section.

4.1. 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. Undoubtedly this makes sense in non-monotonic frameworks as well.

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

In contrast to non-monotonic logics, for consistent $\mathcal{K}$ the *consistency* postulates implies $\mathcal{I}(\mathcal{H}) = 0$ for each $\mathcal{H} \subseteq \mathcal{K}$ if the underlying logic is monotonic. This follows immediately from Lemma 2.29.

Proposition 4.1. *If L is monotonic, $\mathcal{K}$ is consistent and $\mathcal{I}$ satisfies consistency, then we also have $\mathcal{I}(\mathcal{H}) = 0$ for each $\mathcal{H} \subseteq \mathcal{K}$.*

Since we do not see any reason to doubt the *consistency* postulate for our setting, let us continue with the second one, which is *monotonicity*. *Monotonicity* is a fairly accepted postulate which formalizes the intuition that moving from a *monotonic* knowledge base to a superset should not decrease the inconsistency degree since adding information to a knowledge base does not resolve conflicts.

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

In non-monotonic frameworks, *monotonicity* does not make sense because additional information might resolve some conflicts or even render $\mathcal{K}$ consistent. We should thus only expect a monotonic behavior if $\mathcal{K}'$ does not resolve conflicts occurring in $\mathcal{K}$. More precisely, if $\mathcal{H} \subseteq \mathcal{K}$ is strongly inconsistent, i. e., $\mathcal{H} \in SI(\mathcal{K})$, then there should be no subset $\mathcal{H}' \subseteq \mathcal{K}'$ such that $\mathcal{H} \cup \mathcal{H}'$ is consistent. Otherwise, if $\mathcal{H} \cup \mathcal{H}'$ is consistent, then it is unclear whether $\mathcal{H}$ even contributes to inconsistency of $\mathcal{K} \cup \mathcal{K}'$, which makes a comparison between $\mathcal{I}(\mathcal{K})$ and $\mathcal{I}(\mathcal{K} \cup \mathcal{K}')$ questionable. To illustrate this, consider the following example:

Example 4.2. Recall our logic program

$$\begin{array}{lrr} P: & a \vee b. & a \leftarrow b. \\ & c \leftarrow \text{not } b. & \neg c \leftarrow \text{not } b. \end{array}$$

Consider $P' = \{b.,\ d.,\ \neg d.\}$ containing "$b.$", which resolves the conflict within P: The subprogram $H \in SI_{min}(P)$ with $H = \{c \leftarrow \text{not } b., \neg c \leftarrow \text{not } b., a \leftarrow b.\}$ is not strongly $(P \cup P')$-inconsistent due to the consistent superprogram $H \cup H'$ with $H' = \{b.\} \subseteq P'$:

$$H \cup H': \qquad c \leftarrow \text{not } b. \qquad \neg c \leftarrow \text{not } b. \qquad a \leftarrow b. \qquad b.$$

In particular, $P \cup \{b.\}$ is consistent as well, but P' involves the conflict "$d.$" vs. "$\neg d.$". We have $SI_{min}(P \cup P') = \{\{d., \neg d.\}\}$ which only represents the conflict within P'. So the comparison between inconsistency of P and $P \cup P'$ does not seem to make sense.

We thus need to adjust the *monotonicity* postulate in order to obtain a meaningful one for non-monotonic frameworks. Intuitively, we do not want *monotonicity* to hold for *each* additional piece of information. We have to restrict the postulate to the case where *the additional information has no influence on the conflicts* contained in the original knowledge base; in other words, where the original conflicts are preserved by the new information. We have already seen that the conflicts relevant for non-monotonic frameworks are those that manifest themselves as strongly inconsistent subsets. This leads to the following notion of conflict preservation:

Definition 4.3. Let $\mathcal{K}$ and $\mathcal{K}'$ be knowledge bases. We say that $\mathcal{K}'$ *preserves conflicts of* $\mathcal{K}$ if $\mathcal{H} \in SI(\mathcal{K} \cup \mathcal{K}')$ for any $\mathcal{H} \in SI(\mathcal{K})$.

We observe that this property transfers to *minimal* strongly inconsistent subsets as well:

Proposition 4.4. *If $\mathcal{K}$ and $\mathcal{K}'$ are knowledge bases and $\mathcal{K}'$ preserves conflicts of $\mathcal{K}$, then $SI_{min}(\mathcal{K}) \subseteq SI_{min}(\mathcal{K} \cup \mathcal{K}')$.*

Proof. Let $\mathcal{H} \in SI_{min}(\mathcal{K})$. Since $\mathcal{K}'$ preserves conflicts, $\mathcal{H} \in SI(\mathcal{K} \cup \mathcal{K}')$. Now assume there is a set $\mathcal{H}' \subsetneq \mathcal{H}$ with $\mathcal{H}' \in SI(\mathcal{K} \cup \mathcal{K}')$. Since $\mathcal{H}' \subseteq \mathcal{H}$, we have $\mathcal{H}' \in SI(\mathcal{K})$, yielding a contradiction as $\mathcal{H}$ was assumed to be minimal. Hence $\mathcal{H} \in SI_{min}(\mathcal{K} \cup \mathcal{K}')$. □

Let us now consider our running examples again to see the definition at work. We first start with the monotonic propositional knowledge base from before. As the reader may already expect, monotonicity renders the property of preserving conflicts trivial.

Example 4.5. Recall $\mathcal{K} = \{a,\ a \to b,\ \neg b,\ c,\ \neg c\}$. Any propositional knowledge base $\mathcal{K}'$ preserves conflicts of $\mathcal{K}$ due to monotonicity of the logic, so we have $\{c, \neg c\} \in SI_{min}(\mathcal{K} \cup \mathcal{K}')$.

More generally, the following statement can be easily inferred from Lemma 2.29.

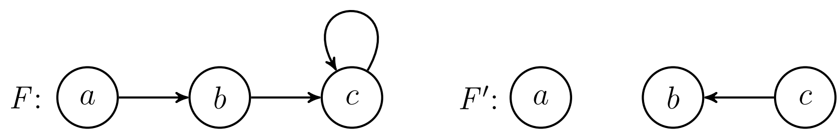
Proposition 4.6. *Let L be monotonic and $\mathcal{K}$ and $\mathcal{K}'$ be two knowledge bases of L. Then $\mathcal{K}'$ preserves conflicts of $\mathcal{K}$.*

Example 4.7. Recall the logic program

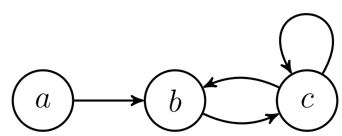
$$
\begin{array}{lrr}
P: & a \vee b. & a \leftarrow b. \\
 & c \leftarrow \text{not } b. & \neg c \leftarrow \text{not } b.
\end{array}
$$

The program $P' = \{b.,\ d.,\ \neg d.\}$ from Example 4.2 does not preserve conflicts of P as we already noted above. More precisely, we saw that $H = \{c \leftarrow \text{not } b.,\ \neg c \leftarrow \text{not } b.,\ a \leftarrow b.\}$ is not strongly $(P \cup P')$-inconsistent since P' contains the fact "$b.$".

Example 4.8. Now consider the running example AF $F = (A, R)$ with $A = \{a, b, c\}$ represented by the knowledge base $R = \{(a, b), (b, c), (c, c)\}$. A quite simple choice of a knowledge base R' which preserves conflicts of R is $R' = \{(c, b)\}$, which induces the AF $F' = (A, R')$:



We see that $SI_{min}(R \cup R') = \{\{(a, b),\ (c, c)\}\} = SI_{min}(R)$. The AF represented by $R \cup R'$ is the following:



With the notion of conflict preservation at hand, we are ready to formulate the restriction of *monotonicity* we were aiming for. The obvious idea is to require $\mathcal{K}'$ not to resolve conflicts within $\mathcal{K}$.

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

The name *strong monotonicity* emphasizes the role of strongly inconsistent subsets. It is obvious, though, that the new postulate actually weakens *monotonicity* due to its additional precondition. From Prop. 4.6 we immediately obtain that for monotonic logics strong and regular monotonicity coincide, since in these logics additional information is always conflict preserving.

We now turn to the *free formula independence* postulate (Hunter and Konieczny, 2010). Intuitively, a formula α of a knowledge base $\mathcal{K}$ is *free* in $\mathcal{K}$ if it does not cause any conflicts. The classical definition of a free formula—which we also use a starting point for our discussion—is a formula that is not a member of any minimal inconsistent subset. Mu (2019) provides another interpretation where "freeness" is not based on minimal inconsistency on a formula level but on an atom level (using three-valued semantics to discover which atoms take part in inconsistencies). In any interpretation, a free formula should thus not be "blamed" for inconsistency of $\mathcal{K}$ and hence not change the inconsistency degree. We consider two ways to generalize the classical definition of a free formula to arbitrary logics, which we believe to be reasonable. We leave an investigation of Mu's interpretation for future work.

Consider a monotonic logic. Formally, a free formula $\alpha \in \mathcal{K}$ is one that does not occur in a minimal inconsistent subset $\mathcal{H} \in SI_{min}(\mathcal{K}) = I_{min}(\mathcal{K})$.

Definition 4.9. Let $\mathcal{K}$ be a monotonic knowledge base. A formula $\alpha \in \mathcal{K}$ is called *free* if

$$\alpha \in \mathcal{K} \setminus \bigcup_{\mathcal{H} \in I_{min}(\mathcal{K})} \mathcal{H}. \tag{3}$$

Denote by $\mathit{Free}(\mathcal{K})$ the set of all free formulas of $\mathcal{K}$.

This notion matches the intuition that α is not responsible for any conflict within $\mathcal{K}$. Whenever there is a consistent subset $\mathcal{H} \subseteq \mathcal{K}$, then $\mathcal{H} \cup \{\alpha\}$ is consistent as well, so α does not cause any harm. In fact, for classical inconsistency measurement, the postulate *free formula independence*—which requires $\mathcal{I}(\mathcal{K}) = \mathcal{I}(\mathcal{K} \setminus \{\alpha\})$ for $\alpha \in \mathit{Free}(\mathcal{K})$—requires that free formulas do not play a role in assessing the inconsistency degree.

Example 4.10. Recall that the propositional knowledge base $\mathcal{K} = \{a,\, a \to b,\, \neg b,\, c,\, \neg c\}$ possesses $I_{min}(\mathcal{K}) = \{\{a,\, a \to b,\, \neg b\},\, \{c,\, \neg c\}\}$. Hence it does not contain any free formula since every formula occurs in a minimal inconsistent subset.

In order to generalize the postulate free-formula independence, let us first take a look at an alternative way to define free formulas. Assume we are given a monotonic logic. Recall Reiter's hitting set duality (Reiter, 1987), i. e., $\mathcal{K} \setminus \mathcal{S} \in C_{max}(\mathcal{K})$ iff $\mathcal{S}$ is a minimal hitting set of $I_{min}(\mathcal{K})$. This strong connection between minimal inconsistent and maximal consistent subsets facilitates a definition of free formulas via the maximal consistent subsets of $\mathcal{K}$. Indeed,

$$\mathcal{K} \setminus \bigcup_{\mathcal{H} \in I_{min}(\mathcal{K})} \mathcal{H} = \bigcap_{\mathcal{H} \in C_{max}(\mathcal{K})} \mathcal{H} \tag{4}$$

holds and hence, a formula α is free iff it occurs in every maximal consistent subset of $\mathcal{K}$. We note that the intuition is the same: Since α can be added to any subset $\mathcal{H} \subseteq \mathcal{K}$ without introducing inconsistency, it occurs in any $\mathcal{H} \in C_{max}(\mathcal{K})$. For the purpose of our generalization to non-monotonic logics, we observe that (4) is a corollary of Reiter's hitting set duality, which has been generalized to arbitrary logics (Brewka et al., 2019) (Theorem 3.6). We thus expect a similar result when replacing $I_{min}(\mathcal{K})$ with $SI_{min}(\mathcal{K})$. Indeed:

Corollary 4.11. *Let $\mathcal{K}$ be a knowledge base. Then*

$$\mathcal{K} \setminus \bigcup_{\mathcal{H} \in SI_{min}(\mathcal{K})} \mathcal{H} = \bigcap_{\mathcal{H} \in C_{max}(\mathcal{K})} \mathcal{H}.$$

Proof. "$\subseteq$": Let $\alpha \in \mathcal{K} \setminus \bigcup_{\mathcal{H} \in SI_{min}(\mathcal{K})} \mathcal{H}$. Hence, α does not occur in any minimal hitting set of $SI_{min}(\mathcal{K})$ and thus, due to Theorem 3.6 of Brewka et al. (2019), it occurs in all maximal consistent sets $\mathcal{H} \in C_{max}(\mathcal{K})$.
"$\supseteq$": Now assume $\alpha \notin \mathcal{K} \setminus \bigcup_{\mathcal{H} \in SI_{min}(\mathcal{K})} \mathcal{H}$, i. e., $\alpha \in \mathcal{H}$ for a minimal strongly inconsistent set $\mathcal{H} \in SI_{min}(\mathcal{K})$. Hence, $\mathcal{H} \setminus \{\alpha\} \notin SI_{min}(\mathcal{K})$ and thus, there is a maximal consistent set $\mathcal{H}'$ with $\mathcal{H} \setminus \{\alpha\} \subseteq \mathcal{H}'$. This means $\alpha \notin \mathcal{H}'$ because otherwise, $\mathcal{H}'$ would contain a strongly inconsistent set. It follows that $\alpha \notin \bigcap_{\mathcal{H} \in C_{max}(\mathcal{K})} \mathcal{H}$. □

Corollary 4.11 suggests a very natural and smooth generalization of free formulas, which lifts the intuitive as well as the formal meaning with respect to both aspects: A free formula α does not introduce conflicts ("$\alpha \notin \mathcal{H}$ for each $\mathcal{H} \in SI_{min}(\mathcal{K})$"), with "conflict" of a non-monotonic knowledge base being a minimal strongly inconsistent subset; and α can be added to any subset of $\mathcal{K}$ without introducing inconsistency ("$\alpha \in \mathcal{H}$ for each $\mathcal{H} \in C_{max}(\mathcal{K})$"). So we define:

Definition 4.12. Let $\mathcal{K}$ be a knowledge base. A formula $\alpha \in \mathcal{K}$ is called *free with respect to strong inconsistency* (or *SI-free* or simply *free* if there is no risk of confusion) if

$$\alpha \in \mathcal{K} \setminus \bigcup_{\mathcal{H} \in SI_{min}(\mathcal{K})} \mathcal{H} = \bigcap_{\mathcal{H} \in C_{max}(\mathcal{K})} \mathcal{H}.$$

Denote by $\mathit{Free}_{SI}(\mathcal{K})$ the set of all SI-free formulas of $\mathcal{K}$.

Example 4.13. Consider our program P again:

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

As already discussed, $SI_{min}(P) = \{c \leftarrow \text{not } b.,\ \neg c \leftarrow \text{not } b.,\ a \leftarrow b.\}$. We thus obtain $\mathit{Free}_{SI}(P) = \{a \vee b.\}$. To see Corollary 4.11 at work we recall

$$
\begin{aligned}
C_{max}(P) = \{&\{a \vee b.,\ a \leftarrow b.,\ c \leftarrow \text{not } b.\}, \\
&\{a \vee b.,\ a \leftarrow b.,\ \neg c \leftarrow \text{not } b.\}, \\
&\{a \vee b.,\ c \leftarrow \text{not } b.,\ \neg c \leftarrow \text{not } b.\}\}.
\end{aligned}
$$

Since "$a \vee b.$" is the only formula occurring in all maximal consistent sets, we obtain $\bigcap_{H \in C_{max}(P)} H = \{a \vee b.\}$.

Example 4.14. For the AF represented by $R = \{(a,b),(b,c),(c,c)\}$ we obtain the set $\mathit{Free}_{SI}(R) = \{(b,c)\}$.

As expected, $\mathit{Free}_{SI}(\mathcal{K})$ generalizes $\mathit{Free}(\mathcal{K})$ to non-monotonic logics in the sense that they coincide in the monotonic case.

Proposition 4.15. *Let $\mathcal{K}$ be a monotonic knowledge base. Then, $\mathit{Free}(\mathcal{K}) = \mathit{Free}_{SI}(\mathcal{K})$.*

Proof. Due to monotonicity we have $I_{min}(\mathcal{K}) = SI_{min}(\mathcal{K})$. In particular,

$$
\mathit{Free}(\mathcal{K}) = \mathcal{K} \setminus \bigcup_{\mathcal{H} \in I_{min}(\mathcal{K})} \mathcal{H} = \mathcal{K} \setminus \bigcup_{\mathcal{H} \in SI_{min}(\mathcal{K})} \mathcal{H} = \mathit{Free}_{SI}(\mathcal{K}),
$$

which proves our claim. $\square$

Finally, let us mention that $\mathit{Free}_{SI}(\mathcal{K})$ can also be defined without explicitly mentioning minimality.

Proposition 4.16. *Let $\mathcal{K}$ be a knowledge base. If $\alpha \in \mathcal{K}$, then $\alpha \in \mathit{Free}_{SI}(\mathcal{K})$ iff*

$$
\forall \mathcal{H} \subseteq \mathcal{K}:\ \mathcal{H} \notin SI(\mathcal{K}) \Rightarrow \mathcal{H} \cup \{\alpha\} \notin SI(\mathcal{K}). \tag{5}
$$

Proof. The implication "$\Leftarrow$" is trivial, so we show "$\Rightarrow$": Assume (5) is wrong, i. e., there is a set $\mathcal{H} \notin \mathit{SI}(\mathcal{K})$ with $\mathcal{H} \cup \{\alpha\} \in \mathit{SI}(\mathcal{K})$. Then $\mathcal{H} \cup \{\alpha\}$ contains a minimal strongly inconsistent set $\mathcal{H}'$. Observe that $\alpha \in \mathcal{H}'$, because otherwise, $\mathcal{H}'$ has a consistent superset as it is the case for $\mathcal{H}$. Since $\mathcal{H}'$ is minimal, it holds that $\mathcal{H}' \setminus \{\alpha\} \notin \mathit{SI}_{min}(\mathcal{K})$, but $\mathcal{H}' \in \mathit{SI}_{min}(\mathcal{K})$ and thus, $\alpha \notin \mathit{Free}_{SI}(\mathcal{K})$. $\square$

The similarities between $\mathit{Free}(\mathcal{K})$ and $\mathit{Free}_{SI}(\mathcal{K})$ motivate a rationality postulate similar in spirit to *free formula independence*. Analogously, one might expect that adding an SI-free formula α to a knowledge base $\mathcal{K}$ will not increase the inconsistency degree of $\mathcal{K}$, since it does not introduce strongly inconsistent subsets. It could still resolve conflicts, motivating the following postulate, similar to *free formula dilution* (Mu et al., 2011):

SI-Free If $\alpha \in \mathit{Free}_{SI}(\mathcal{K})$, then $\mathcal{I}(\mathcal{K}) \leq \mathcal{I}(\mathcal{K} \setminus \{\alpha\})$.

However, SI-free formulas are not as well-behaving as free formulas in monotonic logics. The issue we need to take into account is that ordinary inconsistency of a subset $\mathcal{H} \subseteq \mathcal{K}$ merely depends on $\mathcal{H}$, where strong inconsistency is a property $\mathcal{H}$ has with respect to the whole knowledge base $\mathcal{K}$. Removing an SI-free formula $\alpha \in \mathcal{K}$ might thus change the structure of $\mathit{SI}_{min}(\mathcal{K})$ in an unexpected way. To see this, let us consider the following example.

Example 4.17. Let P be the following logic program:

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

The reader may verify that

$$\mathit{SI}_{min}(P) = \{\{a \leftarrow \text{not } a, b., \; b., \; c.\}\}.$$

In particular,

$$r_1 = d.$$
$$r_2 = a \leftarrow \text{not } c.$$

are in $\mathit{Free}_{SI}(P)$. However, removal of r_1 turns $\{d \leftarrow \text{not } d.\}$ into a strongly inconsistent subset, while removal of r_2 renders "$c.$" irrelevant regarding the conflicts of P, so

$$\mathit{SI}_{min}(P \setminus \{r_1, r_2\}) = \{\{a \leftarrow \text{not } a, b., \; b.\}, \; \{d \leftarrow \text{not } d.\}\}.$$

We make the following observations:

- the conflict "$d \leftarrow \text{not } d.$" appears,
- the conflict $\{a \leftarrow \text{not } a,\ b.,\ b.,\ c.\}$ does not rely on "$c.$" anymore since the option to infer a is removed,
- the number of minimal conflicts increased.

It is thus hard to predict what happens when an SI-free formula is removed from a given knowledge base. In particular, we do not have $\mathit{SI}_{min}(\mathcal{K}) = \mathit{SI}_{min}(\mathcal{K} \setminus \{\alpha\})$ for each $\alpha \in \mathit{Free}_{SI}(\mathcal{K})$ which means that not even $\mathcal{I}_{\mathsf{MSI}}$—a measure based on minimal strongly inconsistent subsets—satisfies the *SI-free* postulate (cf. Section 5 below). Another observation regarding $\mathit{Free}_{SI}(\mathcal{K})$ is relevant: Although free formulas are not supposed to participate in minimal conflicts, the set $\mathit{Free}_{SI}(\mathcal{K})$ itself is in general not consistent.

Example 4.18. Consider the logic program

$$P: \qquad a. \qquad \neg a. \qquad \leftarrow \text{not } a, \text{not } \neg a.$$

We see that $\mathit{SI}_{min}(P) = \{\{a.,\ \neg a.\}\}$ and hence, $\mathit{Free}_{SI}(P) = \{\leftarrow \text{not } a, \text{not } \neg a.\}$ is an inconsistent program.

The previous considerations suggest that this notion depends heavily on the particular knowledge base. We will thus continue by introducing a stronger notion.

For this, we consider an alternative to the definition of free formulas in a way that they "do not induce strong inconsistency". Let us have a look at the monotonic case again. Since a free formula α does not induce inconsistency, one can see that α satisfies

$$\forall \mathcal{H} \subseteq \mathcal{K}:\ \mathcal{H} \in C(\mathcal{K}) \Rightarrow \mathcal{H} \cup \{\alpha\} \in C(\mathcal{K}). \tag{6}$$

In a monotonic framework, (6) formalizes that α is irrelevant regarding conflicts of $\mathcal{K}$ as it cannot turn a consistent set $\mathcal{H} \subseteq \mathcal{K}$ into an inconsistent one. In a non-monotonic logic, α could resolve conflicts, so we need to strengthen the condition:

$$\forall \mathcal{H} \subseteq \mathcal{K}:\ \mathcal{H} \in C(\mathcal{K}) \Leftrightarrow \mathcal{H} \cup \{\alpha\} \in C(\mathcal{K}). \tag{7}$$

This motivates the following definition.

Definition 4.19. Let $\mathcal{K}$ be a knowledge base. A formula $\alpha \in \mathcal{K}$ is called *neutral* if it satisfies

$$\forall \mathcal{H} \subseteq \mathcal{K} : \mathcal{H} \in C(\mathcal{K}) \Leftrightarrow \mathcal{H} \cup \{\alpha\} \in C(\mathcal{K}).$$

The neutral formulas in $\mathcal{K}$ are denoted by $Ntr(\mathcal{K})$.

It is easy to see that both notions coincide for monotonic logics.

Proposition 4.20. *If $\mathcal{K}$ is monotonic, then $Ntr(\mathcal{K}) = Free(\mathcal{K})$.*

Proof. This is clear due to Lemma 2.29. □

Note that in general, Ntr is a stronger notion than $Free_{SI}$.

Proposition 4.21. *If $\mathcal{K}$ is a knowledge base, then $Ntr(\mathcal{K}) \subseteq Free_{SI}(\mathcal{K})$*

Proof. Let $\alpha \in Ntr(\mathcal{K})$. Due to (6), it can be added to any set $\mathcal{H} \subseteq \mathcal{K}$ without introducing inconsistency. Hence, $\alpha \in \bigcap_{\mathcal{H} \in C_{max}(\mathcal{K})} \mathcal{H} = Free_{SI}(\mathcal{K})$. □

We note that in contrast to SI-free formulas, the neutral ones do not make use of strong inconsistency. Even though the hitting set duality from Theorem 3.6 of Brewka et al. (2019) suggests to utilize this notion, the neutral formulas are still quite well-behaving. The reason is that neutral formulas do not depend as much on the structure of the knowledge base and vice versa, do not influence $\mathcal{K}$ and in particular the structure of $SI_{min}(\mathcal{K})$.

Proposition 4.22. *Let $\mathcal{K}$ be a knowledge base and let $\alpha \in Ntr(\mathcal{K})$. Then,*

$$SI_{min}(\mathcal{K}) = SI_{min}(\mathcal{K} \setminus \{\alpha\}).$$

Proof. By definition of $Ntr(\mathcal{K})$ we have $\mathcal{H} \in SI(\mathcal{K})$ if and only if $\mathcal{H} \in SI(\mathcal{K} \setminus \{\alpha\})$ for any set $\mathcal{H} \subseteq \mathcal{K} \setminus \{\alpha\}$. Hence, the claim follows since no set $\mathcal{H} \in SI_{min}(\mathcal{K})$ contains α. □

Moreover, in contrast to $Free_{SI}(\mathcal{K})$, the neutral formulas always form consistent subsets of a knowledge base, as long as the empty knowledge base is considered consistent. Without proof, we state the following obvious fact:

Proposition 4.23. *If L is a logic such that $\emptyset$ is consistent and $\mathcal{K}$ a knowledge base of L, then $Ntr(\mathcal{K})$ is consistent.*

As before, we expect a neutral formula $\alpha \in \mathcal{K}$ not to increase the inconsistency degree of $\mathcal{K}$ since it does not induce inconsistency to any subset $\mathcal{H} \subseteq \mathcal{K}$. By definition, resolving conflicts is impossible as well, motivating:

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

As *neutral* is a quite strong property, our *independence* postulate seems to be rather basic. However, note that *independence* is a generalization of *free formula independence* (Hunter and Konieczny, 2010) which is not free from criticism. For example, it has been noted that the knowledge base $\mathcal{K} = \{a \wedge c, b \wedge \neg c\}$ should be considered less problematic than $\mathcal{K} \cup \{\neg a \vee \neg b\}$ even though the additional formula is free (Besnard, 2014). We want to emphasize that the same concerns apply to *independence*. A weaker version can be found in Thimm (2013), where a formula $\alpha \in \mathcal{K}$ is called *safe* if the atoms in α do not occur elsewhere in $\mathcal{K} \setminus \{\alpha\}$. The corresponding postulate *weak independence* is similar to *free formula independence*. However, this requirement is hard to phrase for an arbitrary logic L as in Definition 2.13. This suggests that weaker versions of *independence* should be tailored for a specific framework like the postulate *safe-rule independence* for answer set programming (see Section 8.1).

The final rationality postulate belonging here is *dominance* (Hunter and Konieczny, 2010). In the propositional setting, *dominance* requires that for two formulas α and β such that α is satisfiable and $\alpha \models \beta$, then $\mathcal{I}(\mathcal{K} \cup \{\alpha\}) \geq \mathcal{I}(\mathcal{K} \cup \{\beta\})$ should hold. The postulate formalizes that α carries more information and is hence more likely to be involved in conflicts than β. Of the postulates we considered so far, it is probably the most disputed one (Besnard, 2014; Jabbour et al., 2014). One of the most notable problems with this postulate was pointed out by De Bona and Hunter (2017). They state the following is true for a propositional knowledge base $\mathcal{K}$: If $\mathcal{I}$ satisfies *monotonicity* and *dominance*, then $\mathcal{I}(\mathcal{K}) = \mathcal{I}(\mathcal{K} \cup \{\beta\})$ if $\alpha \models \beta$ for a satisfiable formula $\alpha \in \mathcal{K}$. They continue with describing the following case: Say $\mathcal{K} = \{\alpha_1, \ldots, \alpha_n\}$ where each α_i is satisfiable and let $\mathcal{K}' = \{\beta_1, \ldots, \beta_n\}$ where $\beta_i \equiv \alpha_i$ for each i. Then, *monotonicity* and *dominance* already imply $\mathcal{I}(\mathcal{K}) = \mathcal{I}(\mathcal{K} \cup \mathcal{K}')$, so copying all conflicts does not change the inconsistency degree. They thus propose a refined version *dominance'* of *dominance*, where $\alpha, \beta \notin \mathcal{K}$ is additionally required.

For our setting of an arbitrary logic, distinguishing between *dominance* and *dominance'* does not make a significant difference, and we do not need to be too concerned about the above objections. The reason is simply that there is a much more fundamental issue with this postulate here. Since adding information to a knowledge base may not only induce, but also potentially resolve conflicts, the intuition does not hold anymore: There is simply no reason why α, which carries more information than β, should be considered more problematic in general. We thus believe there is no meaningful generalization of *dominance* for arbitrary (in particular non-monotonic) logics.

4.2. Extended Postulates

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 $\{a, b\}$. To illustrate this issue, let us recall the lottery paradox from above.

Example 4.24. We considered the knowledge base $\mathcal{K}_n = \{t_1 \vee \ldots \vee t_n, \neg t_1, \ldots, \neg t_n\}$ and argued that the inconsistency degree of $\mathcal{K}_n$ should be lower the bigger n is. This was due to the number of formulas required in order to obtain a contradiction. However, if we express $\mathcal{K}_n$ as the two formulas $\mathcal{K}'_n = \{t_1 \vee \ldots \vee t_n, \neg t_1 \wedge \ldots \wedge \neg t_n\}$, then the single minimal inconsistent set of $\mathcal{K}'$ contains two formulas, independent of n.

One could now argue that even when considering $\mathcal{K}'_n$, the number n of tickets still effects the *size* of the formulas within $\mathcal{K}'_n$; but then again, taking the size of a formula into account raises some other issues: It enforces distinguishing equivalent formulas depending on how they are written down. There are thus rationality postulates in the literature that are concerned about the behavior of inconsistency measures when dealing with equivalent formulas resp. equivalent knowledge bases. Of course, it is desirable that a measure $\mathcal{I}$ is robust wrt. the syntax of $\mathcal{K}$.

The postulate *adjunction invariance* (Besnard, 2014) formalizes the idea that there should be no difference between $\{a \wedge b\}$ and $\{a, b\}$, i. e., $\mathcal{I}(\mathcal{K} \cup \{a \wedge b\}) = \mathcal{I}(\mathcal{K} \cup \{a, b\})$. There are more postulates considering situations where (parts of) semantically equivalent knowledge bases are compared (Thimm, 2013).

In non-monotonic 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 non-monotonicity. This observation has led to the development of strong equivalence (Eiter et al., 2005; Lifschitz et al., 2001; Oikarinen and Woltran, 2011). Strong equivalence can be generalized to arbitrary logics in the following way (Brewka et al., 2019):

Definition 4.25. Let $L = (\mathsf{WF}, \mathsf{BS}, \mathsf{INC}, \mathsf{ACC})$ be a logic. The knowledge bases $\mathcal{K}$ and $\mathcal{K}'$ are *strongly equivalent*, denoted by $\mathcal{K} \equiv_s \mathcal{K}'$, if $\mathsf{ACC}(\mathcal{K} \cup \mathcal{G}) = \mathsf{ACC}(\mathcal{K}' \cup \mathcal{G})$ for each knowledge base $\mathcal{G}$ of L.

Thus, if a subset $\mathcal{H}$ of a knowledge base $\mathcal{K}$ is strongly equivalent to a set $\mathcal{H}'$ then $\mathcal{H}$ can be replaced in $\mathcal{K}$ by $\mathcal{H}'$ without changing the inferences one can draw from $\mathcal{K}$. This also means that $\mathcal{H}$ and $\mathcal{H}'$ should be interchangeably when it comes to the inconsistency they contribute to $\mathcal{K}$. By generalizing this idea to the whole knowledge base, we obtain that strongly equivalent knowledge bases should have the same degree of inconsistency.

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

However, whether or not this is desirable depends on the framework under consideration. In many cases, this postulate does not make sense. For example, in monotonic logics we have $\mathcal{K} \equiv_s \mathcal{K}'$ for any two inconsistent knowledge bases $\mathcal{K}$ and $\mathcal{K}'$, thus satisfying *strong equivalence* contradicts the idea of *quantitatively* assessing the inconsistency of a knowledge base. In ASP it still allows to distinguish between, e.g., $P = \{a.\ \neg a.\}$ and $P' = \{a \leftarrow \text{not } b.\ \neg a \leftarrow \text{not } b.\}$ as they are both inconsistent, but not strongly equivalent.

The issue with *strong equivalence* is quite straightforward: Consideration of the whole knowledge base is not fine-grained enough. One should look at the single formulas within $\mathcal{K}$ instead. This allows to compare equivalent and in particular *consistent* parts of a knowledge base. The technique we utilize is similar to the approach of Thimm (2013) for the postulate "irrelevance of syntax". For our setting we define:

Definition 4.26. 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 $\rho : \mathcal{K} \to \mathcal{K}'$ such that $\{\alpha\} \equiv_s \{\rho(\alpha)\}$ holds for all $\alpha \in \mathcal{K}$.

Equipped with this notion we may phrase a refinement of *strong equivalence*. Instead of requiring $\mathcal{K} \equiv_s \mathcal{K}'$, we consider two formula-wise strongly equivalent knowledge bases which yields a more meaningful rationality postulate. We thus obtain the following generalization of *irrelevance of syntax* (Thimm, 2013) (FW=formula-wise):

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

In contrast to *strong equivalence*, the postulate comes with a quite strong premise. To illustrate this, let us mention that $\mathcal{K} \equiv_\alpha \mathcal{K}'$ induces the same property for *any* subset of $\mathcal{K}$ and $\mathcal{K}'$.

Proposition 4.27. *If $\mathcal{K}$ and $\mathcal{K}'$ are formula-wise strongly equivalent, then there is a bijection $\tilde{\rho} : 2^{\mathcal{K}} \to 2^{\mathcal{K}'}$ such that $\mathcal{H} \equiv_s \tilde{\rho}(\mathcal{H})$ for any $\mathcal{H} \subseteq \mathcal{K}$. In particular, $|\mathcal{H}| = |\tilde{\rho}(\mathcal{H})|$.*

Proof. By assumption, there is a bijection $\rho : \mathcal{K} \to \mathcal{K}'$ with $\{\alpha\} \equiv_s \{\rho(\alpha)\}$ for all $\alpha \in \mathcal{K}$. So let $\tilde{\rho} : 2^{\mathcal{K}} \to 2^{\mathcal{K}'}$ be the mapping with

$$\tilde{\rho}(\mathcal{H}) := \bigcup_{\alpha \in \mathcal{H}} \{\rho(\alpha)\}.$$

Then the claim follows by induction from

$$\gamma \equiv_s \gamma' \wedge \delta \equiv_s \delta' \Rightarrow \{\gamma, \delta\} \equiv_s \{\gamma', \delta'\},$$

which is easy to see. □

A further refinement of this notion is to consider the *replacement* of a formula α with a strongly equivalent formula α'. Note that this postulate is similar to *exchange* (Besnard, 2014).

Strong Equivalent Replacement If $\{\alpha\} \equiv_s \{\alpha'\}$ and $\alpha \notin \mathcal{K}$ as well as $\alpha' \notin \mathcal{K}$, then $\mathcal{I}(\mathcal{K} \cup \{\alpha\}) = \mathcal{I}(\mathcal{K} \cup \{\alpha'\})$.

To conclude this discussion on extended postulates, let us consider two final ones which are concerned about modularisation of $\mathcal{K}$. The first one 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 independent, the inconsistency value of their union should decompose as the sum of the individual values.

As in the propositional case, a measure satisfying *separability* also satisfies *independence*.

Proposition 4.28. *Let L be a logic such that $\emptyset$ is a consistent knowledge base. Let $\mathcal{I}$ be an inconsistency measure satisfying* separability *and* consistency*. Then $\mathcal{I}$ satisfies* independence*.*

Proof. Assume $\mathcal{I}$ satisfies *separability*, i.e., $SI_{min}(\mathcal{K} \cup \mathcal{K}') = SI_{min}(\mathcal{K}) \cup SI_{min}(\mathcal{K}')$ and $SI_{min}(\mathcal{K}) \cap SI_{min}(\mathcal{K}') = \emptyset$ imply $\mathcal{I}(\mathcal{K} \cup \mathcal{K}') = \mathcal{I}(\mathcal{K}) + \mathcal{I}(\mathcal{K}')$. Now let $\mathcal{K}$ be a knowledge base and let $\alpha \in Ntr(\mathcal{K})$. By Proposition 4.22 we know that $SI_{min}(\mathcal{K}) = SI_{min}(\mathcal{K} \setminus \{\alpha\})$.

Set $\mathcal{H} = \mathcal{K} \setminus \{\alpha\}$ and $\mathcal{H}' = \{\alpha\}$. We have

$$SI_{min}(\mathcal{H} \cup \mathcal{H}') = SI_{min}(\mathcal{K}) = SI_{min}(\mathcal{K} \setminus \{\alpha\}) = SI_{min}(\mathcal{H}) \cup SI_{min}(\emptyset)$$

as well as

$$SI_{min}(\mathcal{H}) \cap SI_{min}(\mathcal{H}') = SI_{min}(\mathcal{H}) \cap \emptyset = \emptyset$$

Hence satisfaction of the *separibility* postulate yields

$$\mathcal{I}(\mathcal{H} \cup \mathcal{H}') = \mathcal{I}(\mathcal{H}) + \mathcal{I}(\mathcal{H}') = \mathcal{I}(\mathcal{H}) + \mathcal{I}(\{\alpha\}).$$

Since $\mathcal{I}$ satisfies *consistency* and $\{\alpha\}$ is consistent (see Proposition 4.23) we conclude $\mathcal{I}(\{\alpha\}) = 0$, turning the above equation into

$$\mathcal{I}(\mathcal{H} \cup \mathcal{H}') = \mathcal{I}(\mathcal{H}).$$

By Definition of $\mathcal{H}$ this yields

$$\mathcal{I}(\mathcal{K}) = \mathcal{I}(\mathcal{K} \setminus \{\alpha\}),$$

i.e., *independence.* □

Finally, we end our investigation with a generalization of *monotonicity*, namely *supper-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 $\mathcal{K}$ and $\mathcal{K}'$ are disjoint. As for *strong monotonicity*, we need to take into account that adding information might resolve conflicts in non-monotonic 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 $\mathcal{I}(\mathcal{K}) + \mathcal{I}(\mathcal{K}') \leq \mathcal{I}(\mathcal{K} \cup \mathcal{K}')$.

5. Analysis

As already mentioned, we are going to discuss the behavior of the measures with respect to the introduced postulates. For postulates that are not satisfied by a particular measure in general, we give counterexamples within the logic L^{ASP}. We also briefly discuss relations between the measures in terms of order compatibility (Grant and Hunter, 2011) and their relation to inconsistency graphs (De Bona et al., 2018).

5.1. Compliance with Rationality Postulates

In general, we obtain the following result on the compliance of our measures with the rationality postulates, see also the table below.

Proposition 5.1. *The measures $\mathcal{I}_{\mathsf{MSI}}$, $\mathcal{I}_{\mathsf{MSI^C}}$ and $\mathcal{I}_{\mathsf{p}}$ satisfy* consistency*,* strong monotonicity*,* independence*,* FW-strong equivalence*,* strong equivalent replacement*, and* strong super-additivity*. The measures $\mathcal{I}_{\mathsf{MSI}}$ and $\mathcal{I}_{\mathsf{MSI^C}}$ also satisfy* separability*.*

Proof. Consistency: Since $SI_{min}(\mathcal{K}) = \emptyset$ if and only if $\mathcal{K}$ is consistent, $\mathcal{I}_{\mathsf{MSI}}$, $\mathcal{I}_{\mathsf{MSI^C}}$ and $\mathcal{I}_{\mathsf{p}}$ satisfy *consistency*.

Strong monotonicity: Let $\mathcal{K}$ and $\mathcal{K}'$ be knowledge bases such that $\mathcal{K}'$ preserves conflicts of $\mathcal{K}$. Let $\mathcal{H} \in SI_{min}(\mathcal{K})$. By Proposition 4.4, we have $\mathcal{H} \in SI_{min}(\mathcal{K} \cup \mathcal{K}')$. Hence, we see $\mathcal{I}_{\mathsf{MSI}}(\mathcal{K}) \leq \mathcal{I}_{\mathsf{MSI}}(\mathcal{K} \cup \mathcal{K}')$, $\mathcal{I}_{\mathsf{MSI^C}}(\mathcal{K}) \leq \mathcal{I}_{\mathsf{MSI^C}}(\mathcal{K} \cup \mathcal{K}')$ and $\mathcal{I}_{\mathsf{p}}(\mathcal{K}) \leq \mathcal{I}_{\mathsf{p}}(\mathcal{K} \cup \mathcal{K}')$ follow straightforwardly.

Independence: Let $\alpha \in Ntr(\mathcal{K})$. Then we have $SI_{min}(\mathcal{K}) = SI_{min}(\mathcal{K} \setminus \{\alpha\})$ according to Proposition 4.22. It follows that $\mathcal{I}_{\mathsf{MSI}}$, $\mathcal{I}_{\mathsf{MSI^C}}$ and $\mathcal{I}_{\mathsf{p}}$ satisfy *independence*.

FW-strong equivalence: Let $\mathcal{K}$ and $\mathcal{K}'$ be such that $\mathcal{K} \equiv_{\alpha} \mathcal{K}'$. Proposition 4.27 implies that there is a bijection $\tilde{\rho} : 2^{\mathcal{K}} \to 2^{\mathcal{K}'}$ such that $\mathcal{H} \equiv_s \tilde{\rho}(\mathcal{H})$ for any $\mathcal{H} \subseteq \mathcal{K}$. Furthermore, observe that if $\mathcal{H}$ is strongly $\mathcal{K}$-inconsistent then any $\mathcal{H}'$ with $\mathcal{H}' \equiv_s \mathcal{H}$ is strongly $\mathcal{K} \setminus \mathcal{H} \cup \mathcal{H}'$-inconsistent. It follows that $\mathcal{H} \in SI_{min}(\mathcal{K})$ if and only if $\tilde{\rho}(\mathcal{H}) \in SI_{min}(\mathcal{K}')$. Since $|\mathcal{H}| = |\tilde{\rho}(\mathcal{H})|$ is also guaranteed in Proposition 4.27, $\mathcal{I}_{\mathsf{MSI}}$, $\mathcal{I}_{\mathsf{MSI^C}}$ and $\mathcal{I}_{\mathsf{p}}$ satisfy *FW-strong equivalence*.

Strong equivalent replacement: Similar.

Strong super-additivity: Let $\mathcal{K}'$ and $\mathcal{K}$ preserve each other's conflicts and $\mathcal{K} \cap \mathcal{K}' = \emptyset$. Proposition 4.4 implies $SI_{min}(\mathcal{K}) \cup SI_{min}(\mathcal{K}') \subseteq SI_{min}(\mathcal{K} \cup \mathcal{K}')$. Since $\mathcal{K} \cap \mathcal{K}' = \emptyset$ yields $SI_{min}(\mathcal{K}) \cap SI_{min}(\mathcal{K}') = \emptyset$ we see that the measures satisfy *strong super-additivity*.

Separability: Straightforward for $\mathcal{I}_{\mathsf{MSI}}$ and $\mathcal{I}_{\mathsf{MSI^C}}$. □

As already mentioned in Section 4, *SI-free* is not satisfied by any of the measures.

Example 5.2. Consider the program P given as follows:

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

	$\mathcal{I}_{\mathsf{MSI}}$	$\mathcal{I}_{\mathsf{MSI^C}}$	$\mathcal{I}_{\mathsf{p}}$
Consistency	✓	✓	✓
Strong monotonicity	✓	✓	✓
SI-Free	✗	✗	✗
Independence	✓	✓	✓
Strong Equivalence	✗	✗	✗
FW-Strong Equivalence	✓	✓	✓
Strong Equivalent Replacement	✓	✓	✓
Separability	✓	✓	✗
Strong Super-Additivity	✓	✓	✓

Table 1: Compliance of measures with rationality postulates

We have $r = a \leftarrow \text{not } c, \text{not } d. \in \mathit{Free}_{SI}(P)$: the rule "$a \leftarrow$ 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., $\mathcal{I}_{\mathsf{MSI}}(P) = 2$, $\mathcal{I}_{\mathsf{MSI^C}}(P) = \frac{2}{3}$ and $\mathcal{I}_{\mathsf{p}}(P) = 4$. However $SI_{min}(P \setminus \{r\}) = \{a \leftarrow \text{not } a, b., b.\}$, i. e., $\mathcal{I}_{\mathsf{MSI}}(P \setminus \{r\}) = 1$, $\mathcal{I}_{\mathsf{MSI^C}}(P \setminus \{r\}) = \frac{1}{2}$ and $\mathcal{I}_{\mathsf{p}}(P \setminus \{r\}) = 2$.

Example 5.3. Consider the programs P and P' given via

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

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 $\mathcal{I} \in \{\mathcal{I}_{\mathsf{MSI}}, \mathcal{I}_{\mathsf{MSI^C}}, \mathcal{I}_{\mathsf{p}}\}$ thus showing that *strong equivalence* is violated by all three measures.

A counterexample for *strong equivalence* is easy to find since $\mathcal{K} \equiv_s \mathcal{K}'$ for any two inconsistent propositional knowledge bases. For a counterexample of *separability* wrt. $\mathcal{I}_{\mathsf{p}}$ see Thimm (2017) (already in the propositional case).

Observe that for those postulates that are generalizations of classical ones—i. e., *consistency*, *strong monotonicity*, *independence*, *strong super-additivity*, and *separability*—the compliance of our three measures generalizes their compliance with the corresponding postulates in the classical case (Thimm, 2017). The results are summarized in Table 1.

5.2. Further Aspects

We investigate the measures we introduced wrt. two further aspects from the literature. First, we recall the notion of an IG measure (De Bona et al., 2018). Afterwards, we discuss *order comparability* of our measures. Both aspects can be seen as straight generalizations from the propositional setting.

IG measures. The notion of the *inconsistency graph* is utilized in order to classify inconsistency measures (De Bona et al., 2018). Let us briefly recall the required notions.

Definition 5.4. An *inconsistency graph* for a monotonic knowledge base $\mathcal{K}$ is a bipartite graph $\mathsf{IG}(\mathcal{K}) = (U, V, E)$ such that there are bijections $f_U : U \to \bigcup I_{min}(\mathcal{K})$ as well as $f_V : V \to I_{min}(\mathcal{K})$ with $E = \{\{u, v\} \mid f_U(u) \in f_V(v)\}$.

Then, the class of so called IG measures is defined: A measure $\mathcal{I}$ is IG if it can be written as $\mathcal{I}(\mathcal{K}) = f(\mathsf{IG}(\mathcal{K}))$ for a mapping f assigning non-negative real values to inconsistency graphs.

Example 5.5. Consider the knowledge base $\mathcal{K} = \{a,\ a \to b,\ \neg b,\ c,\ \neg c\}$. Observe that

$$I_{min}(\mathcal{K}) = \{\{a,\ a \to b,\ \neg b\}, \{c,\ \neg c\}\}.$$

The graph $\mathsf{IG}(\mathcal{K}) = (U, V, E)$ is depicted in Figure 3. It is easy to see that the measure $\mathcal{I}_{\mathsf{MI}}(\mathcal{K}) = |I_{min}(\mathcal{K})|$ is an IG measure since $\mathcal{I}_{\mathsf{MI}}(\mathcal{K}) = |V|$.

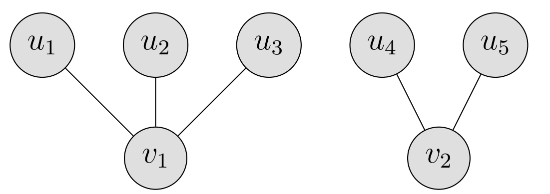


Figure 3: The IG graph of $\mathcal{K}$ from Example 5.5.

It is quite clear that the *inconsistency graph* is not appropriate for non-monotonic logics. We thus consider the straightforward refinement we need.

Definition 5.6. A *strong inconsistency graph* for an arbitrary knowledge base $\mathcal{K}$ is a bipartite graph $\mathsf{SIG}(\mathcal{K}) = (U, V, E)$ such that there are bijections $f_U : U \to \bigcup SI_{min}(\mathcal{K})$ and $f_V : V \to SI_{min}(\mathcal{K})$ with $E = \{\{u, v\} \mid f_U(u) \in f_V(v)\}$.

We define SIG measures in the canonical way. As a corollary of Proposition 1 of De Bona et al. (2018), we see:

Proposition 5.7. *The measures $\mathcal{I}_{\mathsf{MSI}}$, $\mathcal{I}_{\mathsf{MSI^C}}$ and $\mathcal{I}_{\mathsf{p}}$ are SIG measures.*

More precisely, it is easy to see that $\mathcal{I}_{\mathsf{MSI}}$, $\mathcal{I}_{\mathsf{MSI^C}}$ and $\mathcal{I}_{\mathsf{p}}$ can be obtained from the graph SIG the same way their propositional counterparts can be obtained from the graph IG. For example, given $\mathsf{SIG}(\mathcal{K}) = (U, V, E)$, we have $\mathcal{I}_{\mathsf{MSI}}(\mathcal{K}) = |V|$ analogously to propositional logic (see Example 5.5).

De Bona et al. (2018) are mostly concerned about classifying inconsistency measures. For the measures considered in the present section, this classification is quite obvious: All of them are SIG measures. The classification might be more insightful (and more challenging) when it comes to further inconsistency measures, e. g., ones that are specifically tailored for certain logics. We believe this is a promising research direction for future work.

Order compatibility. In order to compare inconsistency measures we can use the notion of order compatibility (Grant and Hunter, 2011). We say that two inconsistency measures $\mathcal{I}_1$ and $\mathcal{I}_2$ are *order-compatible* if $\mathcal{I}_1(\mathcal{K}) \leq \mathcal{I}_1(\mathcal{K}')$ iff $\mathcal{I}_2(\mathcal{K}) \leq \mathcal{I}_2(\mathcal{K}')$ for all knowledge bases $\mathcal{K}, \mathcal{K}'$. So $\mathcal{I}_1$ and $\mathcal{I}_2$ induce the same ranking on knowledge bases without necessarily assigning the same inconsistency values. As a corollary of the corresponding result from for the propositional case (Grant and Hunter, 2011), we obtain:

Proposition 5.8. *The measures $\mathcal{I}_{\mathsf{MSI}}$, $\mathcal{I}_{\mathsf{MSI^C}}$ and $\mathcal{I}_{\mathsf{p}}$ are pairwise not order-compatible.*

Hence, all measures are incompatible and provide different points of view on inconsistency.

6. Computational Complexity

We now address the complexity of several computational problems related to the measures we considered in this paper. Following Thimm and Wallner (2016; 2019), we consider the three decision problems $\textsc{Exact}_{\mathcal{I}}$, $\textsc{Upper}_{\mathcal{I}}$, $\textsc{Lower}_{\mathcal{I}}$, and the natural function problem $\textsc{Value}_{\mathcal{I}}$. Let $L = (\mathsf{WF}, \mathsf{ACC}, \mathsf{BS}, \mathsf{INC})$ be a logic.

$\textsc{Exact}^L_{\mathcal{I}}$	**Input**:	$\mathcal{K} \subseteq \mathsf{WF}$, $x \in [0,\infty]$
	Output:	$\textsc{true}$ iff $\mathcal{I}(\mathcal{K}) = x$
$\textsc{Upper}^L_{\mathcal{I}}$	**Input**:	$\mathcal{K} \subseteq \mathsf{WF}$, $x \in [0,\infty]$
	Output:	$\textsc{true}$ iff $\mathcal{I}(\mathcal{K}) \leq x$
$\textsc{Lower}^L_{\mathcal{I}}$	**Input**:	$\mathcal{K} \subseteq \mathsf{WF}$, $x \in (0,\infty]$
	Output:	$\textsc{true}$ iff $\mathcal{I}(\mathcal{K}) \geq x$
$\textsc{Value}^L_{\mathcal{I}}$	**Input**:	$\mathcal{K} \subseteq \mathsf{WF}$
	Output:	The value of $\mathcal{I}(\mathcal{K})$

After giving necessary preliminaries concerning computational complexity, we start by giving some general membership results for the given problems. Then, we demonstrate that these bounds are tight in general by giving corresponding hardness results for the frameworks L^{ASP^*} and L^{ASP}.

6.1. Background

We assume the reader to be familiar with the classes P, NP and coNP. Furthermore, we consider the polynomial hierarchy as usual: We let $\Sigma^p_0 = \Pi^p_0 = \mathsf{P}$ and for any $m \geq 0$, $\Sigma^p_{m+1} = \mathsf{NP}^{\Sigma^p_m}$ and $\Pi^p_{m+1} = \mathsf{coNP}^{\Sigma^p_m}$. Thereby, as usual, $\mathcal{C}^{\mathcal{D}}$ is the class of decision problems solvable in $\mathcal{C}$ having access to an oracle for some problem that is complete in $\mathcal{D}$.

A *quantified Boolean formula* (QBF) Φ is a formula

$$\Phi = Q_1 X_1 \dots Q_m X_m \, \phi$$

with quantifiers $Q_1, \dots, Q_m \in \{\forall, \exists\}$, pair-wise disjoint sets of variables $X_1, \dots, X_m$, and a propositional formula ϕ over the variables $X_1 \cup \dots \cup X_m$. A QBF Φ is *true* if ϕ evaluates to true with respect to the quantifiers, e. g., $\forall x_1 \exists x_2 (x_1 \vee \neg x_2)$ is true as for every truth value of x_1 one can find a truth value of x_2 such that $x_1 \vee \neg x_2$ evaluates to true. A QBF Φ is in *prenex normal form* if the quantifiers $Q_1, \dots, Q_m$ alternate between $\forall$ and $\exists$. The problem of deciding whether a QBF Φ with m alternating quantifiers starting with $\exists$ (resp. starting with $\forall$) is true is the canonical Σ^p_m-complete (resp. Π^p_m-complete) problem (Papadimitriou, 1994).

We will also consider *open QBFs*. They are defined similar as QBFs, but also contain free variables. Thus, an open QBF is of the form

$$\Phi = \Phi(X) = Q_1 X_1 \dots Q_m X_m \, \phi(X, X_1, \dots, X_n).$$

If $\Phi = \Phi(X)$ is an open QBF, then $\mathsf{Mod}(\Phi)$ is the set of all models of Φ, i. e., the set of all assignments to the X-variables rendering ϕ true with respect to the quantifiers.

We also make use of the classes D^p_m, which are the classes of languages that are intersections of a language in Σ^p_m and a language in Π^p_m (Papadimitriou, 1994). For example, the generic D^p_1-complete problem is SAT-UNSAT, where we are given two propositional formulas ϕ_1 and ϕ_2, and have to decide whether ϕ_1 is satisfiable while ϕ_2 is not.

Examining $\mathcal{I}_{\mathsf{MSI}}$ leads to the consideration of counting complexity classes (Valiant, 1979). They are defined using witness functions w that assign words from an input alphabet Σ to finite subsets of an alphabet Γ. Given a string x from the alphabet Σ, the task is to return $|w(x)|$, i. e., the number of witnesses. Given a class $\mathcal{C}$ of decision problems, by $\#{\cdot}\mathcal{C}$ we denote the class of counting problems such that

- for every input string x, each $y \in w(x)$ is polynomially bounded,
- the decision problem "Is $y \in w(x)$?" is in $\mathcal{C}$.

For example, the generic $\#{\cdot}\mathsf{coNP}$-complete problem is counting $|\mathsf{Mod}(\Phi)|$ for an open QBF $\Phi = \forall Y \phi(X, Y)$. Here, $\mathsf{Mod}(\Phi)$ is the witness function assigning to a given formula the corresponding models. As required, each truth assignment is polynomial bounded and given an assignment to the X-variables, the decision problem whether $\forall Y \phi(X, Y)$ holds is in coNP.

Hardness results for $\textsc{Exact}^L_{\mathcal{I}_{\mathsf{MSI}}}$ are going to be given under subtractive reductions (Durand et al., 2005). For that, let $\#V$ and $\#W$ be counting problems with witness functions v and w. The problem $\#V$ reduces to $\#W$ under strong subtractive reductions if there are polynomial-time computable functions f and g such that for each input x we have $w(f(x)) \subseteq w(g(x))$ and $|v(x)| = |w(g(x))| - |w(f(x))|$. Subtractive reductions are the transitive closure of strong subtractive reductions.

We also make use of the counting polynomial hierarchy (Wagner, 1986). We start by defining the counting quantifier C. Given a predicate $R(x, y)$ with free variables x and y, let

$$\mathsf{C}^k_y R(x, y) :\Leftrightarrow |\{y \mid R(x, y) \text{ true }\}| \geq k. \tag{8}$$

Now for any class C of problems, A is in $\mathsf{C}C$ if there is a $B \in C$, a function f computable in P and a polynomial p such that

$$x \in A :\Leftrightarrow \mathsf{C}^{f(x)}_{|y| \leq p(|x|)} R(x, y).$$

where $|x|$ denotes the length of x. Hence there are at least $f(x)$ many (by a polynomial in x bounded) values y (whose length is bonded by a polynomial $p(|x|)$) such that a given predicate R holds for (x, y). Checking whether $R(x, y)$ is true shall be in B. We will also use the class $\mathsf{C_=NP}$ Wagner (1986) which is a variation of CNP where "$\geq$" is replaced by "$=$" in Equation (8).

Example 6.1. A generic complete problem for the class CP is the following: Given a propositional formula ϕ, is it true that $|\mathsf{Mod}(\phi)| \geq k$? Indeed, k is just a constant, assignments to the x-variables have polynomial length and checking whether ϕ holds given an assignment is in P.

To give a generic example for the class CcoNP, we make use of open QBFs: Given an open QBF $\Phi = \forall Y \phi(X, Y)$, is it true that $|\mathsf{Mod}(\Phi)| \geq k$?

First, we consider an arbitrary, possibly non-monotonic logic $L = (\mathsf{WF}, \mathsf{ACC}, \mathsf{BS}, \mathsf{INC})$. 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.

$\textsc{Sat}_L$	**Input**:	$\mathcal{K} \subseteq \mathsf{WF}$
	Output:	$\textsc{true}$ iff $\mathcal{K}$ is consistent

To keep this section within a reasonable amount of space, we restrict most of our discussion to the measure $\mathcal{I}_{\mathsf{MSI}}$.

6.2. Minimal Strong Inconsistency in General

Let us start with some general membership results which can be obtained dependent on the satisfiability check of the logic. Let us start by checking whether a given input integer is an upper bound for $\mathcal{I}_{\mathsf{MSI}}$.

Theorem 6.2. *Let $m \geq 1$. If the decision problem $\textsc{Sat}_L$ is in*

(a) Σ_m^p, *then* $\textsc{Upper}^L_{\mathcal{I}_{\mathsf{MSI}}}$ *is in* $\mathsf{C}\Sigma_m^p$,

(b) Π_m^p, *then* $\textsc{Upper}^L_{\mathcal{I}_{\mathsf{MSI}}}$ *is in* $\mathsf{C}\Sigma_{m+1}^p$,

(c) Π_m^p *and L is monotonic, then* $\textsc{Upper}^L_{\mathcal{I}_{\mathsf{MSI}}}$ *is in* $\mathsf{C}\Sigma_m^p$.

Proof.

(a) Given an integer k, $(\mathcal{K}, k)$ is a positive instance of $\textsc{Lower}^L_{\mathcal{I}_{\mathsf{MSI}}}$ if there are at least k minimal strongly $\mathcal{K}$-inconsistent sets. Due to Theorem 4.5 of Brewka et al. (2017), deciding whether $\mathcal{H} \subseteq \mathcal{K}$ is in $SI_{min}(\mathcal{K})$ is in D^p_m. Moreover, any $\mathcal{H} \subseteq \mathcal{K}$ is of polynomial bounded size. Due to $\mathsf{CD}^p_m = \mathsf{C}\Sigma^p_m$ (see Theorem 4 of Wagner (1986)[3]), $\textsc{Lower}^L_{\mathcal{I}_{\mathsf{MSI}}}$ is in $\mathsf{C}\Sigma^p_m$. Regarding $\textsc{Upper}^L_{\mathcal{I}_{\mathsf{MSI}}}$, again due to Theorem 4 of Wagner (1986), $\mathsf{C}\Sigma^p_m$ is closed under complement and $(\mathcal{K}, k)$ is a yes instance iff $(\mathcal{K}, k+1)$ is a no instance of $\textsc{Lower}^L_{\mathcal{I}_{\mathsf{MSI}}}$.

(b) Similar, note that deciding whether $\mathcal{H} \in SI_{min}(\mathcal{K})$ holds is in D^p_{m+1} due to Theorem 4.5 by Brewka et al. (2017).

(c) Similar, note that deciding whether $\mathcal{H} \in SI_{min}(\mathcal{K})$ holds is in D^p_m due to Theorem 4.5 by Brewka et al. (2017). □

When considering the proof of the above theorem, it becomes apparent that the complexity of a *counting* problem heavily depends on the complexity of the underlying decision problem. Besides general properties of the counting polynomial hierarchy (Wagner, 1986), the proof only makes use of the complexity results about verifying minimal strong inconsistency given in Theorem 4.5 of Brewka et al. (2017). The same is true for the function problem:

Theorem 6.3. *Let $m \geq 1$. If the decision problem $\textsc{Sat}_L$ is in*

(a) Σ^p_m, then $\textsc{Value}^L_{\mathcal{I}_{\mathsf{MSI}}}$ is in $\#\cdot\Pi^p_m$,

(b) Π^p_m, then $\textsc{Value}^L_{\mathcal{I}_{\mathsf{MSI}}}$ is in $\#\cdot\Pi^p_{m+1}$,

(c) Π^p_m and L is monotonic, then $\textsc{Value}^L_{\mathcal{I}_{\mathsf{MSI}}}$ is in $\#\cdot\Pi^p_m$.

Proof. We make use of the observation that

$$\#\cdot\Delta^p_{m+1} = \#\cdot\Pi^p_m$$

by Hemaspaandra and Vollmer (1995). Furthermore, it is clear that

$$\mathsf{D}^p_m \subseteq \Delta^p_{m+1}.$$

Due to Theorem 4.5 by Brewka et al. (2017), we obtain:

[3] More precisely, Theorem 4, item 5 of Wagner (1986) states $\mathsf{C}\Sigma^p_m = \mathsf{C}B(\Sigma^p_m)$ where $B(\Sigma^p_m)$ is the boolean closure of Σ^p_k implying $\mathsf{D}^p_m \subseteq B(\Sigma^p_m)$ and $\mathsf{CD}^p_m = \mathsf{C}\Sigma^p_m$.

(a) If $\textsc{Sat}_L$ is in Σ_m^p, then checking whether $\mathcal{H} \in SI_{min}(\mathcal{K})$ holds is in D_m^p and hence, $\textsc{Value}^L_{\mathcal{I}_{\mathsf{MSI}}}$ is in $\#\cdot\Delta_{m+1}^p = \#\cdot\Pi_m^p$.

Items (b) and (c) follow analogously from Theorem 4.5 of Brewka et al. (2017). □

This finishes our discussion regarding general membership results. As the reader may have already observed, they are quite simply corollaries of the results for the corresponding decision problems. We want to emphasize that these results are in accordance with results from the literature for monotonic logics. For example, it has been shown that computing $|I_{min}(\mathcal{K})|$ is in $\#\cdot\mathsf{coNP}$ for a propositional knowledge base $\mathcal{K}$ (Thimm and Wallner, 2016, Proposition 11). This is a special case of Theorem 6.3, item (a) with satisfiability check in $\Sigma_1^p = \mathsf{NP}$.

6.3. Hardness Results for Answer Set Programming

We are going to give some exemplary hardness results for the observations we made above by considering the concrete logics L^{ASP^*} (disjunction-free logic programs under the answer set semantics) and L^{ASP} (disjunctive logic programs under the answer set semantics). Recall that deciding whether a program P within the framework L^{ASP^*} is consistent is NP-complete in general, while the decision problem for programs P in L^{ASP} is Σ_2^p-complete (Eiter and Gottlob, 1995).

The following results show that, for L^{ASP^*}, the computational complexity of the problems we consider is similar to the results for the propositional case (Thimm and Wallner, 2019). This seems natural as the satisfiability check for propositional logic is NP-complete as well. As expected, considering L^{ASP} involves going up one level within the corresponding hierarchy. We will give some of the proofs resp. constructions in order to demonstrate the required techniques. However, in order to keep this section within a reasonable space, most of the proofs can be found in Appendix A.

As already mentioned, the groundwork for our hardness results is the following observation. It establishes a required link between minimal strongly inconsistent sets of a program P to models of an open QBF Φ.

Lemma 6.4. *Given an open QBF $\Phi = \forall Y\, \phi(X, Y)$, there is a disjunction-free logic program $P(\Phi) \subseteq \mathsf{WF}^{ASP^*}$ of polynomial size with*

$$|SI_{min}(P(\Phi))| = |X| + |\mathsf{Mod}(\Phi)|.$$

Now the desired hardness results for disjunction-free answer set programming follow via the above lemma. We start with $\textsc{Lower}^{L^{\mathrm{ASP}^*}}_{\mathcal{I}_{\mathsf{MSI}}}$ and $\textsc{Upper}^{L^{\mathrm{ASP}^*}}_{\mathcal{I}_{\mathsf{MSI}}}$.

Proposition 6.5. *The problems* $\textsc{Lower}^{L^{ASP^*}}_{\mathcal{I}_{\mathsf{MSI}}}$ *and* $\textsc{Upper}^{L^{ASP^*}}_{\mathcal{I}_{\mathsf{MSI}}}$ *are* CNP*-complete.*

Proof. Membership of $\textsc{Lower}_{L^{\mathrm{ASP}^*}}$ and $\textsc{Upper}_{L^{\mathrm{ASP}^*}}$ follows from Theorem 6.2, so we need to show hardness. We consider the CNP-complete problem of deciding whether $|\mathsf{Mod}(\Phi)| \geq k$ for an open QBF $\Phi = \forall Y\, \phi(X, Y)$ (recall $\mathsf{C}\Sigma^p_m = \mathsf{C}\Pi^p_m$, see Theorem 4 by Wagner (1986)). Since we can construct the program $P(\Phi)$ as in Lemma 6.4 in P, we already found a polynomial reduction: $\mathsf{Mod}(\Phi) \geq k$ if and only if $SI_{min}(P) \geq k + |X|$. □

Corollary 6.6. $\textsc{Exact}^{L^{ASP^*}}_{\mathcal{I}_{\mathsf{MSI}}}$ *is* $\mathsf{C_{=}NP}$*-hard.*

Next we show that $\textsc{Value}^{L^{\mathrm{ASP}^*}}_{\mathcal{I}_{\mathsf{MSI}}}$ is $\#\cdot\mathsf{coNP}$-complete under subtractive reductions. For $\#\cdot\mathsf{coNP}$, computing $|\mathsf{Mod}(\Phi)|$ for an open QBF $\Phi = \forall Y\, \phi(X, Y)$ is the generic complete problem under subtractive reductions. Recall the idea behind this kind of reduction: We first overcount the value we actually aim at. Then, we correct this by subtracting unintended items. Consequently, the above construction where we found a program $P = P(\Phi)$ satisfying $SI_{min}(P)| = |X| + |\mathsf{Mod}(\Phi)|$ is a suitable starting point for a subtractive reduction (corresponding to the "overcount" part). We have left to find a program P' with

- $SI_{min}(P') \subseteq SI_{min}(P)$,
- $\mathsf{Mod}(\Phi) = SI_{min}(P) - SI_{min}(P')$.

Given these programs P and P' we have shown that $\mathsf{Mod}(\Phi)$ can be computed for an open QBF $\Phi = \forall Y\, \phi(X, Y)$ via $|SI_{min}(P)| - |SI_{min}(P')|$ in $\#\cdot\mathsf{coNP}$ (Durand et al., 2005). This yields completeness under subtractive reductions. The proof which is based on Lemma 6.4 can be found in Appendix A.

Theorem 6.7. *The problem* $\textsc{Value}^{L^{ASP^*}}_{\mathcal{I}_{\mathsf{MSI}}}$ *is* $\#\cdot\mathsf{coNP}$*-complete under subtractive reductions.*

This ends our discussion pertaining to L^{ASP^*}. As already mentioned, the case L^{ASP} is similar but involves going up one level in the corresponding hierarchy. Analogously, the fundamental step is constructing a program P with $|SI_{min}(P(\Phi))| = |X| + |\mathsf{Mod}(\Phi)|$ for an open QBF $\Phi = \forall Y \exists Z\, \phi(X, Y, Z)$. The construction is rather similar: We augment our previous construction with features from the program in the proof of Theorem 3.1 by Eiter and Gottlob (1995) which is used to show Σ_2^p-completeness of the satisfiability check for disjunctive logic programs. Then, the subsequent steps are as above:

Lemma 6.8. *Given an open QBF $\Phi = \forall Y \exists Z\, \phi(X, Y, Z)$, there is a disjunctive logic program $P(\Phi) \subseteq \mathsf{WF}^{ASP}$ of polynomial size with*

$$|SI_{min}(P(\Phi))| = |X| + |\mathsf{Mod}(\Phi)|.$$

Proposition 6.9. *The problems* $\textsc{Lower}_{\mathcal{I}_{\mathsf{MSI}}}^{L^{ASP}}$ *and* $\textsc{Upper}_{\mathcal{I}_{\mathsf{MSI}}}^{L^{ASP}}$ *are $\mathsf{C}\Sigma_2^p$-complete. The problem* $\textsc{Exact}_{\mathcal{I}_{\mathsf{MSI}}}^{L^{ASP}}$ *is $\mathsf{C}_{=}\Sigma_2^p$-hard. The problem* $\textsc{Value}_{\mathcal{I}_{\mathsf{MSI}}}^{L^{ASP}}$ *is $\#\cdot\Pi_2^p$-complete under subtractive reductions.*

6.4. Summary of Results for $\mathcal{I}_{\mathsf{MSI^C}}$ and $\mathcal{I}_{\mathsf{p}}$

As already mentioned, we restrict most of the discussion in this section to the measure $\mathcal{I}_{\mathsf{MSI}}$. Hence most of the work is done. In order to complete the picture, we report the results established for the other measures briefly in this section. Proofs can be found in Appendix A.

Regarding general membership results we find the following for $\mathcal{I}_{\mathsf{p}}$.

Theorem 6.10. *Let $m \geq 1$. If the decision problem* $\textsc{Sat}_L$ *is in*

(a) Σ_m^p, *then* $\textsc{Lower}_{\mathcal{I}_{\mathsf{p}}}^{L}$ *is in* Σ_{m+1}^p,

(b) Π_m^p, *then* $\textsc{Lower}_{\mathcal{I}_{\mathsf{p}}}^{L}$ *is in* Σ_{m+2}^p,

(c) Π_m^p *and L is monotonic, then* $\textsc{Lower}_{\mathcal{I}_{\mathsf{p}}}^{L}$ *is in* Σ_{m+1}^p.

Within the scope of the proof techniques we worked with in the present paper, it does not appear to be straightforward to establish membership results for the measure $\mathcal{I}_{\mathsf{MSI^C}}$. This problem is thus left for future work. The results in Theorem 6.10 can however used to infer corresponding upper bounds for $\textsc{Upper}$, $\textsc{Exact}$ and $\textsc{Value}$ for the measure $\mathcal{I}_{\mathsf{p}}$:

Corollary 6.11. *Let $m \geq 81$. If the decision problem* $\textsc{Sat}_L$ *is in*

(a) Σ^p_m*, then* $\textsc{Upper}^L_{\mathcal{I}_\mathsf{p}}$ *is in* Π^p_{m+1}*,* $\textsc{Exact}^L_{\mathcal{I}_\mathsf{p}}$ *is in* D^p_{m+1} *and* $\textsc{Value}^L_{\mathcal{I}_\mathsf{p}}$ *is in* $\mathsf{FP}^{\Sigma^p_{m+1}[\log n]}$*,*

(b) Π^p_m*, then* $\textsc{Upper}^L_{\mathcal{I}_\mathsf{p}}$ *is in* Π^p_{m+2}*,* $\textsc{Exact}^L_{\mathcal{I}_\mathsf{p}}$ *is in* D^p_{m+2} *and* $\textsc{Value}^L_{\mathcal{I}_\mathsf{p}}$ *is in* $\mathsf{FP}^{\Sigma^p_{m+2}[\log n]}$*,*

(c) Π^p_m *and L is monotonic, then* $\textsc{Upper}^L_{\mathcal{I}_\mathsf{p}}$ *is in* Π^p_{m+1}*,* $\textsc{Exact}^L_{\mathcal{I}_\mathsf{p}}$ *is in* D^p_{m+1} *and* $\textsc{Value}^L_{\mathcal{I}_\mathsf{p}}$ *is in* $\mathsf{FP}^{\Sigma^p_{m+1}[\log n]}$*.*

Following the investigation of $\mathcal{I}_\mathsf{MSI}$, we also investigated lower bounds for ASP for the measures $\mathcal{I}_\mathsf{p}$ as well as $\mathcal{I}_{\mathsf{MSI}^\mathsf{C}}$. Interestingly, lower bounds for $\mathcal{I}_{\mathsf{MSI}^\mathsf{C}}$ can be established (and are similar to those of $\mathcal{I}_\mathsf{MSI}$). The measure $\mathcal{I}_\mathsf{p}$ induces much easier problems since it does not involve counting several subsets of a given knowledge base. The hardness results for ASP are reported in Table 2 as well as the propositions below.

	$\mathcal{I}_\mathsf{MSI}$	$\mathcal{I}_{\mathsf{MSI}^\mathsf{C}}$	$\mathcal{I}_\mathsf{p}$
$\textsc{Upper}^{L^{\mathrm{ASP}^*}_A}_{\mathcal{I}}$	CNP-c	CNP-h	Σ^p_2-c
$\textsc{Lower}^{L^{\mathrm{ASP}^*}_A}_{\mathcal{I}}$	CNP-c	CNP-h	Π^p_2-c
$\textsc{Exact}^{L^{\mathrm{ASP}^*}_A}_{\mathcal{I}}$	$\mathsf{C_{=}NP}$-h	$\mathsf{C_{=}NP}$-h	D^p_2-c
$\textsc{Upper}^{L^{\mathrm{ASP}}_A}_{\mathcal{I}}$	$\mathsf{C}\Sigma^p_2$-c	$\mathsf{C}\Sigma^p_2$-h	Σ^p_3-c
$\textsc{Lower}^{L^{\mathrm{ASP}}_A}_{\mathcal{I}}$	$\mathsf{C}\Sigma^p_2$-c	$\mathsf{C}\Sigma^p_2$-h	Π^p_3-c
$\textsc{Exact}^{L^{\mathrm{ASP}}_A}_{\mathcal{I}}$	$\mathsf{C_{=}}\Sigma^p_2$-h	$\mathsf{C_{=}}\Sigma^p_2$-h	D^p_3-c

Table 2: Hardness results for $L^{\mathrm{ASP}^*}_A$ and L^{ASP}_A

Disjunction-free logic programs yield the following lower bounds:

Proposition 6.12. *The problems* $\textsc{Lower}^{L^{ASP^*}}_{\mathcal{I}_{\mathsf{MSI}^\mathsf{C}}}$ *and* $\textsc{Upper}^{L^{ASP^*}}_{\mathcal{I}_{\mathsf{MSI}^\mathsf{C}}}$ *are* CNP*-hard,* $\textsc{Exact}^{L^{ASP^*}}_{\mathcal{I}_{\mathsf{MSI}^\mathsf{C}}}$ *is* $\mathsf{C_{=}NP}$*-hard.*

Proposition 6.13. *The problem* $\textsc{Lower}^{L^{ASP^*}}_{\mathcal{I}_\mathsf{p}}$ *is* Σ^p_2*-complete. The problem* $\textsc{Upper}^{L^{ASP^*}}_{\mathcal{I}_\mathsf{p}}$ *is* Π^p_2*-complete. The problem* $\textsc{Exact}^{L^{ASP^*}}_{\mathcal{I}_\mathsf{p}}$ *is* D^p_2*-complete.*

As expected, the same problems for disjunctive logic programs require moving up one level within the polynomial hierarchy.

Proposition 6.14. *The problems* $\textsc{Lower}^{L^{ASP}}_{\mathcal{I}_{\mathsf{MSI^C}}}$ *and* $\textsc{Upper}^{L^{ASP}}_{\mathcal{I}_{\mathsf{MSI^C}}}$ *are* $\mathsf{C}\Sigma^p_2$-*hard,* $\textsc{Exact}^{asplang}_{\mathcal{I}_{\mathsf{MSI^C}}}$ *is* $\mathsf{C}_{=}\Sigma^p_2$-*hard.*

We proceed similar as for L^{ASP^*} and obtain:

Proposition 6.15. *The problem* $\textsc{Lower}^{L^{ASP}}_{\mathcal{I}_{\mathsf{p}}}$ *is* Σ^p_3-*complete,* $\textsc{Upper}^{L^{ASP}}_{\mathcal{I}_{\mathsf{p}}}$ *is* Π^p_3-*complete and* $\textsc{Exact}^{L^{ASP}}_{\mathcal{I}_{\mathsf{p}}}$ *is* D^p_3-*complete.*

7. Inconsistency and Context

Let us recall the motivation for the notion of strong inconsistency. In monotonic logics, we have: If $\mathcal{H} \subseteq \mathcal{K}$ is inconsistent, then the same is true for each $\mathcal{H}'$ with $\mathcal{H} \subseteq \mathcal{H}' \subseteq \mathcal{K}$. In a non-monotonic logic this is not necessarily the case which led to the definition of *strong inconsistency*. However, if we are aiming to repair an inconsistent knowledge base $\mathcal{K}$, instead of insisting on moving to maximal consistent subsets of $\mathcal{K}$, this observation suggests a novel approach, namely resolving conflicts via *adding* information. Especially in frameworks like ASP where the absence of answer sets is oftentimes due to a minimality requirement, this appears to be a quite promising method to restore consistency. This also suggests that inconsistency measurement should take potential supersets of a given knowledge base into account. As we will see additional information does not only contribute to inconsistency of a knowledge base $\mathcal{K}$, but is also worth investigating when it comes to finding repairs. This leads to more varied situations that we also want to address in the context of measuring inconsistency.

In this section we will first establish some theoretical aspects, namely a generalization of the hitting set duality (Brewka et al., 2019) to situations where we also allow adding information to a knowledge base. We will then point out some implications for the field of measuring inconsistency. Since we want to focus on the latter aspect, all the proofs required for our novel duality results are moved to Appendix B.

Let us first repeat the basic notions required for this result. It will become apparent that all of them do have a counterpart in each setting we are going to investigate. Recall Definition 3.1:

For $\mathcal{H}, \mathcal{K} \subseteq \mathsf{WF}$ with $\mathcal{H} \subseteq \mathcal{K}$, $\mathcal{H}$ is called *strongly* $\mathcal{K}$-*inconsistent* if $\mathcal{H} \subseteq \mathcal{H}' \subseteq \mathcal{K}$ implies $\mathcal{H}'$ is inconsistent. Let $SI(\mathcal{K})$ denote the set of all strongly $\mathcal{K}$-inconsistent sets. If $\mathcal{H}$ is minimal with this property, we call $\mathcal{H}$

minimal strongly $\mathcal{K}$-inconsistent. Let $SI_{min}(\mathcal{K})$ denote the set of all minimal strongly $\mathcal{K}$-inconsistent sets.

Moreover, we let $C_{max}(\mathcal{K})$ be the collection of $\subseteq$-maximal consistent subsets of a knowledge base $\mathcal{K}$.

Definition 7.1. Let $\mathcal{M}$ be a set of sets. We call $\mathcal{S}$ a *hitting set* of $\mathcal{M}$ if $\mathcal{S} \cap M \neq \emptyset$ for each $M \in \mathcal{M}$. If $\mathcal{S}$ is minimal with this property, we call $\mathcal{S}$ a *minimal hitting set* of $\mathcal{M}$.

Now let us restate Theorem 3.6 by Brewka et al. (2019).

Theorem 7.2. *Let $\mathcal{K}$ be a knowledge base. Then, $\mathcal{S}$ is a minimal hitting set of $SI_{min}(\mathcal{K})$ if and only if $\mathcal{K} \setminus \mathcal{S} \in C_{max}(\mathcal{K})$.*

A Hitting Set Duality for Addition-Based Repairs

Let us start with repairs based on additional information. In general, it is not quite clear which additional information might be appropriate, especially when considering an arbitrary abstract logic as in Definition 2.13. Moreover, phrasing meaningful results appears hard when investigating an *arbitrary* superset of a knowledge base $\mathcal{K}$. We thus assume the set of potential additional information is given.

More precisely, we consider knowledge bases $\mathcal{K}$ (as usual) and $\mathcal{G}$ (of potential additional assumptions). The set $\mathcal{G}$ itself is not necessarily consistent. For technical convenience we assume $\mathcal{K}$ and $\mathcal{G}$ to be disjoint. This assumption also matches the intuitive meaning of $\mathcal{G}$ as a set of potential additional information. The following definition formally introduces repairs that utilize $\mathcal{G}$.

Definition 7.3. Let $\mathcal{K}$ and $\mathcal{G}$ be disjoint knowledge bases. If for $\mathcal{A} \subseteq \mathcal{G}$, $\mathcal{K} \cup \mathcal{A}$ is consistent, then we call $\mathcal{A}$ a *repairing subset of* $\mathcal{G}$ wrt. $\mathcal{K}$. Let $\textsc{Rep}(\mathcal{K}, \mathcal{G})$ and $\textsc{Rep}_{min}(\mathcal{K}, \mathcal{G})$ denote the set of repairing subsets of $\mathcal{G}$ wrt. $\mathcal{K}$ and the minimal ones, respectively.

Of course, no additional information is capable of repairing an inconsistent knowledge base of a monotonic logic. So our running example of a propositional knowledge base is meaningless in this context, and we move straight to our non-monotonic formalisms.

Example 7.4. Recall our running example P.

$$
\begin{array}{lrr}
P: & a \vee b. & a \leftarrow b. \\
 & c \leftarrow \text{not } b. & \neg c \leftarrow \text{not } b.
\end{array}
$$

Assume we have

$$
\begin{array}{lrr}
G: & a. & b. \\
 & d. & b \leftarrow d.
\end{array}
$$

We see that $P \cup \{b.\}$ and $P \cup \{d., \ b \leftarrow d.\}$ are already consistent and thus

$$\textsc{Rep}_{min}(P, G) = \{\{b.\}, \{d., \ b \leftarrow d.\}\}.$$

Minimality is clear.

Let us now extend our running example for abstract argumentation. Recall Example 2.17 where we defined the logic

$$L_A^{\mathrm{AAF}} = \left(\mathsf{WF}_A^{\mathrm{AAF}}, \mathsf{BS}_A^{\mathrm{AAF}}, \mathsf{INC}_A^{\mathrm{AAF}}, \mathsf{ACC}_A^{\mathrm{AAF}}\right).$$

We pointed out that an AF is actually a tuple and not a set. We thus represent AFs as a knowledge base in a way that a set A of arguments is fixed and $\mathcal{K}$ contains the attacks. Hence augmenting F with another knowledge base G means in our setup additional *attacks* rather than novel arguments. Of course, it would be possible to interpret our running example framework F as an AF over, e. g., $A' = \{a, b, c, d\}$ resulting in an AF containing the argument "d" which does not participate in any attack. We will stick however with an AF over $A = \{a, b, c\}$.

Example 7.5. So consider the AF represented by $R = \{(a, b), (b, c), (c, c)\}$. Assume we are given additional attacks $G = \{(a, c), (b, a), (c, b)\}$. Observe that $R \cup \{(a, c)\}$ and $R \cup \{(b, a)\}$ represent AFs that possess stable extensions. We thus see

$$\textsc{Rep}_{min}(R, G) = \{\{(a, c)\}, \{(b, a)\}\}$$

Again, the repairs are clearly minimal.

Our goal is to characterize the minimal repairs for a given knowledge base $\mathcal{K}$ in terms of a hitting set duality similar in spirit to Theorem 7.2. In the

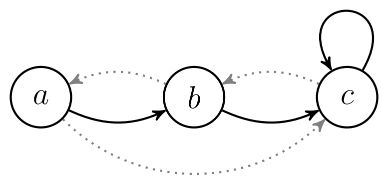


Figure 4: The argumentation framework represented by $R \cup G$ (attacks in G dotted and gray) from Example 7.5

Figure 5: The argumentation frameworks represented by $R \cup \{(a, c)\}$ and $R \cup \{(b, a)\}$ from Example 7.5 including their respective stable extensions (gray)

latter theorem the central notion is strong inconsistency i. e., subsets $\mathcal{H}$ of a knowledge base $\mathcal{K}$ such that each set $\mathcal{H}'$ with $\mathcal{H} \subseteq \mathcal{H}' \subseteq \mathcal{K}$ is inconsistent. For addition-based repairs, our central notion is a natural counterpart to this, which we want to develop by considering Example 7.4 again.

Assume for the moment the goal was already achieved, implying we had certain sets whose minimal hitting sets are

$$\textsc{Rep}_{min}(P, G) = \{\{b.\}, \{d., \, b \leftarrow d.\}\}.$$

Even without being aware of a general technique, this is easy to obtain for this particular example: We consider $G_1 = \{b., \, d.\}$ and $G_2 = \{b., \, b \leftarrow d.\}$. The reader may observe that the two minimal hitting sets of G_1 and G_2 are $\{b.\}$ and $\{d., \, b \leftarrow d.\}$. By removing the G_i from G, we find their meaning: We obtain

$$\begin{array}{rrr} P: & a \lor b. & a \leftarrow b. \\ & c \leftarrow \text{not } b. & \neg c \leftarrow \text{not } b. \\ G \setminus G_1: & a. & b \leftarrow d. \\ G \setminus G_2: & a. & d. \end{array}$$

and we see that P is now strongly inconsistent in both cases. More precisely, P is strongly $(P \cup G \setminus G_1)$-inconsistent as well as strongly $(P \cup G \setminus G_2)$-inconsistent. The intuitive reason is that in both cases we removed all possibilities to repair P via G. Hence, in general we are looking for sets $\mathcal{H} \subseteq \mathcal{G}$ such that $\mathcal{K}$ is strongly $(\mathcal{K} \cup \mathcal{G} \setminus \mathcal{H})$-inconsistent. If we set $\mathcal{A} = \mathcal{G} \setminus \mathcal{H}$, this means $\mathcal{K}$ has to be strongly $(\mathcal{K} \cup \mathcal{A})$-inconsistent.

Definition 7.6. Let $\mathcal{K}$ and $\mathcal{G}$ be disjoint knowledge bases. If for $\mathcal{A} \subseteq \mathcal{G}$, $\mathcal{K}$ is strongly $(\mathcal{K} \cup \mathcal{A})$-inconsistent, i. e., $\mathcal{K} \in SI(\mathcal{K} \cup \mathcal{A})$, then we call $\mathcal{A}$ a *non-repairing subset* of $\mathcal{G}$ wrt. $\mathcal{K}$. Let $\textsc{Nrep}(\mathcal{K}, \mathcal{G})$ and $\textsc{Nrep}_{max}(\mathcal{K}, \mathcal{G})$ denote the set of non-repairing subsets of $\mathcal{G}$ wrt. $\mathcal{K}$ and the maximal ones, respectively.

To see the non-repairing subsets of $\mathcal{G}$ at work, we consider the following examples:

Example 7.7. Recall our running example P from above

$$\begin{array}{lrr} P: & a \vee b. & a \leftarrow b. \\ & c \leftarrow \text{not } b. & \neg c \leftarrow \text{not } b. \end{array}$$

with

$$\begin{array}{lrr} G: & a. & b. \\ & d. & b \leftarrow d. \end{array}$$

Indeed, the maximal non-repairing subsets of G are A_1 and A_2 where

$$\begin{array}{lrr} A_1: & a. & b \leftarrow d. \\ A_2: & a. & d. \end{array}$$

Example 7.8. Now recall the framework represented by $R = \{(a, b), (b, c), (c, c)\}$ with $G = \{(a, c), (b, a), (c, b)\}$. There is only one non-repairing subset of G, namely $\{(c, b)\}$. To see this recall from Figure 5 that adding "(a, c)" or "(b, a)" results in a consistent AF. Hence

$$\textsc{Nrep}_{max}(R, G) = \{\{(c, b)\}\}.$$

We are almost ready to phrase a duality result similar in spirit to Theorem 7.2. As before, we require one additional auxiliary notion, namely *co*-$\textsc{Nrep}_{max}(\mathcal{K}, \mathcal{G})$ which is defined as expected:

Definition 7.9. Let $\mathcal{K}$ and $\mathcal{G}$ be disjoint knowledge bases. The set co-$\textsc{NRep}_{max}(\mathcal{K}, \mathcal{G})$ consists of all $\overline{\mathcal{A}} \subseteq \mathcal{G}$ such that $\mathcal{G} \setminus \overline{\mathcal{A}}$ is in $\textsc{NRep}_{max}(\mathcal{K}, \mathcal{G})$.

As desired, the following theorem gives a characterization of $\textsc{Rep}_{min}(\mathcal{K}, \mathcal{G})$ in terms of a hitting set duality. Analogously to the characterization of $C_{max}(\mathcal{K})$ via minimal hitting sets of $SI_{min}(\mathcal{K})$ (see Theorem 7.2), we may now characterize $\textsc{Rep}_{min}(\mathcal{K}, \mathcal{G})$ via minimal hitting sets of co-$\textsc{NRep}_{max}(\mathcal{K}, \mathcal{G})$. In this sense, co-$\textsc{NRep}_{max}(\mathcal{K}, \mathcal{G})$ plays an analogous role for addition-based repairs as $SI_{min}(\mathcal{K})$ for removing-repairs: A removing-based repair of a knowledge base $\mathcal{K}$ must rule out at least one formula of each set in $SI_{min}(\mathcal{K})$; an adding-based repair of $\mathcal{K}$ (given additional formulas $\mathcal{G}$) must add at least one formula of each set in co-$\textsc{NRep}_{max}(\mathcal{K}, \mathcal{G})$.

Theorem 7.10 (Superset Duality)**.** *Let $\mathcal{K}$ and $\mathcal{G}$ be disjoint knowledge bases. Then $\mathcal{S}$ is a minimal hitting set of co-$\textsc{NRep}_{max}(\mathcal{K}, \mathcal{G})$ if and only if $\mathcal{S} \in \textsc{Rep}_{min}(\mathcal{K}, \mathcal{G})$.*

As the following theorem shows, we can also characterize $\textsc{NRep}_{max}(\mathcal{K}, \mathcal{G})$ in terms of minimal hitting sets of $\textsc{Rep}_{min}(\mathcal{K}, \mathcal{G})$.

Theorem 7.11 (Superset Duality II)**.** *Let $\mathcal{K}$ and $\mathcal{G}$ be disjoint knowledge bases. Then $\mathcal{S}$ is a minimal hitting set of $\textsc{Rep}_{min}(\mathcal{K}, \mathcal{G})$ if and only if $\mathcal{G} \setminus \mathcal{S} \in \textsc{NRep}_{max}(\mathcal{K}, \mathcal{G})$.*

Example 7.12. Consider again P and G:

$$\begin{array}{lrrlrr} P: & a \vee b. & a \leftarrow b. & G: & a. & b. \\ & c \leftarrow \text{not } b. & \neg c \leftarrow \text{not } b. & & d. & b \leftarrow d. \end{array}$$

Let us summarize:

$$\begin{aligned} \textsc{Rep}_{min}(P, G) &= \{\{b.\}, \{d.,\ b \leftarrow d.\}\}, \\ \textsc{NRep}_{max}(P, G) &= \{\{a.,\ b \leftarrow d.\}, \{a., d.\}\}, \\ co\text{-}\textsc{NRep}_{max}(P, G) &= \{\{b.,\ d.\}, \{b.,\ b \leftarrow d.\}\}. \end{aligned}$$

Indeed, $\textsc{Rep}_{min}(P, G)$ consists of the minimal hitting sets of co-$\textsc{NRep}_{max}(P, G)$ (Theorem 7.10) and vice versa (Theorem 7.11).

Example 7.13. Recall $R = \{(a, b), (b, c), (c, c)\}$ with $G = \{(a, c), (b, a), (c, b)\}$. Here, the relevant sets are:

$$\begin{aligned} \textsc{Rep}_{min}(R, G) &= \{\{(a, c)\}, \{(b, a)\}\}, \\ \textsc{NRep}_{max}(R, G) &= \{\{(c, b)\}\}, \\ co\text{-}\textsc{NRep}_{max}(R, G) &= \{\{(a, c), (b, a)\}\}. \end{aligned}$$

The set $\textsc{Rep}_{min}(R, G)$ consists of the two minimal hitting sets of co-$\textsc{NRep}_{max}(R, G)$. Moreover, co-$\textsc{NRep}_{max}(R, G)$ contains the unique minimal hitting set of $\textsc{Rep}_{min}(R, G)$.

A Hitting Set Duality for Arbitrary Repairs

Of course, Theorem 7.10 is only meaningful if $\mathcal{K}$ is not strongly $(\mathcal{K} \cup \mathcal{G})$-inconsistent, i. e., whenever $\mathcal{G} \notin \textsc{NRep}_{max}(\mathcal{K}, \mathcal{G})$. For example, this is naturally violated whenever the underlying logic is monotonic, but also when $\mathcal{G}$ is inappropriate when it comes to providing repair options for $\mathcal{K}$. This is an advantage of Theorem 7.2: Usually, a knowledge base contains consistent subsets and thus the theorem yields non-trivial results. Clearly, the finest solution would be combining the benefits of both Theorem 7.2 and Theorem 7.10. As it turns out, this can be achieved in a smooth and natural way.

So assume we are given knowledge bases $\mathcal{K}$ and $\mathcal{G}$ with $\mathcal{K} \cap \mathcal{G} = \emptyset$ as before. Our goal is to find a consistent knowledge base $\mathcal{H}$ which is as close as possible to $\mathcal{K}$. In Theorem 7.2 the result was a maximal consistent subset of $\mathcal{K}$, i. e., a knowledge base $\mathcal{H}$ of the form $\mathcal{H} = \mathcal{K} \setminus \mathcal{D}$ where $\mathcal{D}$ is minimal such that $\mathcal{H}$ is consistent. In Theorem 7.10 the result was a minimal consistent superset of $\mathcal{K}$, i. e., a knowledge base $\mathcal{H}$ of the form $\mathcal{H} = \mathcal{K} \cup \mathcal{A}$ where $\mathcal{A}$ is minimal such that $\mathcal{H}$ is consistent. Combining both approaches yields the following notion:

Definition 7.14. Let $\mathcal{K}$ and $\mathcal{G}$ be disjoint knowledge bases. We call $(\mathcal{D}, \mathcal{A})$ a *bidirectional repair* for $\mathcal{K}$ with respect to $\mathcal{G}$ if

- $\mathcal{D} \subseteq \mathcal{K}$ and $\mathcal{A} \subseteq \mathcal{G}$,
- $\mathcal{K} \setminus \mathcal{D} \cup \mathcal{A}$ is consistent.

By $\textsc{bi-Rep}(\mathcal{K}, \mathcal{G})$ we denote the set of all bidirectional repairs for $\mathcal{K}$ with respect to $\mathcal{G}$. Let $\textsc{bi-Rep}_{min}(\mathcal{K}, \mathcal{G})$ be the set of all minimal ones, i. e., if $(\mathcal{D}, \mathcal{A}) \in \textsc{bi-Rep}_{min}(\mathcal{K}, \mathcal{G})$, then $(\mathcal{D}', \mathcal{A}') \in \textsc{bi-Rep}(\mathcal{K}, \mathcal{G})$ and $\mathcal{A}' \subseteq \mathcal{A}$ and $\mathcal{D}' \subseteq \mathcal{D}$ implies $(\mathcal{D}', \mathcal{A}') = (\mathcal{D}, \mathcal{A})$.

Example 7.15. Recall our programs P and G.

$$
\begin{array}{lccclcc}
P: & a \vee b. & a \leftarrow b. & & G: & a. & b. \\
 & c \leftarrow \text{not } b. & \neg c \leftarrow \text{not } b. & & & d. & b \leftarrow d.
\end{array}
$$

We see that $\text{BI-REP}_{min}(P, G)$ consists of the tuples

$(\{a \leftarrow b.\}, \emptyset)$, $(\{c \leftarrow \text{not } b.\}, \emptyset)$, $(\{\neg c \leftarrow \text{not } b.\}, \emptyset)$, $(\emptyset, \{b.\})$, $(\emptyset, \{d.,\ b \leftarrow d.\})$.

Here, one of the sets in $(\mathcal{D}, \mathcal{A})$ is always empty for $(\mathcal{D}, \mathcal{A}) \in \text{BI-REP}_{min}(P, G)$. We want to illustrate that this is not necessarily the case in general.

Example 7.16. Consider P' given via

$$
P': \qquad \leftarrow \text{not } b. \qquad \leftarrow \text{not } c.
$$

and G as before. Note that "$\leftarrow$ not c." will cause inconsistency no matter which rules from G are added. We thus find

$$
\text{BI-REP}_{min}(P', G) = \{(P, \emptyset),\ (\{\leftarrow \text{not } c.\}, \{b.\}),\ (\{\leftarrow \text{not } c.\}, \{d.,\ b \leftarrow d.\})\}.
$$

Indeed, for all tuples $(\mathcal{D}, \mathcal{A}) \in \text{BI-REP}_{min}(P', G)$ we have $\leftarrow \text{not } c. \in \mathcal{D}$ which formalizes that this constraint needs to be removed.

Let us now continue with our running AF example.

Example 7.17. For $R = \{(a, b), (b, c), (c, c)\}$ and $G = \{(a, c), (b, a), (c, b)\}$ we recall from previous examples the following options to turn the represented AF F into one which possesses a stable extension (see Figure 6): Remove (c, c) (F_1), remove (a, b) (F_2), or add (a, c) (F_3) or (b, a) (F_4): Hence

$$
\text{BI-REP}_{min}(R, G) = \{(\{(c, c)\}, \emptyset), (\{(a, b)\}, \emptyset), (\emptyset, \{(a, c)\}), (\emptyset, \{(b, a)\})\}.
$$

We want to emphasize that repair options generalize the notion of consistent subsets of a knowledge base.

Proposition 7.18. *Let $\mathcal{K}$ and $\mathcal{G}$ be disjoint knowledge bases. A tuple of the form $(\mathcal{D}, \emptyset)$ is in $\text{BI-REP}_{min}(\mathcal{K}, \mathcal{G})$ if and only if $\mathcal{H} = \mathcal{K} \setminus \mathcal{D} \in C_{max}(\mathcal{K})$.*

The same is true for addition-based repairs, which demonstrates the symmetry of these notions.

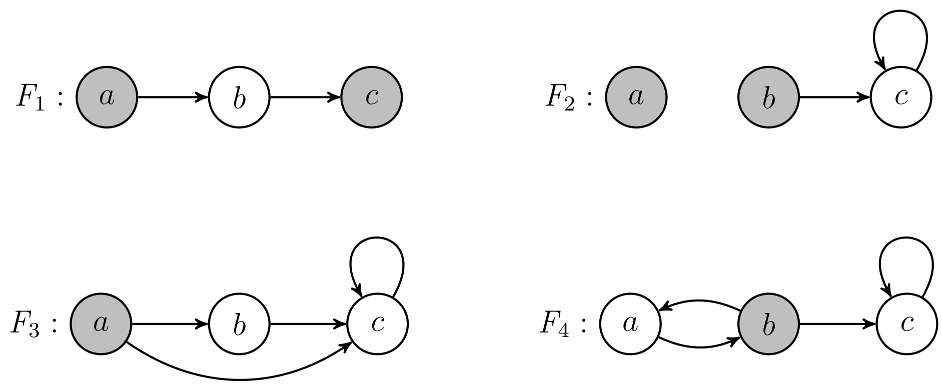


Figure 6: The consistent argumentation frameworks $F_1, \ldots, F_4$ from Example 7.17

Proposition 7.19. *Let $\mathcal{K}$ and $\mathcal{G}$ be disjoint knowledge bases. A tuple of the form $(\emptyset, \mathcal{A})$ is in* $\textsc{bi-Rep}_{min}(\mathcal{K}, \mathcal{G})$ *if and only if* $\mathcal{A} \in \textsc{Rep}_{min}(\mathcal{K}, \mathcal{G})$.

Now let us develop the "inconsistency" notion inducing the desired hitting set characterization of $\textsc{Rep}_{min}(\mathcal{K}, \mathcal{G})$. For addition-based repairs, our first intermediate step (before moving to their complements) was consideration of $\textsc{Nrep}_{max}(\mathcal{K}, \mathcal{G})$, i. e., maximal sets $\mathcal{A} \subseteq \mathcal{G}$ such that $\mathcal{K}$ is strongly $(\mathcal{K} \cup \mathcal{A})$-inconsistent. A strongly $\mathcal{K}$-inconsistent subset $\mathcal{H}$ as used to characterize $C_{max}(\mathcal{K})$ is itself complementary to a set $\mathcal{D} \subseteq \mathcal{K}$ which is maximal such that $\mathcal{K} \setminus \mathcal{D}$ is strongly $\mathcal{K}$-inconsistent. Combining both notions now yields:

Definition 7.20. Let $\mathcal{K}$ and $\mathcal{G}$ be disjoint knowledge bases. We call $(\mathcal{D}, \mathcal{A})$ a *bidirectional non-repair* for $\mathcal{K}$ with respect to $\mathcal{G}$ if

- $\mathcal{D} \subseteq \mathcal{K}$ and $\mathcal{A} \subseteq \mathcal{G}$,
- $\mathcal{K} \setminus \mathcal{D}$ is strongly $(\mathcal{K} \cup \mathcal{A})$-inconsistent, i. e., $\mathcal{K} \setminus \mathcal{D} \in SI(\mathcal{K} \cup \mathcal{A})$.

Denote by $\textsc{bi-Nrep}(\mathcal{K}, \mathcal{G})$ the set of all bidirectional non-repair for $\mathcal{K}$ with respect to $\mathcal{G}$ and by $\textsc{bi-Nrep}_{max}(\mathcal{K}, \mathcal{G})$ the maximal ones.

We want to emphasize that a bidirectional non-repair is not just about $\mathcal{K} \setminus \mathcal{D} \cup \mathcal{A}$ being inconsistent - it requires that *all* sets $\mathcal{H}$ with $\mathcal{K} \setminus \mathcal{D} \subseteq \mathcal{H} \subseteq \mathcal{K} \cup \mathcal{A}$ are inconsistent (see Example 7.23 below).

To see the notion of bidirectional non-repairs at work, let us reconsider the three examples from above.

Example 7.21. Recall our programs P and G.

$$P: \quad a \vee b. \quad a \leftarrow b. \qquad G: \quad a. \quad b.$$
$$c \leftarrow \text{not } b. \quad \neg c \leftarrow \text{not } b. \qquad d. \quad b \leftarrow d.$$

We already noted that the non-repairing subsets of G are $A_1 = \{a., b \leftarrow d.\}$ and $A_2 = \{a., d.\}$. Moreover, removal of "$a \vee b.$" is never beneficial, no matter which subset of P or G is under consideration. Hence

$$\text{BI-NREP}_{max}(P, G) = \{(\{a \vee b.\}, \{a., b \leftarrow d.\}), (\{a \vee b.\}, \{a., d.\})\}.$$

Example 7.22. Consider again P'

$$P': \qquad \leftarrow \text{not } b. \qquad \leftarrow \text{not } c.$$

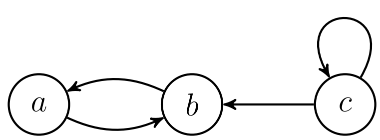
and G as above. Since "$\leftarrow$ not c." will cause inconsistency no matter which rules from G are added we see $P' \setminus \{\leftarrow \text{not } b.\} \in SI(P' \cup G)$, i. e., $(\{\leftarrow \text{not } b.\}, G) \in \text{BI-NREP}_{max}(P', G)$. Moreover, even if "$\leftarrow$ not c." is removed, $\{b.\} \subseteq G$ or $\{d., b \leftarrow d.\} \subseteq G$ is also required in order to repair P'. Thus, $\text{BI-NREP}_{max}(P', G)$ consists of the following tuples:

$$(\{\leftarrow \text{not } b.\}, G), (\{\leftarrow \text{not } c.\}, \{a., b \leftarrow d.\}), (\{\leftarrow \text{not } c.\}, \{a., d.\}).$$

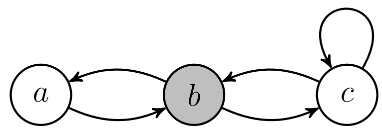
Example 7.23. Recall the AF represented by $R = \{(a, b), (b, c), (c, c)\}$ and the additional attacks $G = \{(a, c), (b, a), (c, b)\}$. Observe that removing "(b, c)" or adding "(c, b)" is in no situation beneficial. Hence,

$$\text{BI-NREP}_{max}(R, G) = \{(\{(b, c)\}, \{(c, b)\})\}.$$

To verify this, recall that in Example 7.17 we saw that any other modification to F yields one possessing a stable extension (also recall Figure 6). We want to mention that $(\{(b, c)\}, \{(b, a), (c, b)\})$ is not a bidirectional non-repair. Suppose the induced AF is called F':



Then F' is inconsistent since it does not possess any stable extension. However, $R \setminus \{(b, c)\}$ is not strongly $R \cup \{(b, a), (c, b)\}$-inconsistent since moving from $R \setminus \{(b, c)\}$ to $R \cup \{b, a\}$ yields a consistent AF:



We make the following observations to emphasize how bi-Nrep$_{max}(\mathcal{K},\mathcal{G})$ generalizes minimal (strong) inconsistency (via moving to the complement of $\mathcal{D}$). First let us consider a monotonic logic. Recall that in this case, $SI_{min}(\mathcal{K}) = I_{min}(\mathcal{K})$. We do not expect $\mathcal{G}$ to play any role here. Indeed, we find:

Proposition 7.24. *Let $\mathcal{K}$ and $\mathcal{G}$ be disjoint knowledge bases of a monotonic logic. If $(\mathcal{D},\mathcal{A}) \in$ bi-Nrep$_{max}(\mathcal{K},\mathcal{G})$, then $\mathcal{A} = \mathcal{G}$. Moreover, $(\mathcal{D},\mathcal{G}) \in$ bi-Nrep$_{max}(\mathcal{K},\mathcal{G})$ if and only if $\mathcal{H} = \mathcal{K} \setminus \mathcal{D} \in SI_{min}(\mathcal{K})$.*

In the previous proposition $\mathcal{G}$ was irrelevant as the underlying logic was assumed to be monotonic. Clearly, a similar result holds whenever there is no set $\mathcal{G}$ at all.

Proposition 7.25. *Let $\mathcal{K}$ be a knowledge base and $\mathcal{G} = \emptyset$. A tuple of the form $(\mathcal{D},\emptyset)$ is in bi-Nrep$_{max}(\mathcal{K},\mathcal{G})$ if and only if $\mathcal{H} = \mathcal{K} \setminus \mathcal{D} \in SI_{min}(\mathcal{K})$.*

A more advanced version of this result without restricting $\mathcal{G}$ is the following. It shows that there is a general connection between bi-Nrep$_{max}(\mathcal{K},\mathcal{G})$ and $SI_{min}(\mathcal{K})$, but it is not as straightforward as the connection between bi-Rep$_{min}(\mathcal{K},\mathcal{G})$ and $C_{max}(\mathcal{K})$ (see Proposition 7.18). The difference is that we are now looking for maximal instead of minimal tuples with a certain property. Assume we are given a set $\mathcal{H} \in SI_{min}(\mathcal{K})$. We can be sure that $(\mathcal{D},\emptyset) \in$ bi-Nrep$(\mathcal{K},\mathcal{G})$ with $\mathcal{D} = \mathcal{K} \setminus \mathcal{H}$, but there are in general several sets $\mathcal{A} \subseteq \mathcal{G}$ such that $(\mathcal{D},\mathcal{A})$ is maximal in bi-Nrep$(\mathcal{K},\mathcal{G})$. On the other hand, given $(\mathcal{D},\mathcal{A}) \in$ bi-Nrep$_{max}(\mathcal{K},\mathcal{G})$ we can guarantee $\mathcal{K} \setminus \mathcal{D} \in SI(\mathcal{K})$; but not minimality:

Proposition 7.26. *Let $\mathcal{K}$ and $\mathcal{G}$ be disjoint knowledge bases.*

- *If $(\mathcal{D},\mathcal{A}) \in$ bi-Nrep$_{max}(\mathcal{K},\mathcal{G})$, then $\mathcal{H} = \mathcal{K} \setminus \mathcal{D} \in SI(\mathcal{K})$. In particular, there is a set $\mathcal{D}'$ with $\mathcal{D} \subseteq \mathcal{D}'$ such that $\mathcal{K} \setminus \mathcal{D}' \in SI_{min}(\mathcal{K})$.*
- *If $\mathcal{H} = \mathcal{K} \setminus \mathcal{D} \in SI_{min}(\mathcal{K})$, then there is a (not necessarily uniquely defined) $\mathcal{A} \subseteq \mathcal{G}$ such that $(\mathcal{D},\mathcal{A}) \in$ bi-Nrep$_{max}(\mathcal{K},\mathcal{G})$.*

Thus, the second item is quite nice, but the first one suggests some caution: Given a tuple $(\mathcal{D}, \mathcal{A}) \in$ BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$, we cannot just "project away" the second component when looking for sets in $SI_{min}(\mathcal{K})$.

Example 7.27. Example 7.21 shows that BI-NREP$_{max}(P, G)$ contains two distinct tuples (D, A) with $D = \{a \vee b.\}$. We already have $H = P \setminus D \in SI_{min}(P)$, so we can choose $D' = D$ for the first item in Proposition 7.26. Since there are two tuples of this form in Example 7.21, it illustrates in particular that A in the second item in Proposition 7.26 is not uniquely defined in general.

Now let us compare BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$ to the non-repairing subsets of $\mathcal{G}$ from Definition 7.6. Considering cases where the underlying logic is monotonic or $\mathcal{G}$ is empty will clearly not yield insightful results when investigating NREP$(\mathcal{K}, \mathcal{G})$. However, we find a counterpart to Proposition 7.26.

Proposition 7.28. *Let $\mathcal{K}$ and $\mathcal{G}$ be disjoint knowledge bases.*

- *If $(\mathcal{D}, \mathcal{A}) \in$ BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$ then $\mathcal{A} \in$ NREP$(\mathcal{K}, \mathcal{G})$. In particular, there is a set $\mathcal{A}'$ with $\mathcal{A} \subseteq \mathcal{A}'$ such that $\mathcal{A}' \in$ NREP$_{max}(\mathcal{K})$.*
- *If $\mathcal{A} \in$ NREP$_{max}(\mathcal{K}, \mathcal{G})$, then there is a (not necessarily uniquely defined) $\mathcal{D} \subseteq \mathcal{K}$ such that $(\mathcal{D}, \mathcal{A}) \in$ BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$.*

We now return to the main goal of this section, namely our duality characterization for BI-REP$_{min}(\mathcal{K}, \mathcal{G})$. It should not be surprising that a notion of *co*-BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$ is required. The following is natural and well-behaving, extending the previous one component-wise.

Definition 7.29. Let $\mathcal{K}$ and $\mathcal{G}$ be disjoint knowledge bases. The set *co*-BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$ consists of all $(\overline{\mathcal{A}}, \overline{\mathcal{D}})$ such that $(\mathcal{G} \setminus \overline{\mathcal{A}}, \mathcal{K} \setminus \overline{\mathcal{D}})$ is in BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$.

Let us summarize the important sets from the previous examples.

Example 7.30. Recall our programs P and G.

$$\begin{array}{lrrlrr} P: & a \vee b. & a \leftarrow b. & G: & a. & b. \\ & c \leftarrow \text{not } b. & \neg c \leftarrow \text{not } b. & & d. & b \leftarrow d. \end{array}$$

We found that BI-REP$_{min}(P, G)$ consists of the following tuples:

$(\{a \leftarrow b.\}, \emptyset)$, $(\{c \leftarrow \text{not } b.\}, \emptyset)$, $(\{\neg c \leftarrow \text{not } b.\}, \emptyset)$, $(\emptyset, \{b.\})$, $(\emptyset, \{d., b \leftarrow d.\})$.

Moreover,

$$\text{BI-NREP}_{max}(P,G) = \{(\{a \vee b.\}, \{a., b \leftarrow d.\}), (\{a \vee b.\}, \{a., d.\})\}.$$

Hence if we set $H = P \setminus \{a \vee b.\}$ we obtain

$$co\text{-BI-NREP}_{max}(P,G) = \{(H, \{b., d.\}), (H, \{b., b \leftarrow d.\})\}.$$

The following theorem states that the desired duality result is indeed obtained. Before stating it, we need to extend the notion of a hitting set to tuples of sets. This is done in the natural way: Given tuples $\mathcal{M} = \{(X_i, Y_i) \mid i = 1, \dots, n\}$, we say (X, Y) is a hitting set of $\mathcal{M}$ if $X \cap X_i \neq \emptyset$ or $Y \cap Y_i \neq \emptyset$ for each $i = 1, \dots, n$. This is natural as it corresponds to extending the intersection to tuples of sets component-wise and letting $(\emptyset, \emptyset)$ be the empty tuple.

Theorem 7.31 (Subset-Superset Duality)**.** *Let $\mathcal{K}$ and $\mathcal{G}$ be disjoint knowledge bases. Then $\mathcal{S}$ is a minimal hitting set of $co\text{-BI-NREP}_{max}(\mathcal{K}, \mathcal{G})$ iff $\mathcal{S} \in \text{BI-REP}_{min}(\mathcal{K}, \mathcal{G})$.*

For this, proving the following dual statement almost suffices (see Appendix Appendix B)

Theorem 7.32 (Subset-Superset Duality II)**.** *Let $\mathcal{K}$ and $\mathcal{G}$ be disjoint knowledge bases. Then $\mathcal{S}$ is a minimal hitting set of $\text{BI-REP}_{min}(\mathcal{K}, \mathcal{G})$ iff $\mathcal{S} \in co\text{-BI-NREP}_{max}(\mathcal{K}, \mathcal{G})$.*

To see the duality at work, we recall our examples.

Example 7.33. Consider again P'

$$P': \qquad \leftarrow \text{not } b. \qquad \leftarrow \text{not } c.$$

with G as usual. We found

$$\text{BI-REP}_{min}(P', G) = \{(P, \emptyset), (\{\leftarrow \text{not } c.\}, \{b.\}), (\{\leftarrow \text{not } c.\}, \{d., b \leftarrow d.\})\}$$

and $co\text{-BI-NREP}_{max}(P', G)$ consists of

$$(\{\leftarrow \text{not } c.\}, \emptyset), (\{\leftarrow \text{not } b.\}, \{b., d.\}), (\{\leftarrow \text{not } b.\}, \{b., b \leftarrow d.\}).$$

Take $(\{\leftarrow \text{not } c.\}, \{b.\}) \in \textsc{bi-Rep}_{min}(P', G)$. Indeed,

$$\begin{aligned}
(\{\leftarrow \text{not } c.\}, \{b.\}) \cap (\{\leftarrow \text{not } c.\}, \emptyset) &= (\{\leftarrow \text{not } c.\}, \emptyset) \\
(\{\leftarrow \text{not } c.\}, \{b.\}) \cap (\{\leftarrow \text{not } b.\}, \{b.,\ d.\}) &= (\emptyset, \{b.\}) \\
(\{\leftarrow \text{not } c.\}, \{b.\}) \cap (\{\leftarrow \text{not } b.\}, \{b., b \leftarrow d.\}) &= (\emptyset, \{b.\})
\end{aligned}$$

where all intersections are non-empty. It is thus a hitting set of $co\text{-}\textsc{bi-NRep}_{max}(P', G)$. Minimality can bee seen straightforwardly.

Example 7.34. For the AF represented by $R = \{(a, b), (b, c), (c, c)\}$ and additional attacks $G = \{(a, c), (b, a), (c, b)\}$ we found:

$$\begin{aligned}
\textsc{bi-Rep}_{min}(R, G) &= \{(\{(c, c)\}, \emptyset), (\{(a, b)\}, \emptyset), (\emptyset, \{(a, c)\}), (\emptyset, \{(b, a)\})\}, \\
\textsc{bi-NRep}_{max}(R, G) &= \{(\{(b, c)\},\ \{(c, b)\})\}, \\
co\text{-}\textsc{bi-NRep}_{max}(R, G) &= \{(\{(a, b), (c, c)\},\ \{(a, c), (b, a)\})\}.
\end{aligned}$$

Now consider $(\{(c, c)\}, \emptyset) \in \textsc{bi-Rep}_{min}(R, G)$. Then

$$(\{(c, c)\}, \emptyset) \cap (\{(a, b), (c, c)\},\ \{(a, c), (b, a)\}) = (\{(c, c)\}, \emptyset),$$

so it is a hitting set of (the singleton) $co\text{-}\textsc{bi-NRep}_{max}(R, G)$. As above, minimality is clear.

Properties of Hitting Sets And Former Dualities

We now demonstrate how to infer Theorem 7.2 as well as Theorem 7.10 from the more general Theorem 7.31. To see this, we need to investigate the structure of (the minimal hitting sets of) $co\text{-}\textsc{bi-NRep}_{max}(\mathcal{K}, \mathcal{G})$. Let us start with Theorem 7.2. Here, the key observation is that—when trying to restore consistency of $\mathcal{K}$—one is not reliant on $\mathcal{G}$ as long as $\mathcal{K}$ possesses consistent subsets. We need to formally find what this means regarding the minimal hitting sets of $SI_{min}(\mathcal{K})$ resp. $co\text{-}\textsc{bi-NRep}_{max}(\mathcal{K}, \mathcal{G})$. We may then translate the duality characterization from Theorem 7.31 into the special case Theorem 7.2. The first and most important step is the following observation.

Proposition 7.35. *Let $\mathcal{K}$ and $\mathcal{G}$ be disjoint knowledge bases. Let $C_{max}(\mathcal{K}) \neq \emptyset$, i. e., $\mathcal{K}$ possesses consistent subsets and let $SI_{min}(\mathcal{K}) \neq \emptyset$, i. e., $\mathcal{K}$ is inconsistent. A set $\mathcal{S}_D$ is a minimal hitting set of $SI_{min}(\mathcal{K})$ if and only if $(\mathcal{S}_\mathcal{D}, \emptyset)$ is a minimal hitting set of $co\text{-}\textsc{bi-NRep}_{max}(\mathcal{K}, \mathcal{G})$.*

We want to emphasize that Proposition 7.35 in particular implies the following: If (the conditions of Proposition 7.35 are met and) $(\overline{\mathcal{D}}, \overline{\mathcal{A}}) \in$ co-BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$, then $\overline{\mathcal{D}} \neq \emptyset$. Otherwise, $(\mathcal{S}_{\mathcal{D}}, \emptyset)$ could not be a (minimal) hitting set of co-BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$:

Proposition 7.36. *Let $\mathcal{K}$ and $\mathcal{G}$ be disjoint knowledge bases. Let $C_{max}(\mathcal{K}) \neq \emptyset$, i. e., $\mathcal{K}$ possesses consistent subsets and let $SI_{min}(\mathcal{K}) \neq \emptyset$, i. e., $\mathcal{K}$ is inconsistent. Then there is no tuple $(\overline{\mathcal{D}}, \overline{\mathcal{A}}) \in co$-BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$ with $\overline{\mathcal{D}} = \emptyset$.*

Having established those properties of (minimal) hitting sets, it is possible to infer Theorem 7.2 from Theorem 7.31 as a corollary.

Proof of Theorem 7.2. In case $\emptyset \in SI(\mathcal{K})$, i. e., any subset of $\mathcal{K}$ is inconsistent, then the claim holds trivially. So assume $\mathcal{K}$ possesses consistent subsets.

Let $\mathcal{S}$ be a minimal hitting set of $SI_{min}(\mathcal{K})$. Consider an arbitrary knowledge base $\mathcal{G}$ with $\mathcal{K} \cap \mathcal{G} = \emptyset$. Due to Proposition 7.35, $\mathcal{S}$ is a hitting set of $SI_{min}(\mathcal{K})$ if and only if $(\mathcal{S}, \emptyset)$ is a minimal hitting set of co-BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$. By Theorem 7.31 this is equivalent to $(\mathcal{S}, \emptyset) \in$ BI-REP$_{min}(\mathcal{K}, \mathcal{G})$. Due to Proposition 7.18, this is the case if and only if $\mathcal{K} \setminus \mathcal{S} \in C_{max}(\mathcal{K})$. □

Let us now see how to analogously derive Theorem 7.10. Note that the previous derivation was based on Proposition 7.35, so towards Theorem 7.10 we require a counterpart to it:

Proposition 7.37. *Let $\mathcal{K}$ and $\mathcal{G}$ be disjoint knowledge bases. Let* REP$_{min}(\mathcal{K}, \mathcal{G}) \neq$ *$\emptyset$, i. e., $\mathcal{K}$ possesses addition-based repairs and let $SI_{min}(\mathcal{K}) \neq \emptyset$, i. e., $\mathcal{K}$ is inconsistent.*

*A set $\mathcal{S}_{\mathcal{A}}$ is a minimal hitting set of co-*NREP$_{max}(\mathcal{K}, \mathcal{G})$ *if and only if $(\emptyset, \mathcal{S}_{\mathcal{A}})$ is a minimal hitting set of co-*BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$.

We may infer an analogous result about the tuples in co-BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$:

Proposition 7.38. *Let $\mathcal{K}$ and $\mathcal{G}$ be disjoint knowledge bases. Let* REP$_{min}(\mathcal{K}, \mathcal{G}) \neq$ *$\emptyset$, i. e., $\mathcal{G}$ possesses repairing subsets wrt. $\mathcal{K}$ and let $SI_{min}(\mathcal{K}) \neq \emptyset$, i. e., $\mathcal{K}$ is inconsistent.. Then there is no $(\overline{\mathcal{D}}, \overline{\mathcal{A}}) \in co$-*BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$ *with $\overline{\mathcal{A}} = \emptyset$.*

Now Theorem 7.10 is a corollary of Theorem 7.31.

Proof of Theorem 7.10. The case REP$_{min}(\mathcal{K}, \mathcal{G}) = \emptyset$ is clear. So let REP$_{min}(\mathcal{K}, \mathcal{G}) \neq \emptyset$. Let $\mathcal{S}$ be a minimal hitting set of co-NREP$_{max}(\mathcal{K}, \mathcal{G})$. Due to Proposition 7.37, $\mathcal{S}$ is a hitting set of co-NREP$_{max}(\mathcal{K})$ if and only if $(\mathcal{S}, \emptyset)$ is a minimal hitting set of co-BI-NREP$_{max}(\mathcal{K},\mathcal{G})$. By Theorem 7.31 this is equivalent to $(\mathcal{S},\emptyset) \in$ BI-REP$_{min}(\mathcal{K},\mathcal{G})$. Due to Proposition 7.19, this is the case if and only if $\mathcal{S} \in$ REP$_{min}(\mathcal{K})$. $\square$

We thus see that both Theorem 7.2 as well as Theorem 7.10 can be inferred from Theorem 7.31 by establishing appropriate connections between the hitting sets of the different inconsistency notions.

Measuring Inconsistent Subsets

Let us now turn to measuring inconsistency. Assume we are given a knowledge base $\mathcal{K}$ as well as a disjoint knowledge base $\mathcal{G}$. In the present section, we do not interpret $\mathcal{G}$ as a set of potential additional assumptions. We will instead assume $\mathcal{K} \cup \mathcal{G}$ is our whole knowledge base and thus, $\mathcal{K}$ is not an isolated one, but seen as a subset of $\mathcal{K} \cup \mathcal{G}$. This should be taken into account when assessing the conflicts within $\mathcal{K}$ as the following examples illustrate.

Example 7.39. Consider the logic program $P = \{\leftarrow \text{not } a, \text{not } c.\}$. Since there is no way to infer a or c, the program P is inconsistent and, e.g., the measure $\mathcal{I}_{\mathsf{MSI}}$ assigns 1 to it. However, interpreted as part of the program $P \cup G$ with

$$\begin{array}{llll} P \cup G: & a \leftarrow \text{not } b. & c \leftarrow \text{not } d. & \leftarrow \text{not } a, \text{not } c. \\ & b \leftarrow \text{not } a. & d \leftarrow \text{not } c. & \end{array}$$

the program P simply constrains the answer sets rather than causing inconsistency. Hence, although $\mathcal{I}_{\mathsf{MSI}}(P) = 1$ seems reasonable on its own, it does not appear to make sense when considering $P \cup G$.

Example 7.40. Let $P = \{\leftarrow \text{not } a., \leftarrow \text{not } b.\}$. Consider the program $P \cup G$ given as follows:

$$P \cup G: \ a \vee b. \qquad \leftarrow \text{not } a. \qquad \leftarrow \text{not } b.$$

Inconsistency of $P \cup G$ stems from the two constraints "$\leftarrow$ not a." and "$\leftarrow$ not b.". As answer sets are required to be minimal models, it is not possible to satisfy both constraints simultaneously. The subset $P = \{\leftarrow \text{not } a., \leftarrow \text{not } b.\}$ obviously consists of two conflicts and this intuition is confirmed by the observation that $\mathcal{I}_{\mathsf{MSI}}(P) = 2$. However, given the disjunctive rule

"$a \vee b.$", this is peculiar since there is actually only one conflict which cannot be resolved (either a or b is missing). This is confirmed by the observation $\mathcal{I}_{\mathsf{MSI}}(P \cup G) = 1$.

A simple solution simulates the concept of strong inconsistency. Recall that a subset of a knowledge base is strongly inconsistent if it contains conflicts that cannot be resolved (within $\mathcal{K}$). We can proceed similarly here and take all supersets of $\mathcal{K}$ within $\mathcal{K} \cup \mathcal{G}$ into account, looking for the smallest possible inconsistency degree. As this approach depends on a given measure $\mathcal{I}$, we obtain the following notion.

Definition 7.41. Let $\mathcal{I} : 2^{\mathsf{WF}} \to \mathbb{R}^{\infty}_{\geq 0}$ be an inconsistency measure and $\mathcal{K} \cup \mathcal{G}$ a knowledge base. We call

$$\mathsf{Co}_{\mathcal{G},\mathcal{I}}(\mathcal{K}) := \min_{\mathcal{G}' \subseteq \mathcal{G}} \mathcal{I}(\mathcal{K} \cup \mathcal{G}') \tag{9}$$

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

This approach is quite well-behaving for the two examples we considered before.

Example 7.42. For the logic program $P = \{\leftarrow \text{not } a, \text{not } c.\}$ with

$$\begin{array}{llll} P \cup G: & a \leftarrow \text{not } b. & c \leftarrow \text{not } d. & \leftarrow \text{not } a, \text{not } c. \\ & b \leftarrow \text{not } a. & d \leftarrow \text{not } c. & \end{array}$$

we immediately see $\mathsf{Co}_{G,\mathcal{I}_{\mathsf{MSI}}}(P) = 0$ caused by consistency of $P \cup G$.

Example 7.43. Consider again $P = \{\leftarrow \text{not } a., \leftarrow \text{not } b.\}$ where $P \cup G$ is

$$P \cup G: \quad a \vee b. \qquad \leftarrow \text{not } a. \qquad \leftarrow \text{not } b.$$

Clearly, there is only one strongly inconsistent subset of $P \cup G$, namely P. We thus see $\mathsf{Co}_{G,\mathcal{I}_{\mathsf{MSI}}}(P) = 1$.

Let us collect some properties of $\mathsf{Co}_{\mathcal{G},\mathcal{I}}(\mathcal{K})$, depending on the given measure $\mathcal{I}$. We see that some desirable properties of $\mathcal{I}$ transfer to $\mathsf{Co}_{\mathcal{G},\mathcal{I}}(\mathcal{K})$.

Proposition 7.44. *Let $\mathcal{K}, \mathcal{K}'$ and $\mathcal{G}$ be a knowledge bases.*

(a) If $\mathcal{I}$ satisfies consistency*, then $\mathsf{Co}_{\mathcal{G},\mathcal{I}}(\mathcal{K}) = 0$ if and only if $\mathcal{K} \notin SI(\mathcal{K} \cup \mathcal{G})$,*

(b) if $\mathcal{I}$ satisfies independence *and* $\alpha \in \mathit{Ntr}(\mathcal{K} \cup \mathcal{G})$*, then* $\mathsf{Co}_{\mathcal{G},\mathcal{I}}(\mathcal{K}) = \mathsf{Co}_{\mathcal{G},\mathcal{I}}(\mathcal{K} \setminus \{\alpha\})$*,*

(c) if $\mathcal{I}$ satisfies strong equivalence *and* $\mathcal{K} \equiv_s \mathcal{K}'$*, then* $\mathsf{Co}_{\mathcal{G},\mathcal{I}}(\mathcal{K}) = \mathsf{Co}_{\mathcal{G},\mathcal{I}}(\mathcal{K}')$*.*

Proof. Let $\mathcal{K}, \mathcal{K}'$ and $\mathcal{G}$ be knowledge bases.
(a): Assume the measure $\mathcal{I}$ satisfies the *consistency* postulate. If $\mathsf{Co}_{\mathcal{G},\mathcal{I}}(\mathcal{K}) = 0$ then there is $\mathcal{K} \cup \mathcal{G}' \subseteq \mathcal{K} \cup \mathcal{G}$ such that $\mathcal{I}(\mathcal{K} \cup \mathcal{G}') = 0$. Since $\mathcal{I}$ satisfies *consistency*, $\mathcal{K} \cup \mathcal{G}'$ is consistent and therefore $\mathcal{K}$ is not strongly $(\mathcal{K} \cup \mathcal{G})$-inconsistent. On the other hand, if $\mathcal{K} \notin \mathit{SI}(\mathcal{K} \cup \mathcal{G})$ then there is $\mathcal{G}' \subseteq \mathcal{G}$ such that $\mathcal{K} \cup \mathcal{G}'$ is consistent and $\mathcal{I}(\mathcal{K} \cup \mathcal{G}') = 0$. Then $\mathsf{Co}_{\mathcal{G},\mathcal{I}}(\mathcal{K}) = 0$ as well.
(b): Let $\mathcal{I}$ satisfy *independence*. For $\alpha \in \mathit{Ntr}(\mathcal{K} \cup \mathcal{G})$ we have $\alpha \in \mathit{Ntr}(\mathcal{H})$ for any subset $\mathcal{H}$ of $\mathcal{K} \cup \mathcal{G}$. Now the claim follows immediately.
(c): Let $\mathcal{I}$ satisfy *strong equivalence* and let $\mathcal{K} \equiv_s \mathcal{K}'$. It follows that $\mathcal{I}(\mathcal{K}) = \mathcal{I}(\mathcal{K}')$ and, as $\mathcal{K} \equiv_s \mathcal{K}'$ implies $\mathcal{K} \cup \mathcal{G}' \equiv_s \mathcal{K}' \cup \mathcal{G}'$ for all $\mathcal{G}' \subseteq \mathcal{G}$, $\mathcal{I}(\mathcal{K} \cup \mathcal{G}') = \mathcal{I}(\mathcal{K}' \cup \mathcal{G}')$ and by this the claim. $\square$

While the expression (9) simply takes the minimum over all possible supersets, it would be rather appealing to utilize a general framework for measuring the information which is added to a knowledge base. The idea of measuring information is not novel (Shannon, 1948; Lozinskii, 1994). An *information measure* (Grant and Hunter, 2011) is defined as a mapping $\mathcal{J}$ assigning non-negative real numbers to propositional knowledge bases and satisfying

- if $\mathcal{K} = \emptyset$, then $\mathcal{J}(\mathcal{K}) = 0$,
- if $\mathcal{K} \subseteq \mathcal{K}'$ and $\mathcal{K}'$ is consistent, then $\mathcal{J}(\mathcal{K}) \leq \mathcal{J}(\mathcal{K}')$,
- if $\mathcal{K}$ is consistent and at least one formula $\alpha \in \mathcal{K}$ is not a tautology, then $\mathcal{J}(\mathcal{K}) > 0$.

Equipped with an appropriate technique to measure information in non-monotonic logics, one could utilize $\mathcal{J}(\mathcal{A})$ (where $\mathcal{A} \subseteq \mathcal{G}$) to measure the information added to $\mathcal{K}$ and then consider $\mathcal{I}(\mathcal{K} \cup \mathcal{A})$ to measure inconsistency of the remaining conflicts within $\mathcal{K}$. This approach induces inconsistency measures based on, e.g., the expression

$$\min_{\mathcal{A} \subseteq \mathcal{G}} \mathcal{J}(\mathcal{A}) + \mathcal{I}(\mathcal{K} \cup \mathcal{A}). \tag{10}$$

Although we leave a thorough investigation of this issue for future work, we want to mention that (9) is a special case of this approach, utilizing the trivial function $\mathcal{J} \equiv 0$.[4] In the subsequent Section 8.1 on measuring inconsistency in ASP we will consider a measure $\mathcal{I}_{\pm}$ which can be interpreted as an expression of this form, where $\mathcal{J}$ counts the number of facts we add to a given program.

In order to calculate (9), we require a given inconsistency measure $\mathcal{I}$. We also want to consider a novel approach here. Recall the measures introduced in Section 3. They are based on (the number of) strongly inconsistent subsets of a knowledge base $\mathcal{K}$. The notion of strong inconsistency facilitated the previously considered generalization of inconsistency measures from the literature. The results of the previous sections suggest that bidirectional non-repairs are worth investigating when given a knowledge base $\mathcal{K}$ as a subset of $\mathcal{K} \cup \mathcal{G}$. This motivates considering the following measures of the type

$$\mathcal{I}_{\mathcal{G}} : 2^{\mathsf{WF}} \to \mathbb{R}^{\infty}_{\geq 0} \text{ with } \mathcal{K} \mapsto \mathcal{I}_{\mathcal{G}}(\mathcal{K}).$$

Definition 7.45. Given disjoint knowledge bases $\mathcal{K}$ and $\mathcal{G}$, define

- $\mathcal{I}_{\mathrm{NR},\mathcal{G}}$ via

$$\mathcal{I}_{\mathrm{NR},\mathcal{G}}(\mathcal{K}) = |\textit{co}\text{-}\textsc{bi-Nrep}_{max}(\mathcal{K},\mathcal{G})| = |\textsc{bi-Nrep}_{max}(\mathcal{K},\mathcal{G})|,$$

- $\mathcal{I}_{\mathrm{NR}^c,\mathcal{G}}$ via

$$\mathcal{I}_{\mathrm{NR}^c,\mathcal{G}}(\mathcal{K}) = \sum_{\substack{(\mathcal{D},\mathcal{A}) \in \\ \textit{co}\text{-}\textsc{bi-Nrep}_{max}(\mathcal{K},\mathcal{G})}} \frac{1}{|\mathcal{D} \cup \mathcal{A}|}.$$

We observe the similarities to the measures $\mathcal{I}_{\mathsf{MSI}}$ and $\mathcal{I}_{\mathsf{MSI^C}}$. Those given in Definition 7.45 are similar in their spirit, replacing $SI_{min}(\mathcal{K})$ with $\textit{co}\text{-}\textsc{bi-Nrep}_{max}(\mathcal{K},\mathcal{G})$. We want to emphasize that the resulting measures are rather pessimistic when it comes to assessing $\mathcal{K}$ as subset of $\mathcal{K} \cup \mathcal{G}$. To illustrate this, let us consider $\mathcal{I}_{\mathrm{NR},\mathcal{G}}$ applied to the previous examples.

[4]More precisely, $\mathcal{J} \equiv 0$ is no information measure since the third condition is violated (Grant and Hunter, 2011)

Example 7.46. Recall the logic program $P = \{\leftarrow \text{not } a, \text{not } c.\}$ with

$$P \cup G: \; a \leftarrow \text{not } b. \qquad c \leftarrow \text{not } d. \qquad \leftarrow \text{not } a, \text{not } c.$$
$$b \leftarrow \text{not } a. \qquad d \leftarrow \text{not } c.$$

Even though $P \cup G$ itself is consistent, there is a non-repairing subset G' of G, namely

$$G': \; b \leftarrow \text{not } a. \qquad d \leftarrow \text{not } c.$$

We note that

$$P \cup G': \; b \leftarrow \text{not } a. \qquad d \leftarrow \text{not } c. \qquad \leftarrow \text{not } a, \text{not } c.$$

is inconsistent. We thus see $\textsc{bi-Nrep}_{max}(P, G) = \{(\emptyset, G')\}$, yielding $\mathcal{I}_{\mathrm{NR},G}(P) = 1$. Moreover, $co\text{-}\textsc{bi-Nrep}_{max}(P, G) = \{(P, G \setminus G')\} = \{(P, \{a \leftarrow \text{not } b., \; c \leftarrow \text{not } d.\})\}$. Hence, we have $\mathcal{I}_{\mathrm{NR}^c,G}(P) = 1/3$.

So these measures punish knowledge bases $\mathcal{K}$ and $\mathcal{G}$ for each maximal bidirectional non-repair. This can be seen as a counterpart to $\mathsf{Co}_{\mathcal{G},\mathcal{I}}(\mathcal{K})$ which rewards $\mathcal{K}$ for any possibility $\mathcal{G}$ possesses to resolve a conflict. In this sense, one also could interpret $\mathcal{I}_{\mathrm{NR},\mathcal{G}}$ and $\mathcal{I}_{\mathrm{NR}^c,\mathcal{G}}$ as measures for the quality of the repair options provided by $\mathcal{G}$. To see the two approaches at work, let us consider the following example:

Example 7.47. Let P and G be the following programs:

$$P: \quad a. \qquad \leftarrow \text{not } b. \qquad G: \quad \neg a. \qquad b.$$

We see that $\mathsf{Co}_{G,\mathcal{I}_{\mathsf{MSI}}}(P) = 0$ since P has the consistent superset $P \cup \{b\}$. However, there is also one bidirectional non-repair, namely $(\{a.\}, \{\neg a.\})$, thus we find $\mathcal{I}_{\mathrm{NR},G}(P) = 1$.

We also want to mention a quite special feature of the measure $\mathcal{I}_{\mathrm{NR}^c,\mathcal{G}}$. Recall the motivation for defining $\mathcal{I}_{\mathsf{MSI^C}}$ in contrast to $\mathcal{I}_{\mathsf{MSI}}$, namely taking the size of a set $\mathcal{H} \in I_{min}(\mathcal{K})$ into account. The measure $\mathcal{I}_{\mathrm{NR}^c,\mathcal{G}}$ attains larger values the bigger the sets in $\textsc{bi-Nrep}_{max}(\mathcal{K}, \mathcal{G})$ are. This is similar in spirit to $\mathcal{I}_{\mathsf{MSI}}$, but not quite the same since punishing the size of a tuple $(\mathcal{D}, \mathcal{A}) \in \textsc{bi-Nrep}_{max}(\mathcal{K}, \mathcal{G})$ means punishing the fact that there is a large number of formulas which are not capable of providing a repair.

Example 7.48. Recall $P = \{\leftarrow \text{not } a., \leftarrow \text{not } b.\}$ and $G = \{a \vee b.\}$, i. e.,

$$P \cup G: \ a \vee b. \qquad\qquad \leftarrow \text{not } a. \qquad\qquad \leftarrow \text{not } b.$$

As $P \cup G$ is inconsistent, we see $\text{BI-NREP}_{max}(P, G) = \{(\emptyset, G)\}$, i. e., $\mathcal{I}_{\mathrm{NR},G}(P) = 1$.

Let us now compare the measures from Definition 7.45 to those from Section 3. We note that $\mathcal{I}_{\mathrm{NR},\mathcal{G}}(\mathcal{K}))$ attains larger values in general. This is a corollary of Proposition 7.26.

Corollary 7.49. *Let $\mathcal{K}$ and $\mathcal{G}$ be disjoint knowledge bases. Then, $\mathcal{I}_{\mathsf{MSI}}(\mathcal{K}) \leq \mathcal{I}_{\mathrm{NR},\mathcal{G}}(\mathcal{K})$.*

Proof. We have $\mathcal{I}_{\mathrm{NR},\mathcal{G}}(\mathcal{K}) = |\text{BI-NREP}_{max}(\mathcal{K}, \mathcal{G})|$. As we know from Proposition 7.26, if $\mathcal{H} = \mathcal{K} \setminus \mathcal{D} \in SI_{min}(\mathcal{K})$, then there is an $\mathcal{A} \subseteq \mathcal{G}$ with $(\mathcal{D}, \mathcal{A}) \in \text{BI-NREP}_{max}(\mathcal{K}, \mathcal{G})$. In particular, $|SI_{min}(\mathcal{K})| \leq |\text{BI-NREP}_{max}(\mathcal{K}, \mathcal{G})|$, i. e., $\mathcal{I}_{\mathsf{MSI}}(\mathcal{K}) \leq \mathcal{I}_{\mathrm{NR},\mathcal{G}}(\mathcal{K})$. □

We observe that $\mathcal{I}_{\mathsf{MSI^c}}(\mathcal{K}) \leq \mathcal{I}_{\mathrm{NR}^c,\mathcal{G}}(\mathcal{K})$ does not hold in general as one can already see from Example 7.46. Here we had $\mathcal{I}_{\mathrm{NR}^c,G}(P) = 1/3$ where $\mathcal{I}_{\mathsf{MSI^c}}(P) = 1$. We want to emphasize that a comparison between $\mathcal{I}_{\mathsf{MSI^c}}(\mathcal{K})$ and $\mathcal{I}_{\mathrm{NR}^c,\mathcal{G}}(\mathcal{K})$ does not appear to be very meaningful. This is because both measures depend positively on the number of undesired sets, but negatively on their size.

However we note that the expected outcome is obtained whenever $\mathcal{G}$ is empty. This is a corollary of Proposition 7.25.

Corollary 7.50. *Let $\mathcal{K}$ be a knowledge base. Then, $\mathcal{I}_{\mathrm{NR},\emptyset}(\mathcal{K}) = \mathcal{I}_{\mathsf{MSI}}(\mathcal{K})$ and $\mathcal{I}_{\mathrm{NR}^c,\emptyset}(\mathcal{K}) = \mathcal{I}_{\mathsf{MSI^c}}(\mathcal{K})$.*

Proof. Due to Proposition 7.25, $(\mathcal{D}, \emptyset) \in \text{BI-NREP}_{max}(\mathcal{K}, \emptyset)$ iff $\mathcal{K} \setminus \mathcal{D} \in SI_{min}(\mathcal{K})$. Equivalently, $(\mathcal{H}, \emptyset) \in co\text{-BI-NREP}_{max}(\mathcal{K}, \emptyset)$ iff $\mathcal{H} \in SI_{min}(\mathcal{K})$. This proves both equations. □

We make an analogous observation whenever the underlying logic is monotonic.

Corollary 7.51. *Let $\mathcal{K}$ and $\mathcal{G}$ be disjoint knowledge bases. Let the underlying logic be monotonic. Then, $\mathcal{I}_{\mathrm{NR},\mathcal{G}}(\mathcal{K}) = \mathcal{I}_{\mathsf{MSI}}(\mathcal{K})$ and $\mathcal{I}_{\mathrm{NR}^c,\mathcal{G}}(\mathcal{K}) = \mathcal{I}_{\mathsf{MSI^c}}(\mathcal{K})$.*

Proof. Due to Proposition 7.24, $(\mathcal{D}, \mathcal{G}) \in$ BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$ iff $\mathcal{K} \setminus \mathcal{D} \in I_{min}(\mathcal{K})$. Equivalently, $(\mathcal{H}, \emptyset) \in$ *co*-BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$ iff $\mathcal{H} \in SI_{min}(\mathcal{K})$. This proves both equations. □

Let us now collect some properties of the two measures we just introduced. As before, they are adapted rationality postulates from Section 4. Note the differences and similarities to Proposition 7.44.

Proposition 7.52. *Given two disjoint knowledge bases $\mathcal{K}$ and $\mathcal{G}$, the measures $\mathcal{I}_{\mathrm{NR},\mathcal{G}}(\cdot)$ and $\mathcal{I}_{\mathrm{NR}^c,\mathcal{G}}(\cdot)$ satisfy*

(a) $\mathcal{I}_{\mathcal{G}}(\mathcal{K}) = 0$ *if and only if* $\mathcal{K}$ *is consistent,*

(b) if $\alpha \in Ntr(\mathcal{K} \cup \mathcal{G})$*, then* $\mathcal{I}_{\mathcal{G}}(\mathcal{K}) = \mathcal{I}_{\mathcal{G}}(\mathcal{K} \setminus \{\alpha\}) = \mathcal{I}_{\mathcal{G} \setminus \{\alpha\}}(\mathcal{K})$*,*

(c) if $\mathcal{G} \equiv_\alpha \mathcal{G}'$*, then* $\mathcal{I}_{\mathcal{G}}(\mathcal{K}) = \mathcal{I}_{\mathcal{G}'}(\mathcal{K})$*.*

Proof. (a): Observe that $\mathcal{K} \setminus \mathcal{D}$ can never be strongly $(\mathcal{K} \cup \mathcal{A})$-inconsistent, whenever $\mathcal{K}$ itself is consistent. So BI-NREP$_{max}(\mathcal{K}, \mathcal{G}) = \emptyset$, i. e., $\mathcal{I}_{\mathrm{NR},\mathcal{G}}(\mathcal{K}) = \mathcal{I}_{\mathrm{NR}^c,\mathcal{G}}(\mathcal{K}) = 0$. If $\mathcal{K}$ is inconsistent, then BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$ contains at least one tuple since BI-NREP$(\mathcal{K}, \mathcal{G})$ contains at least $(\emptyset, \emptyset)$. So $\mathcal{I}_{\mathrm{NR},\mathcal{G}}(\mathcal{K}), \mathcal{I}_{\mathrm{NR}^c,\mathcal{G}}(\mathcal{K}) > 0$.
(b): Clear, similar to the proof of Proposition 5.1.
(c): Clear, similar to the proof of Proposition 5.1. □

This finishes our discussion on the measures based on bidirectional non-repairs. In contrast to the measure (9) defined in Definition 7.41, the measures from Definition 7.45 are based on bidirectional non-repairs and thus do not require a given measure $\mathcal{I}$. One drawback of the latter is that inconsistencies which stem from defaults ("$a \leftarrow$ not a.") are not distinguished from hard-coded ones ("a." vs. "$\neg a$."). This can be achieved by measures defined via (10), when making appropriate choices for $\mathcal{I}$ and $\mathcal{J}$.

8. Related Work

Inconsistency measurement in non-classical frameworks has been addressed in some limited fashion before (Thimm, 2013; Potyka, 2014; De Bona and Finger, 2015; Condotta et al., 2016; Ulbricht et al., 2016; Hunter, 2017; Amgoud and Ben-Naim, 2017). The latter two papers study 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. However, the closest work to this one is the work of Ulbricht et al. (2016) where the special case of L^{ASP^*} is considered rather than a general, non-monotonic logic. It is quite obvious that considering one particular framework only —while being not as general as the present paper— allows to phrase more meaningful postulates as they can be tailored for the given framework. We want to discuss this work in some detail here.

Moreover, Hunter and Konieczny (2010) consider the following problem: Given an inconsistent knowledge base $\mathcal{K}$, how much of the "blame" should be assigned to a given formula $\alpha \in \mathcal{K}$? For this setting, minimal inconsistent subsets of a knowledge base are a rather useful tool. We want to indicate how our notion of strong inconsistency might help to lift some results, but also limitations we face.

8.1. Measuring Inconsistency in ASP

Many of the postulates of Ulbricht et al. (2016) are concerned about situations where some kind of *monotonicity* should hold. However, all of them do have in common that conflicts are preserved as in *strong monotonicity* 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 8.1. Let P be a logic program, i. e., a set of rules of the form

$$l_0 \vee ... \vee l_k \leftarrow l_{k+1}, \ldots, l_m, \text{not } l_{m+1}, \ldots, \text{not } l_n.$$

A set U of literals is called a *splitting set* for P, if $\{l_0, \ldots, l_k\} \cap U \neq \emptyset$ implies $\{l_0, \ldots, l_n\} \subseteq U$ for any rule $r \in P$. For a splitting set U, let $bot_U(P)$ be the set of all rules $r \in P$ with $\{l_0, \ldots, l_n\} \subseteq U$.

Now we consider the following postulates by Ulbricht et al. (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 $\mathcal{I}(bot_U(P)) \leq \mathcal{I}(P)$.

Con-monotonicity If $P \subseteq \mathsf{WF}^{\mathrm{ASP}^*}$ and $r \in P$ is a constraint, then $\mathcal{I}(P) \leq \mathcal{I}(P \cup \{r\})$.

All of them describe situations where the additional rules preserve conflicts of the program on the left hand side. Thus, we see:

Proposition 8.2. *If a measure $\mathcal{I}$ satisfies* strong monotonicity*, it satisfies* CLP-monotonicity*,* Split-monotonicity *and* Con-monotonicity *as well.*

This, however, does not mean that the former postulates are pointless. We consider the measure $\mathcal{I}_\pm$ by Ulbricht et al. (2016).

Definition 8.3. Define $\mathcal{I}_\pm : \mathcal{P} \to \mathbb{R}^\infty_{\geq 0}$ via

$$\mathcal{I}_\pm(P) = \min\{|A| + |D| \mid A, D \subseteq \mathsf{WF}^{\mathrm{ASP}^*} \text{ such that } (P \cup A) \setminus D \text{ is consistent}\}$$

for all $P \in \mathcal{P}$.

Despite satisfying *CLP-monotonicity*, *Split-monotonicity* and *Con-monotonicity*, this measure does not satisfy *strong monotonicity*.

Example 8.4. Consider the following programs P and P':

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

It is easy to see that $SI_{min}(P)$ consists of the three unsatisfied constraints. Moreover, as x cannot be entailed, P' preserves conflicts of P. Yet, $\mathcal{I}_\pm(P) = 3$, while $\mathcal{I}_\pm(P \cup P') = 1$.

This example shall illustrate that the notion of preserving conflicts might be too strong in some cases. The fact that $\mathcal{I}_\pm$ satisfies the three weak versions of *monotonicity* mentioned above confirms the intuition that $\mathcal{I}_\pm$ behaves quite "monotonic" as long as the additional information does not resolve conflicts. This is not surprising as $\mathcal{I}_\pm$ counts the number of modifications that are required on the level of formulas to restore consistency. Moving from P to $P \cup P'$ weakens the severity of the inconsistency, since a single additional rule suffices to satisfy all constraints. However, these are considerations on the level of the language of the given programs and thus hard to capture within the general notion of a logic.

8.2. Culpability Measures

While inconsistency measures aim at assessing the inconsistency of a whole knowledge base, several works have addressed the issue of leveraging such approaches to assign degrees of "blame" for the individual formulas of a knowledge base. More concretely, a *culpability measure* (Daniel, 2009) is a measure $\mathcal{I}$ that takes a knowledge base $\mathcal{K}$ and a formula $\alpha \in \mathcal{K}$ and returns a non-negative number $\mathcal{I}(\mathcal{K}, \alpha)$ representing a degree of responsibility of α on the overall inconsistency of $\mathcal{K}$.[5] As with inconsistency measures, culpability measures can be defined using minimal inconsistent subsets of a propositional knowledge base, leading to a smooth generalization within our setting. Due to non-monotonicity, however, a formula α might resolve conflicts of a knowledge base $\mathcal{K}$. It thus becomes apparent that assessing the blame of one particular formula possess new challenges in a non-monotonic framework. We will briefly discuss this issue here.

Definition 8.5. For a knowledge base $\mathcal{K}$, a culpability measure is a mapping $\mathcal{I}(\mathcal{K}, \cdot) : \mathcal{K} \to \mathbb{R}$ with $\alpha \mapsto \mathcal{I}(\mathcal{K}, \alpha)$.

We consider two approaches by Hunter and Konieczny (2008). The first one is similar in spirit of $\mathcal{I}_{\mathsf{MSI^C}}$. Straightforwardly making use of minimal strongly $\mathcal{K}$-inconsistent subsets yields:

Definition 8.6. Define $MSIV_C(\mathcal{K}, \alpha)$ via

$$MSIV_C(\mathcal{K}, \alpha) = \sum_{\substack{H \in SI_{min}(\mathcal{K}), \\ \alpha \in H}} \frac{1}{|H|}. \tag{11}$$

The second utilizes a given inconsistency measure $\mathcal{I}$ by applying the Shapley value. The latter was proposed as a solution concept for cooperative game theory. Here, players form coalitions with a certain payoff assigned to each coalition. The question is how to assess each player in terms of her "value" for the present coalition (Shapley, 1953). Hunter and Konieczny realized

[5]In some works these functions have been called *inconsistency values* (Hunter and Konieczny, 2008); we stick to the term *culpability measure* as the term *inconsistency value* is also used to refer to the inconsistency degree of a knowledge base wrt. some inconsistency measure

that the Shapley value provides a smooth tool to asses the "blame" of a formula $\alpha \in \mathcal{K}$ for the inconsistency degree of $\mathcal{K}$. Instead of the payoff function assigning coalitions to their payoff simply consider an inconsistency measure:

Definition 8.7. Let $\mathcal{I}$ be an inconsistency measure. The corresponding Shapley culpability measure is given as

$$S^{\mathcal{I}}(\mathcal{K}, \alpha) = \sum_{C \subseteq \mathcal{K}} \frac{(|C|-1)!(|\mathcal{K}|-|C|)!}{|\mathcal{K}|!} (\mathcal{I}(C) - \mathcal{I}(C \setminus \{\alpha\}))$$

Hunter and Konieczny obtain $MSIV_C(\mathcal{K}, \alpha) = S^{\mathcal{I}_{\mathsf{MSI}}}(\mathcal{K}, \alpha)$ for propositional knowledge bases (that is, $\mathcal{I}_{\mathsf{MSI}}$ coincides with the Shapley culpability measure using the established measure $\mathcal{I}_{\mathsf{MI}}$). Since they do not make explicit use of properties of propositional logic (see Proposition 8 by Hunter and Konieczny (2010)) we find:

Proposition 8.8. *If $\mathcal{K}$ is a monotonic knowledge base, then $MSIV_C(\mathcal{K}, \alpha) = S^{\mathcal{I}_{\mathsf{MSI}}}(\mathcal{K}, \alpha)$.*

It is quite easy to see that this equation does not hold in non-monotonic logics. The following example shall illustrate why this is actually good news for the Shapley inconsistency value $S^{\mathcal{I}_{\mathsf{MSI}}}(\mathcal{K}, \alpha)$. As one realizes considering (11), $MSIV_C(\mathcal{K}, \alpha)$ is only capable of "blaming" formulas for occurring in a strongly inconsistent subset. No notion of "reward" for restoring consistency in some cases is taken into consideration.

Example 8.9. Consider the program P given as follows.

$$P: \; a. \quad b. \qquad\qquad \leftarrow \text{not } a, \text{not } c_1. \quad \leftarrow \text{not } a, \text{not } c_2.$$

As P is consistent, $MSIV_C(P, r) = 0$ for all $r \in P$ despite "$a.$" should be rewarded for restoring consistency while the rules "$\leftarrow$ not a, not c_i." should be punished for being dependent on "$a.$".

Now consider the Shapley culpability measure. We have $S^{\mathcal{I}_{\mathsf{MSI}}}(P, b.) = 0$ because "$b.$" is neutral in P. Moreover, $S^{\mathcal{I}_{\mathsf{MSI}}}(P, \leftarrow \text{not } a, \text{not } c_1.) = \frac{1}{2} = S^{\mathcal{I}_{\mathsf{MSI}}}(P, \leftarrow \text{not } a, \text{not } c_2.)$ as they are punished for introducing inconsistency in some cases and $S^{\mathcal{I}_{\mathsf{MSI}}}(P, a.) = -1$ holds because "$a.$" resolves conflicts within P.

The Shapley culpability measure possesses some desirable properties which where already found by Hunter and Konieczny (2008). The proofs work in the same way.

Proposition 8.10. *If $\mathcal{I}$ is an inconsistency measure satisfying* consistency *and* independence*, then $S^{\mathcal{I}}(\mathcal{K}, \cdot)$ satisfies*

Distribution $\sum_{\alpha \in \mathcal{K}} S^{\mathcal{I}}(\mathcal{K}, \alpha) = \mathcal{I}(\mathcal{K})$,

Symmetry *if $\mathcal{I}(\mathcal{K}' \cup \{\alpha\}) = \mathcal{I}(\mathcal{K}' \cup \{\beta\})$ for all $\mathcal{K}' \subseteq \mathcal{K}$ with $\alpha, \beta \notin \mathcal{K}'$, then $S^{\mathcal{I}}(\mathcal{K}, \alpha) = S^{\mathcal{I}}(\mathcal{K}, \beta)$ and*

Neutral independence *if $\alpha \in Ntr(\mathcal{K})$, then $S^{\mathcal{I}}(\mathcal{K}, \alpha) = 0$.*

We believe that consideration of culpability measures for non-monotonic logics is an interesting and quite promising research direction. Besides the culpability measures mentioned in this section, Mu (2015; 2018) also introduced further such measures, in particular one that can be interpreted through Pearl and Halpern's (2005) model of causality. However, the notion of strong inconsistency might not be appropriate in all cases as it depends on the structure of the *whole* knowledge base. Even though $S^{\mathcal{I}_{\mathsf{MSI}}}(\mathcal{K}, \alpha)$ is capable of capturing a notion of "reward" for restoring inconsistency, the underlying measure $\mathcal{I}_{\mathsf{MSI}}$ is actually tailored for consideration of a whole knowledge base $\mathcal{K}$. Investigation of culpability measures for non-monotonic logics is thus left for future work.

9. Summary and Conclusion

In this paper, we gave a comprehensive account on the challenges of handling and measuring inconsistency in general, i. e., not necessarily monotonic, logics using the concept of strong inconsistency. In particular, using a very general notion of a logic, we generalized three popular approaches to inconsistency measurement from the literature. Furthermore, we generalized many of the existing rationality postulates for inconsistency measurement as well and presented some new ones, particularly focusing on measure relating to monotonicity properties. We investigated the compliance of the discussed measures wrt. the postulates and also generalized the notion of IG measures (De Bona et al., 2018). It has to be noted here that our analysis is general insofar as it does not depend on the actual logic. All results are valid wrt. to any logic that can be cast into the form of Definition 2.13.

We continued with an analysis of the computational complexity of various problem related to inconsistency measurement. We obtained various membership results which are parameterized by the complexity of the satisfiability problem of the used logic and, as an example, provided hardness results if the used logic is answer set programming. As another contribution, we had a closer look at the influence of context in measuring (and repairing) inconsistencies. In contrast to monotonic logics, inconsistency may be repaired in non-monotonic logics by *adding* information. For that we assumed the existence of a set $\mathcal{G}$ containing information that may be added to the knowledge base if it helps in restoring consistency. We phrased novel hitting set dualities for this setting, characterising the relationship between adjusted notions of maximal consistent and minimal inconsistent sets. We then also investigated the problem of measuring inconsistency in this setting. Finally, we also looked briefly at applying the notion of strong inconsistency to define culpability measures.

In our analysis we focused on three specific inconsistency measures: $\mathcal{I}_{\mathsf{MSI}}$, $\mathcal{I}_{\mathsf{MSI}^{\mathsf{C}}}$, and $\mathcal{I}_{\mathsf{p}}$. However, we can find further measures making use of minimal inconsistent subsets in the literature (Jabbour et al., 2016; Jabbour and Sais, 2016). These measures take a closer look at the relationships between minimal inconsistent subsets—i. e., whether there are disjoint sets or many sets having large overlaps—and take this into account to provide a finer assessment of the inconsistency. In the same vein as we did for $\mathcal{I}_{\mathsf{MSI}}$, $\mathcal{I}_{\mathsf{MSI}^{\mathsf{C}}}$, and $\mathcal{I}_{\mathsf{p}}$, we can generalize these measures as well, simply by substituting the set of minimal inconsistent sets by the set of minimal strong inconsistent sets in their formal definitions. Investigating these measures could be a fruitful endeavour for future work.

Again, it has to be noted that the developed approaches and our analysis is general wrt. to the underlying formalism and encompasses many different logical instantiations such as classical logic, answer set programming, abstract argumentation, and many others. In some of these logics, the challenges of handling and measuring inconsistency have been addressed only to a limited extent before or not at all. For example, in abstract argumentation the only works considering a similar challenge are those by Hunter (2017) and Amgoud and Ben-Naim (2017). They propose measures to assess the "disagreement" in abstract argumentation frameworks, which is more about assessing the amount of conflicts between arguments than our notion of inconsistency (which is defined for that logic as the absence of extensions). Our measures can be readily applied to formalisms such as abstract argumentation and may provide new insights on their semantics.

Acknowledgements

This work was partially funded by Deutsche Forschungsgemeinschaft DFG (Research Training Group 1763; project BR 1817/7-2; project DE 1983/9-1).

References

Amgoud, L., Ben-Naim, J., 2017. Measuring disagreement in argumentation graphs. In: Moral, S., Pivert, O., Sánchez, D., Marín, N. (Eds.), Proceedings of the 11th International Conference on Scalable Uncertainty Management (SUM'17). pp. 208–222.

Atkinson, K., Baroni, P., Giacomin, M., Hunter, A., Prakken, H., Reed, C., Simari, G. R., Thimm, M., Villata, S., October 2017. Toward artificial argumentation. AI Magazine 38 (3), 25–36.

Baroni, P., Caminada, M., Giacomin, M., 2018. Abstract argumentation frameworks and their semantics. In: Baroni, P., Gabbay, D., Giacomin, M., van der Torre, L. (Eds.), Handbook of Formal Argumentation. College Publications, pp. 159–236.

Berge, C., 1989. Hypergraphs: Combinatorics of finite sets, vol. 45.

Besnard, P., 2014. Revisiting postulates for inconsistency measures. In: Proceedings of the 14th European Conference on Logics in Artificial Intelligence (JELIA'14). pp. 383–396.

Besnard, P., 2017. Basic postulates for inconsistency measures. Transactions Large-Scale Data- and Knowledge-Centered Systems 34, 1–12.

Béziau, J.-Y., Carnielli, W., Gabbay, D. (Eds.), 2007. Handbook of Paraconsistency. College Publications, London.

Brewka, G., Eiter, T., 2007. Equilibria in heterogeneous nonmonotonic multi-context systems. In: Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, July 22-26, 2007, Vancouver, British Columbia, Canada. pp. 385–390.

Brewka, G., Eiter, T., Truszczynski, M., 2011. Answer set programming at a glance. Commun. ACM 54 (12), 92–103.

Brewka, G., Thimm, M., Ulbricht, M., 2017. Strong inconsistency in non-monotonic reasoning. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17. pp. 901–907.

Brewka, G., Thimm, M., Ulbricht, M., 2019. Strong inconsistency. Artificial Intelligence 267, 78–117.

Cholvy, L., Hunter, A., 1997. Information fusion in logic: A brief overview. In: Qualitative and Quantitative Practical Reasoning (ECSQARU'97/FAPR'97). Vol. 1244 of Lecture Notes in Computer Science. Springer, pp. 86–95.

Condotta, J.-F., Raddaoui, B., 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). pp. 443–452.

Daniel, L., December 2009. Paraconsistent Probabilistic Reasoning. Ph.D. thesis, L'École Nationale Supérieure des Mines de Paris.

De Bona, G., Finger, M., 2015. Measuring inconsistency in probabilistic logic: Rationality postulates and dutch book interpretation. Artificial Intelligence 227, 140–164.

De Bona, G., Grant, J., Hunter, A., Konieczny, S., 2018. Towards a unified framework for syntactic inconsistency measures. In: Proceedings of the 32nd AAAI Conference on Artificial Intelligence. pp. 1803–1810.

De Bona, G., Hunter, A., 2017. Localising iceberg inconsistencies. Artificial Intelligence 246, 118–151.

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.

Durand, A., Hermann, M., Kolaitis, P. G., 2005. Subtractive reductions and complete problems for counting complexity classes. Theoretical Computer Science 340 (3), 496–513.

Eiter, T., Fink, M., Tompits, H., Woltran, S., 2005. Strong and uniform equivalence in answer-set programming: Characterizations and complexity results for the non-ground case. In: Proceedings, The Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference, July 9-13, 2005, Pittsburgh, Pennsylvania, USA. pp. 695–700.

Eiter, T., 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., Leone, N., 2002. Logic programming and knowledge representation – the A-Prolog perspective. Artificial Intelligence 138 (1–2), 3–38.

Gelfond, M., Lifschitz, V., 1991. Classical negation in logic programs and disjunctive databases. New Generation Comput. 9 (3/4), 365–386.

Grant, J., Hunter, A., 2006. Measuring Inconsistency in Knowledgebases. Journal of Intelligent Information Systems 27, 159–184.

Grant, J., Hunter, A., 2011. Measuring consistency gain and information loss in stepwise inconsistency resolution. In: European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty. Springer, pp. 362–373.

Halpern, J. Y., Pearl, J., 2005. Causes and explanations: A structural-model approach. part ii: Explanations. The British Journal for the Philosophy of Science 56 (4), 889–911.

Hansson, S. O., 2001. A Textbook of Belief Dynamics. Kluwer Academic Publishers, Norwell, MA, USA.

Hemaspaandra, L. A., Vollmer, H., 1995. The satanic notations: counting classes beyond# p and other definitional adventures. ACM SIGACT News 26 (1), 2–13.

Hunter, A., 2017. Measuring inconsistency in argument graphs. Tech. rep., University College London.

Hunter, A., Konieczny, S., 2004. Approaches to Measuring Inconsistent Information. In: Inconsistency Tolerance. Vol. 3300 of Lecture Notes in Computer Science. Springer International Publishing, pp. 189–234.

Hunter, A., 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). AAAI Press, pp. 358–366.

Hunter, A., Konieczny, S., July 2010. On the measure of conflicts: Shapley inconsistency values. Artificial Intelligence 174 (14), 1007–1026.

Jabbour, S., Ma, Y., Raddaoui, B., Sais, L., 2014. On the characterization of inconsistency: A prime implicates based framework. In: Tools with Artificial Intelligence (ICTAI), 2014 IEEE 26th International Conference on. IEEE, pp. 146–153.

Jabbour, S., Ma, Y., Raddaoui, B., Sais, L., 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). pp. 84–93.

Jabbour, S., Sais, L., 2016. Exploiting MUS Structure to Measure Inconsistency of Knowledge Bases. In: Proceedings of the 22nd European Conference on Artificial Intelligence (ECAI'16). pp. 991–998.

Knight, K. M., 2001. Measuring inconsistency. Journal of Philosophical Logic 31, 77–98.

Konieczny, S., Pérez, R. P., 1998. On the logic of merging. In: Proceedings of the Sixth International Conference on Principles of Knowledge Representationa and Reasoning (KR'98). Morgan Kaufmann, pp. 488–498.

Lifschitz, V., Pearce, D., Valverde, A., 2001. Strongly equivalent logic programs. ACM Trans. Comput. Log. 2 (4), 526–541.

Lozinskii, E. L., 1994. Information and Evidence in Logic Systems. Journal of Experimental and Theoretical Artificial Intelligence 6, 163–193.

Madrid, N., Ojeda-Aciego, M., 2010. Measuring instability in normal residuated logic programs: adding information. In: Proceedings of the 19th IEEE International Conference on Fuzzy Systems. pp. 1–7.

Mu, K., 2015. Responsibility for inconsistency. International Journal of Approximate Reasoning 61, 43–60.

Mu, K., 2018. Measuring inconsistency with constraints for propositional knowledge bases. Artificial Intelligence 259, 52–90.

Mu, K., 2019. Formulas free from inconsistency: An atom-centric characterization in priest's minimally inconsistent lp. Journal of Artificial Intelligence Research 66, 279–296.

Mu, K., Liu, W., Jin, Z., 2011. A general framework for measuring inconsistency through minimal inconsistent sets. Knowledge and Information Systems 27 (1), 85–114.

Oikarinen, E., Woltran, S., 2011. Characterizing strong equivalence for argumentation frameworks. Artif. Intell. 175 (14-15), 1985–2009.

Papadimitriou, C., 1994. Computational Complexity. Addison-Wesley.

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). pp. 568–577.

Reiter, R., 1980. A logic for default reasoning. Artificial Intelligence 13, 81–132.

Reiter, R., 1987. A theory of diagnosis from first principles. Artif. Intell. 32 (1), 57–95.

Schulz, C., Satoh, K., Toni, F., 2015. Characterising and explaining inconsistency in logic programs. In: Calimeri, F., Ianni, G., Truszczynski, M. (Eds.), Logic Programming and Nonmonotonic Reasoning: 13th International Conference, LPNMR 2015, Lexington, KY, USA, September 27-30, 2015. Proceedings. Springer International Publishing, Cham, pp. 467–479.

Shannon, C. E., 1948. A Mathematical Theory of Communication. Bell System Technical Journal 27, 379–423.

Shapley, L. S., 1953. A value for n-person games. Contributions to the Theory of Games 2 (28), 307–317.

Thimm, M., 2009. Measuring inconsistency in probabilistic knowledge bases. In: Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence. AUAI Press, pp. 530–537.

Thimm, M., 2013. Inconsistency Measures for Probabilistic Logics. Artificial Intelligence 197, 1–24.

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.

Thimm, M., February 2018. On the evaluation of inconsistency measures. In: Grant, J., Martinez, M. V. (Eds.), Measuring Inconsistency in Information. Vol. 73 of Studies in Logic. College Publications.

Thimm, M., Wallner, J. P., April 2016. Some complexity results on inconsistency measurement. In: Proceedings of the 15th International Conference on Principles of Knowledge Representation and Reasoning (KR'16). pp. 114–124.

Thimm, M., Wallner, J. P., October 2019. On the complexity of inconsistency measurement. Artificial Intelligence 275, 411–456.

Ulbricht, M., 2019. Repairing non-monotonic knowledge bases. In: Calimeri, F., Leone, N., Manna, M. (Eds.), Logics in Artificial Intelligence. Springer International Publishing, Cham, pp. 151–167.

Ulbricht, M., Thimm, M., Brewka, G., November 2016. Measuring Inconsistency in Answer Set Programs. In: Proceedings of the 15th European Conference on Logics in Artificial Intelligence (JELIA'16). pp. 577–583.

Ulbricht, M., Thimm, M., Brewka, G., February 2018. Measuring strong inconsistency. In: Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI'18). pp. 1989–1996.

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.

Appendix A. Technical Proofs of Section 6

Lemma 6.4. *Given an open QBF $\Phi = \forall Y\, \phi(X,Y)$, there is a disjunction-free logic program $P(\Phi) \subseteq \mathsf{WF}^{ASP^*}$ of polynomial size with*

$$|SI_{min}(P(\Phi))| = |X| + |\mathsf{Mod}(\Phi)|.$$

Proof. Let $X = \{x_1, \ldots, x_n\}$ and $Y = \{y_1, \ldots, y_m\}$. We abuse notation and write $\phi(X,Y) = 1$ ($\phi(X,Y) = 0$) if ϕ evaluates to true (false) under a given assignment to the X and Y variables, so we have

$$\begin{aligned} |\mathsf{Mod}(\Phi)| &= |\{X \mid \forall Y\, \phi(X,Y) = 1)\}| \\ &= |\{X \mid \forall Y\, \neg\phi(X,Y) = 0\}|. \end{aligned}$$

We can assume ϕ to be a formula in 3-DNF and thus, $\neg\phi$ is in 3-CNF, i. e., the conjunction of $C_1, \ldots, C_r$ with $C_k = l_{k,1} \vee \ldots \vee l_{k,3}$. Let $x_1', \ldots, x_n', y_1', \ldots, y_m'$ be fresh atoms. Intuitively, they shall correspond to the negated atoms $\neg x_1, \ldots, \neg x_n, \neg y_1, \ldots, \neg y_m$. Let σ be the appropriate mapping, i. e.,

$$\sigma(a) = \begin{cases} 7a & \text{if } a \in X \cup Y, \\ a' & \text{if } a \in \{\neg x_1, \ldots, \neg x_n\} \cup \{\neg y_1, \ldots, \neg y_m\}. \end{cases}$$

We construct a program $P = P(\Phi)$ whose minimal strongly inconsistent subsets correspond to $\{X \mid \forall Y\, \neg\phi(X,Y) = 0\}$. We include a fresh atom w not occurring in $X \cup Y$ which is going to witness whether $\neg\phi$ is satisfied. Hence, P contains a constraint

$$\leftarrow \text{not } w.$$

Moreover, $\neg\phi$ is satisfied if all conjuncts are true, where a conjunct $C_k = l_{k,1} \vee \ldots \vee l_{k,3}$ is true if one of the literals occurring in C_k is true. Thus, we introduce atoms $w_1, \ldots, w_r$ and rules

$$\begin{array}{lll} w_1 \leftarrow \sigma(l_{1,1}). & \ldots & w_1 \leftarrow \sigma(l_{1,3}). \\ \ldots & \ldots & \ldots \\ w_r \leftarrow \sigma(l_{r,1}). & \ldots & w_r \leftarrow \sigma(l_{r,3}). \\ w \leftarrow w_1, \ldots w_r, w_{x_1} \ldots, w_{x_n}. & & \end{array}$$

The meaning of the atoms w_{x_i} is as follows:

We want subsets of the program P to correspond to assignments to the X-variables. Thus, we introduce rules "x_i." and "x_i'." for $i = 1, \ldots, n$. A subprogram $H \subseteq P$ may contain both "x_i." and "x_i'." for an i. We want to ensure that H can only be consistent if "x_i." or "x_i.'" is *not* contained in H. That is the reason we need the atoms of the form w_{x_i} in the rule "$w \leftarrow w_1, \ldots w_r, w_{x_1} \ldots, w_{x_n}$.". We include the following rules.

$$
\begin{array}{ll}
x_1., \ldots, x_n. & x_1'., \ldots, x_n'. \\
w_{x_1} \leftarrow \text{not } x_1. & w_{x_1} \leftarrow \text{not } x_1'. \\
\ldots & \ldots \\
w_{x_n} \leftarrow \text{not } x_n. & w_{x_n} \leftarrow \text{not } x_n'.
\end{array}
$$

Now, if H does not contain both "x_i." and "x_i.'", then the corresponding rule can be added to H to ensure w_{x_i} is entailed. This way we check that a subset $\mathcal{H} \subseteq P$ (which is either strongly inconsistent or not) corresponds to a proper assignment to the X-variables.

Moreover, we want to make sure that a strongly inconsistent subset corresponds to an assignment to *all* X-variables (and not to a partial assignment). In other words, if $H \subseteq P$ does *not* model an assignment to *all* X-variables, then H is *not* supposed to be strongly inconsistent. We thus ensure existence of a consistent superset H' with $H \subseteq H' \subseteq P$ as follows: We allow entailment of w if "x_i." or "x_i'." is missing for any i. This shall, however, only work for proper assignments, i. e., w_{x_i} is required for any i. We hence introduce a rule "$w \leftarrow \text{not } x_i, \text{not } x_i', w_{x_1}, ..., w_{x_n}$" for $i = 1, ..., n$. Now, if a subset H of P does not contain "x_i." or "x_i'." (and not both) for any $i = 1, \ldots, n$, we see that adding the corresponding rule to H ensures w is entailed, rendering H consistent. We include:

$$
w \leftarrow \text{not } x_i, \text{not } x_i', w_{x_1}, ..., w_{x_n}. \qquad\qquad i = 1, ..., n
$$

Since our goal is counting $|\{X \mid \forall Y \neg\phi(X, Y) = 0\}|$, we do not want rules corresponding to the choice of Y-variables to occur in a minimal strongly P-inconsistent subset H. The following translates assignments to Y-variables without any restrictions:

$$
\begin{array}{ll}
y_1 \leftarrow \text{not } y_1'. & y_1' \leftarrow \text{not } y_1. \\
\ldots & \ldots \\
y_m \leftarrow \text{not } y_m'. & y_m' \leftarrow \text{not } y_m.
\end{array}
$$

To summarize, P is given as follows.

$$
\begin{array}{lll}
P: & & \\
x_1., \ldots, x_n. & & \\
x'_1., \ldots, x'_n. & & \\
w \leftarrow \text{not } x_i, \text{not } x'_i, w_{x_1}, ..., w_{x_n}. & & i = 1, ..., n \\
w_{x_i} \leftarrow \text{not } x_i. & w_{x_i} \leftarrow \text{not } x'_i. & i = 1, ..., n \\
y_j \leftarrow \text{not } y'_j. & y'_j \leftarrow \text{not } y_j. & j = 1, ..., m \\
w_k \leftarrow \sigma(l_{k,1}). & & k = 1, ..., r \\
w_k \leftarrow \sigma(l_{k,1}). & & k = 1, ..., r \\
w_k \leftarrow \sigma(l_{k,3}). & & k = 1, ..., r \\
w \leftarrow w_1, \ldots w_r, w_{x_1} \ldots, w_{x_n}. & & \\
\leftarrow \text{not } w. & &
\end{array}
$$

Note that the construction is polynomial. We show

$$|SI_{min}(P)| = |X| + |\{X \mid \forall Y \neg\phi(X, Y) = 0\}|.$$

We make a few observations in order to obtain this result.

(a) Any inconsistent subset of P contains the constraint "$\leftarrow$ not w." and the inconsistency stems from it.

(b) Let $H \in SI_{min}(P)$. Then, H only contains "$\leftarrow$ not w." and rules of the form "x_i." or "x'_i.".

For this, let $W \subseteq P$ be the following program

$$
\begin{array}{lll}
W: & & \\
w \leftarrow \text{not } x_i, \text{not } x'_i, w_{x_1}, ..., w_{x_n}. & & i = 1, ..., n \\
w_{x_i} \leftarrow \text{not } x_i. & w_{x_i} \leftarrow \text{not } x'_i. & i = 1, ..., n \\
y_j \leftarrow \text{not } y'_j. & y'_j \leftarrow \text{not } y_j. & j = 1, ..., m \\
w_k \leftarrow \sigma(l_{k,1}). & & k = 1, ..., r \\
w_k \leftarrow \sigma(l_{k,1}). & & k = 1, ..., r \\
w_k \leftarrow \sigma(l_{k,3}). & & k = 1, ..., r \\
w \leftarrow w_1, \ldots w_r, w_{x_1} \ldots, w_{x_n}. & &
\end{array}
$$

It is easy to see that rules in W can never introduce inconsistency, because they facilitate entailment of w. Hence, a set $H \in SI_{min}(P)$ consists of rules in $P \setminus W$.

(c) Let $H \in SI_{min}(P)$. If H does not contain both "x_i." and "x'_i." for an $i \in \{1, ..., n\}$, then it contains either "x_i." or "x'_i." for *all* $i \in \{1, ..., n\}$.

For this, assume neither "x_i." nor "x'_i." is in H. Augment H with the following rules:

- "$w \leftarrow \text{not } x_i, \text{not } x'_i, w_{x_1}, ..., w_{x_n}.$",
- "$w_{x_j} \leftarrow \text{not } x_j.$" and "$w_{x_j} \leftarrow \text{not } x'_j.$" for $j = 1, ..., n$.

We obtain a consistent program since w is entailed now (cf. (a)). Hence, $H \notin SI(P)$.

Now, we explicitly give the two kinds of minimal strongly P-inconsistent subsets. The first one corresponds to inconsistent assignments, the second one to assignments where $\forall Y \neg\phi(X, Y) = 0$ holds.

(d) Let $i \in \{1, ..., n\}$ and let $H_i := \{x_i.,\ x'_i.,\ \leftarrow \text{not } w.\}$. Then, $H_i \in SI_{min}(P)$.

Since satisfying the constraint "$\leftarrow$ not w." requires the atom w_{x_i}, any program H with $H_i \subseteq H \subseteq P$ is clearly inconsistent. Hence, $H_i \in SI(P)$. For minimality, assume "x_i." is removed from H_i. Augment the obtained subprogram with the following rules:

- "$w \leftarrow \text{not } x_j, \text{not } x'_j, w_{x_1}, ..., w_{x_n}.$" for any $j \neq i$ (w. l. o. g. assume that $n \geq 2$),
- "$w_{x_j} \leftarrow \text{not } x_j.$" and "$w_{x_j} \leftarrow \text{not } x'_j.$" for $j = 1, ..., n$.

Since "x_i." is not contained in the subprogram anymore, w can be entailed now. Thus, we found a consistent superprogram. Hence, $H_i \setminus \{x_i.\} \notin SI(P)$. For the same reason, $H_i \setminus \{x'_i.\} \notin SI(P)$. The observation that $H_i \setminus \{\leftarrow \text{not } w.\} \notin SI(P)$ is trivial. Thus, H_i is minimal in $SI(P)$.

For our last step, let

$$H_I = \bigcup_{i \in \{1,...,n\}} H_i. \tag{A.1}$$

As we can conclude now, $H \in SI_{min}(P) \setminus H_I$ contains either "x_i." or "x'_i." for all indices $i \in \{1, ..., n\}$. More general, let $\mathcal{H}_\Omega \subseteq 2^P$ be the set of all subprograms of P containing either "x_i." or "x'_i." for all $i \in \{1, ..., n\}$. Hence for $H \in \mathcal{H}_\Omega$, it makes sense to define a corresponding assignment $\omega(H) : X \to \{0, 1\}$ with

$$\omega(H)(x_i) := \begin{cases} 1 & \text{if } x_i. \in H, \\ 0 & \text{if } x'_i. \in H. \end{cases}$$

We are now ready to prove our last step.

(e) Let $H \in \mathcal{H}_\Omega$. Then, $H \in SI_{min}(P) \setminus H_I$ if and only if $\forall Y\, (\neg\phi(X, Y) = 0)$ holds for the assignment $\omega(H)$ to the X-variables.

"$\Rightarrow$": Let $H \in SI_{min}(P) \setminus H_I$. Assume $\forall Y\, (\neg\phi(X, Y) = 0)$ is false for the assignment $\omega(H)$, i. e., $\exists Y\, \neg\phi(X, Y)$ holds. Consider the program $H \cup W$ where W is as in (b). Any answer set of W corresponds to one particular assignment to the Y-variables. By construction of P and since $\exists Y\, \neg\phi(X, Y)$ holds, $H \cup W$ has a stable model M with $w \in M$. Hence, $H \cup W$ is consistent and we conclude $H \notin SI_{min}(P)$.

"$\Leftarrow$": Minimality of H in $SI(P)$ follows from the observations we made above. Assume H' is consistent with $H \subseteq H'$. Due to (d), $H' \setminus H \subseteq W$ with W as above, because adding additional rules of the form "x_i." or "x'_i." to H renders the subprogram inconsistent. However, rules in W do not introduce inconsistency. Hence H' with $H \subseteq H' \subseteq H \cup W$ being consistent implies that $H \cup W$ is consistent as well. As above, we conclude $\exists Y\, \phi(X, Y)$ is true.

In (d) and (e), we found two possible cases for sets $H \in SI_{min}(P)$. Due to (b) and (c), no other case occurs. Thus,

$$|SI_{min}(P)| = |X| + |\{X \mid \forall Y\, \neg\phi(X, Y) = 0\}|$$

is proved. ☐

Theorem 6.7. *The problem* $\textsc{Value}^{L^{ASP^*}}_{\mathcal{I}_{\mathsf{MSI}}}$ *is* $\#\cdot$*coNP-complete under subtractive reductions.*

Proof. As membership is due to Theorem 6.3, we prove hardness. Consider an open QBF $\Phi = \forall Y\, \phi(X, Y)$. We use the same construction (and the same notation for ϕ, X and Y) as in Lemma 6.4, i. e., consider $P = P(\Phi)$ given as follows:

$$
\begin{array}{lll}
P: & & \\
x_1., \ldots, x_n. & & \\
x'_1., \ldots, x'_n. & & \\
w \leftarrow \text{not } x_i, \text{not } x'_i, w_{x_1}, ..., w_{x_n}. & & i = 1, ..., n \\
w_{x_i} \leftarrow \text{not } x_i. & w_{x_i} \leftarrow \text{not } x'_i. & i = 1, ..., n \\
y_j \leftarrow \text{not } y'_j. & y'_j \leftarrow \text{not } y_j. & j = 1, ..., m \\
w_k \leftarrow \sigma(l_{k,1}). & & k = 1, ..., r \\
w_k \leftarrow \sigma(l_{k,1}). & & k = 1, ..., r \\
w_k \leftarrow \sigma(l_{k,3}). & & k = 1, ..., r \\
w \leftarrow w_1, \ldots w_r, w_{x_1} \ldots, w_{x_n}. & & \\
\leftarrow \text{not } w. & &
\end{array}
$$

Recall that P yields $|SI_{min}(P(\Phi))| = |X| + |\mathsf{Mod}(\Phi)|$. So for our subtractive reduction we require a program P' with $|SI_{min}(P')| = |X|$ and $SI_{min}(P') \subseteq SI_{min}(P)$. For this, consider $P' = P'(\Phi)$ as follows:

$$
\begin{array}{lll}
P': & & \\
x_1., \ldots, x_n. & & \\
x'_1., \ldots, x'_n. & & \\
w \leftarrow \text{not } x_i, \text{not } x'_i, w_{x_1}, ..., w_{x_n}. & & i = 1, ..., n \\
w_{x_i} \leftarrow \text{not } x_i. & w_{x_i} \leftarrow \text{not } x'_i. & i = 1, ..., n \\
\leftarrow \text{not } w. & &
\end{array}
$$

It is easy to see that $SI_{min}(P') = H_I$ with H_I as in (A.1). In particular,

$$SI_{min}(P') \subseteq SI_{min}(P)$$

and $|SI_{min}(P')| = |X|$. Since

$$
\begin{aligned}
|\{X \mid \forall Y\, \phi(X, Y)\}| &= |SI_{min}(P)| - |X| \\
&= |SI_{min}(P)| - |SI_{min}(P')|
\end{aligned}
$$

follows, we found a subtractive reduction. $\square$

Theorem 6.10. *Let $m \geq 1$. If the decision problem* $\textsc{Sat}_L$ *is in*

(a) Σ_m^p*, then* $\textsc{Lower}^L_{\mathcal{I}_\mathsf{p}}$ *is in* Σ_{m+1}^p*,*

(b) Π_m^p*, then* $\textsc{Lower}^L_{\mathcal{I}_\mathsf{p}}$ *is in* Σ_{m+2}^p*,*

(c) Π_m^p *and L is monotonic, then* $\textsc{Lower}^L_{\mathcal{I}_\mathsf{p}}$ *is in* Σ_{m+1}^p*.*

Proof. Note that $\mathcal{I}_\mathsf{p}(\mathcal{K}) \leq |\mathcal{K}|$. Given an integer $k \leq |\mathcal{K}|$, guess a set $\mathcal{H} \subseteq \mathcal{K}$ with $|\mathcal{H}| \geq k$ and for any $\alpha \in \mathcal{H}$ guess a set $\mathcal{H}_\alpha$. Perform (linearly many) satisfiability checks to verify that $\mathcal{H}_\alpha$ is minimal strongly $\mathcal{K}$-inconsistent. Due to Brewka et al. (2017), the latter check is in

(a) D_m^p if $\textsc{Sat}_L$ is in Σ_m^p,

(b) D_{m+1}^p if $\textsc{Sat}_L$ is in Π_m^p,

(c) D_m^p if $\textsc{Sat}_L$ is in Π_m^p and L is monotonic.

Due to the nondeterministic guess of $\mathcal{H}$ and the sets $\mathcal{H}_\alpha$, this algorithm is in

(a) Σ_{m+1}^p if $\textsc{Sat}_L$ is in Σ_m^p,

(b) Σ_{m+2}^p if $\textsc{Sat}_L$ is in Π_m^p,

(c) Σ_{m+1}^p if $\textsc{Sat}_L$ is in Π_m^p and L is monotonic. □

Corollary 6.11. *Let $m \geq 81$. If the decision problem* $\textsc{Sat}_L$ *is in*

(a) Σ_m^p*, then* $\textsc{Upper}^L_{\mathcal{I}_\mathsf{p}}$ *is in* Π_{m+1}^p*,* $\textsc{Exact}^L_{\mathcal{I}_\mathsf{p}}$ *is in* D_{m+1}^p *and* $\textsc{Value}^L_{\mathcal{I}_\mathsf{p}}$ *is in* $\mathsf{FP}^{\Sigma_{m+1}^p[\log n]}$*,*

(b) Π_m^p*, then* $\textsc{Upper}^L_{\mathcal{I}_\mathsf{p}}$ *is in* Π_{m+2}^p*,* $\textsc{Exact}^L_{\mathcal{I}_\mathsf{p}}$ *is in* D_{m+2}^p *and* $\textsc{Value}^L_{\mathcal{I}_\mathsf{p}}$ *is in* $\mathsf{FP}^{\Sigma_{m+2}^p[\log n]}$*,*

(c) Π_m^p *and L is monotonic, then* $\textsc{Upper}^L_{\mathcal{I}_\mathsf{p}}$ *is in* Π_{m+1}^p*,* $\textsc{Exact}^L_{\mathcal{I}_\mathsf{p}}$ *is in* D_{m+1}^p *and* $\textsc{Value}^L_{\mathcal{I}_\mathsf{p}}$ *is in* $\mathsf{FP}^{\Sigma_{m+1}^p[\log n]}$*.*

Proof. Follows from Lemmas 4 and 6 by Thimm and Wallner (2019). □

Proposition 6.12. *The problems* $\textsc{Lower}^{L^{ASP^*}}_{\mathcal{I}_{\mathsf{MSI^C}}}$ *and* $\textsc{Upper}^{L^{ASP^*}}_{\mathcal{I}_{\mathsf{MSI^C}}}$ *are* CNP*-hard,* $\textsc{Exact}^{L^{ASP^*}}_{\mathcal{I}_{\mathsf{MSI^C}}}$ *is* $\mathsf{C_{=}NP}$*-hard.*

Proof. Given an open QBF $\Phi = \forall Y\, \phi(X,Y)$, consider the program $P = P(\Phi)$ as in Lemma 6.4. It contains $|X|$ minimal strongly P-inconsistent subsets of the form $\{x_i.,\ x_i',\ \text{not } w.\}$ (case (d) in the proof of Lemma 6.4) as well as $|\mathsf{Mod}(\Phi)|$ many of size $|X|+1$ (case (e) in the proof of Lemma 6.4). Hence,

$$\mathcal{I}_{\mathsf{MSI^c}}(P(\Phi)) = \sum_{H \in SI_{min}(P)} \frac{1}{|H|} = \frac{|X|}{3} + \frac{\mathsf{Mod}(\Phi)}{|X|+1}$$

and hence, for any integer k, $|\mathsf{Mod}(\Phi)| \geq k$ resp. $|\mathsf{Mod}(\Phi)| = k$ holds iff $\mathcal{I}_{\mathsf{MSI^c}}(P(\Phi)) \geq \frac{|X|}{3} + \frac{k}{|X|+1}$ resp. $\mathcal{I}_{\mathsf{MSI^c}}(P(\Phi)) = \frac{|X|}{3} + \frac{k}{|X|+1}$ holds. $\square$

Proposition 6.13. *The problem* $\textsc{Lower}_{\mathcal{I}_\mathsf{p}}^{L^{ASP^*}}$ *is* Σ_2^p*-complete. The problem* $\textsc{Upper}_{\mathcal{I}_\mathsf{p}}^{L^{ASP^*}}$ *is* Π_2^p*-complete. The problem* $\textsc{Exact}_{\mathcal{I}_\mathsf{p}}^{L^{ASP^*}}$ *is* D_2^p*-complete.*

Proof. Membership follows from Theorem 6.10. For hardness, assume we are given an open QBF $\Phi = \forall Y\, \phi(X,Y)$. Deciding whether $|\mathsf{Mod}(\Phi)| \geq 1$ holds is Σ_2^p-complete in general. To prove Lemma 6.4, we already gave a program $P = P(\Phi)$ with

$$|SI_{min}(P)| = |X| + |\mathsf{Mod}(\Phi)|.$$

We make use of the same notations and auxiliary atoms. Consider $P = P(\Phi)$ again.

$$\begin{array}{lll}
P: & & \\
x_1., \dots, x_n. & & \\
x_1'., \dots, x_n'. & & \\
w \leftarrow \text{not } x_i, \text{not } x_i', w_{x_1}, ..., w_{x_n}. & & i = 1, ..., n \\
w_{x_i} \leftarrow \text{not } x_i. & w_{x_i} \leftarrow \text{not } x_i'. & i = 1, ..., n \\
y_j \leftarrow \text{not } y_j'. & y_j' \leftarrow \text{not } y_j. & j = 1, ..., m \\
w_k \leftarrow \sigma(l_{k,1}). & & k = 1, ..., r \\
w_k \leftarrow \sigma(l_{k,1}). & & k = 1, ..., r \\
w_k \leftarrow \sigma(l_{k,3}). & & k = 1, ..., r \\
\leftarrow \text{not } w. & &
\end{array}$$

Our considerations above show that the rules occurring in the minimal strongly P-inconsistent subsets of P are "$x_i.$" and "$x_i'.$" for $i = 1, ..., n$ as well as

"$\leftarrow$ not w.". We thus see that $\mathcal{I}_{\mathsf{p}}(P) = 2n+1$ holds no matter how many models Φ has. Recall: $H \in SI_{min}(P)$ is either of the form

$$H = \{x_i.,\ x_i'.,\ \leftarrow \text{not } w.\}$$

(corresponding to an improper assignment) or it contains rules of the form "x_i." resp. "x_i'." corresponding to an assignment that is a model of Φ.

We now introduce some more rules in order to be able to tell which case occurs. The goal is to find out whether *at least one* $H \in SI_{min}(P)$ corresponds to an assignment. For that, we keep track of assignments as follows. Similar to the rules

$$w_{x_i} \leftarrow \text{not } x_i. \qquad\qquad w_{x_i} \leftarrow \text{not } x_i'.$$

witnessing that a proper assignment is given, we introduce

$$v_{x_i} \leftarrow x_i. \qquad\qquad v_{x_i} \leftarrow x_i'.$$

witnessing that at least one of the rules "x_i." and "x_i'." is chosen. Now, we can make sure that a set H corresponding to an assignment to the X-variables is never strongly P-inconsistent by additionally adding

$$w \leftarrow w_{x_1}, \ldots, w_{x_n}, v_{x_1}, \ldots, v_{x_n}, \text{not } a.$$

Moreover adding

$$a.$$

to the program makes sure that this rule is not always applicable. So consider

$Q = Q(\Phi)$ given as follows.

$$
\begin{array}{ll}
Q: & \\
x_1., \ldots, x_n. & \\
x'_1., \ldots, x'_n. & \\
w \leftarrow \text{not } x_i, \text{not } x'_i, w_{x_1}, ..., w_{x_n}. & i = 1, ..., n \\
w_{x_i} \leftarrow \text{not } x_i. & \\
w_{x_i} \leftarrow \text{not } x'_i. & i = 1, ..., n \\
y_j \leftarrow \text{not } y'_j. & \\
y'_j \leftarrow \text{not } y_j. & j = 1, ..., m \\
w_k \leftarrow \sigma(l_{k,1}). & k = 1, ..., r \\
w_k \leftarrow \sigma(l_{k,1}). & k = 1, ..., r \\
w_k \leftarrow \sigma(l_{k,3}). & k = 1, ..., r \\
w \leftarrow w_1, \ldots w_r, w_{x_1} \ldots, w_{x_n}. & \\
v_{x_i} \leftarrow x_i. & \\
v_{x_i} \leftarrow x'_i. & i = 1, ..., n \\
w \leftarrow w_{x_1}, \ldots, w_{x_n}, v_{x_1}, \ldots, v_{x_n}, \text{not } a. & \\
a. & \\
\leftarrow \text{not } w. &
\end{array}
$$

We do not repeat all the considerations from Lemma 6.4 since the programs P and Q are rather similar. We note however that

$$H_i = \{x_i., \; x'_i., \; \leftarrow \text{not } w.\} \in SI_{min}(Q) \qquad i = 1, ..., n$$

holds. Thus, $\mathcal{I}_{\mathsf{p}}(Q) \geq 2n + 1$. As pointed out in Lemma 6.4, a program $H \subseteq P$ that contains either "$x_i.$" or "$x'_i.$" for each i as well as "$\leftarrow$ not $w.$" is in $SI_{min}(P)$ if and only if the choice of the X-rules corresponds to an assignment ω to the X-variables with $\omega \in \mathsf{Mod}(\Phi)$. Due to the rules we added to Q, this is not the case anymore for $H \subseteq Q$ because the rules

$$w_{x_i} \leftarrow \text{not } x_i. \qquad\qquad w_{x_i} \leftarrow \text{not } x'_i.$$

and

$$v_{x_i} \leftarrow x_i. \qquad\qquad v_{x_i} \leftarrow x'_i.$$

can be added to H in order to obtain each w_{x_i} and v_{x_i} and then

$$w \leftarrow w_{x_1}, \ldots, w_{x_n}, v_{x_1}, \ldots, v_{x_n}, \text{not } a.$$

entails w. Hence, H contains a consistent superset within Q. Thus, $H \notin SI_{min}(Q)$. However, if such an $H \subseteq Q$ corresponds to an assignment $\omega \in \mathsf{Mod}(\Phi)$, this is the *only* way to entail w. This follows from what we established in Lemma 6.4. Hence, adding "a." renders H strongly Q-inconsistent.

Thus, we see: $\mathsf{Mod}(\Phi) \geq 1$ holds if and only if there is an $H \in SI_{min}(Q)$ that is not if the form

$$H_i = \{x_i.,\ x_i'.,\ \leftarrow \text{not } w.\}.$$

This is the case if and only if "a." occurs in an $H \in SI_{min}(Q)$. By definition of $\mathcal{I}_{\mathsf{p}}$ this is the case if and only if $\mathcal{I}_{\mathsf{p}}(Q) \geq 2n+2$.

It immediately follows that $\text{UPPER}_{\mathcal{I}_{\mathsf{p}}}^{L^{\text{ASP}^*}}$ is Π_2^p-complete.

Membership of $\text{EXACT}_{\mathcal{I}_{\mathsf{p}}}^{L^{\text{ASP}^*}}$ in D_2^p is due to Corollary 6.11. For hardness, assume we are given the generic D_2^p-complete problem where we are given two formulas

$$\Phi_j = \exists X \forall Y\, \phi_j(X, Y)$$

and have to decide whether Φ_1 is true while Φ_2 is false. We assume w. l. o. g. that they do not share any atoms. So let $X(\Phi_j) = \{x_1(\Phi_j), \ldots, x_{n(j)}(\Phi_j)\}$ and $Y(\Phi_j) = \{y_1(\Phi_j), \ldots, y_{m(j)}(\Phi_j)\}$ be the variables occurring in Φ_j. Roughly speaking, we apply the construction used in Prosition 6.13 two times, yielding the following programs Q_1 and Q_2. For Q_j, we assume that $x_i = x_i(\Phi_j)$, $w_{x_i} = w_{x_i}(\Phi_j)$ and so on and only the atom w occurs in both programs. Both programs are nearly the same: Q_1 is simply Q from Proposition 6.13 and we let

$$\begin{aligned} Q_2 = Q_1 \setminus \{&w \leftarrow w_{x_1}, \ldots, w_{x_n}, v_{x_1}, \ldots, v_{x_n}, \text{not } a.,\ a.\} \\ &\cup \{w \leftarrow w_{x_1}, \ldots, w_{x_n}, v_{x_1}, \ldots, v_{x_n}, \text{not } b, \text{not } c.,\ b.,\ c.\} \end{aligned}$$

Thus, the only difference is the appearance of b and c in Q_2 rather than just a. Now consider the program

$$Q = Q_1 \cup Q_2.$$

Now we see that $\mathcal{I}_{\mathsf{p}}(Q) \geq |X(\Phi_1)| + |X(\Phi_2)| + 1$, where the one stems from the constraint. Note that Φ_1 is true iff a occurs in an $H \in SI_{min}(Q)$ and Φ_2 is true iff b and c do. Hence, Φ_1 is true while Φ_2 is false iff $\mathcal{I}_{\mathsf{p}}(Q) = |X(\Phi_1)| + |X(\Phi_2)| + 2$. $\square$

Lemma 6.8. *Given an open QBF $\Phi = \forall Y \exists Z\, \phi(X, Y, Z)$, there is a disjunctive logic program $P(\Phi) \subseteq \mathsf{WF}^{ASP}$ of polynomial size with*

$$|SI_{min}(P(\Phi))| = |X| + |\mathsf{Mod}(\Phi)|.$$

Proof. This proof is similar to the one given in Lemma 6.4. Roughly speaking, the main difference is that we use the construction from Theorem 3.1 by Eiter and Gottlob (1995), in order to translate the two quantifiers in the formula.

So let $X = \{x_1, \dots, x_n\}$, $Y = \{y_1, \dots, y_m\}$ and $Z = \{z_1, \dots, z_t\}$. Consider

$$\mathsf{Mod}(\Phi) = \{X \mid \forall Y \exists Z\, \phi(X, Y, Z)\}.$$

We will construct a program $P = P(\Phi)$, where subsets correspond to assignments to the X-variables as in the proof of Lemma 6.4. The subsets that are *not* strongly P-inconsistent will correspond to assignments where

$$\exists Y \forall Z\, \neg\phi(X, Y, Z)$$

holds, and thus, strongly P-inconsistent subsets correspond to assignments in $\mathsf{Mod}(\Phi)$.

We can assume ϕ to be a formula in 3-CNF and thus, $\neg\phi$ is in 3-DNF, i.e., the disjunction of $C_1, \dots, C_r$ with $C_k = l_{k,1} \wedge \ldots \wedge l_{k,3}$. Let $x'_1, \dots, x'_n, y'_1, \dots, y'_m, z'_1, \dots, z'_t$ be fresh atoms and let σ be the mapping

$$\sigma(a) = \begin{cases} a & \text{if } a \in X \cup Y \cup Z, \\ a' & \text{if } a \in \{\neg x_1, \dots, \neg x_n\} \cup \{\neg y_1, \dots, \neg y_m\} \cup \{\neg z_1, \dots, \neg z_t\}. \end{cases}$$

The construction of Eiter and Gottlob (1995) works as follows: Given a formula

$$\Psi = \exists Y \forall Z\, \psi(Y, Z)$$

in 3-DNF (with the notations as here), the following program $\overline{P}$ is consistent if and only if Ψ is valid.

$$
\begin{array}{lll}
\overline{P}: & & \\
y_j \vee y'_j. & & j = 1, ..., m \\
z_l \vee z'_l. & & l = 1, ..., t \\
z_l \leftarrow w^*. & z'_l \leftarrow w^*. & l = 1, ..., t \\
w^* \leftarrow \sigma(l_{k,1}), \sigma(l_{k,2}), \sigma(l_{k,3}). & & k = 1, ..., r \\
w \leftarrow z_1, \ldots, z_t, z'_1, \ldots, z'_t. & & \\
\leftarrow \text{not } w. & &
\end{array}
$$

After consideration of the proof of Lemma 6.4, the following construction should be clear in principle. We give the program and make the few necessary observations afterwards.

$$
\begin{array}{lll}
P: & & \\
x_1., \ldots, x_n. & & \\
x'_1., \ldots, x'_n. & & \\
w \leftarrow \text{not } x_i, \text{not } x'_i, w_{x_1}, ..., w_{x_n}. & & i = 1, ..., n \\
w_{x_i} \leftarrow \text{not } x_i. & w_{x_i} \leftarrow \text{not } x'_i. & i = 1, ..., n \\
y_j \vee y'_j. & & j = 1, ..., m \\
z_l \vee z'_l. & & l = 1, ..., t \\
z_l \leftarrow w^*. & z'_l \leftarrow w^*. & l = 1, ..., t \\
w^* \leftarrow \sigma(l_{k,1}), \sigma(l_{k,2}), \sigma(l_{k,3}). & & k = 1, ..., r \\
w \leftarrow z_1, \ldots z_t, z'_1, \ldots z'_t, w_{x_1}, ..., w_{x_n} & & \\
\leftarrow \text{not } w. & &
\end{array}
$$

Consider a program $H \subseteq \overline{P}$ with $\leftarrow$ not $w. \in H$. Now if H is consistent, then so is $\overline{P}$ because the other rules facilitate entailment of w and can thus never be responsible for inconsistency.

Now the following observations can be made similar as in the proof of Lemma 6.4:

(a) Any inconsistent subset of P contains the constraint "$\leftarrow$ not w." and the inconsistency stems from it.

(b) Let $H \in SI_{min}(P)$. Then, H only contains "$\leftarrow$ not w." and rules of the form "x_i." or "x_i'.".

(c) Let $H \in SI_{min}(P)$. If H does not contain both "x_i." and "x_i'." for an $i \in \{1, ..., n\}$, then it contains either "x_i." or "x_i'." for *all* $i \in \{1, ..., n\}$.

(d) Let $i \in \{1, ..., n\}$ and let $H_i := \{x_i., \ x_i'., \ \leftarrow \text{not } w.\}$. Then, $H_i \in SI_{min}(P)$.

Again, let

$$H_I = \bigcup_{i \in \{1,...,n\}} H_i.$$

As before, $H \in SI_{min}(P) \setminus H_I$ contains either "x_i." or "x_i'." for all $i \in \{1, ..., n\}$. Let $\mathcal{H}_\Omega \subseteq 2^P$ be the set of all subprograms containing either "x_i." or "x_i'." for all $i \in \{1, ..., n\}$. We define a corresponding assignment $\omega(H) : X \to \{0, 1\}$ as before:

$$\omega(H)(x_i) := \begin{cases} 1 & \text{if } x_i. \in H, \\ 0 & \text{if } x_i'. \in H. \end{cases}$$

And the last step is as above.

(e) Let $H \in \mathcal{H}_\Omega \subseteq 2^P$. Then $H \in SI_{min}(P) \setminus H_I$ if and only if $\exists Y \forall Z \, \neg\phi(X, Y, Z)$ does not hold for the assignment $\omega(H)$ to the X-variables.

Hence, any $H \in SI_{min}(P) \setminus H_I$ corresponds to an assignment where

$$\exists Y \forall Z \, \neg\phi(X, Y, Z)$$

is *not* the case, i. e.,

$$|SI_{min}(P)| = |X| + |\{X \mid \forall Y \exists Z \, \phi(X, Y, Z)\}|$$

holds. $\square$

Now, the following Propositions can be inferred from Lemma 6.8 as in the case for disjunction-free logic programs:

Proposition 6.9. *The problems* $\text{Lower}^{L^{ASP}}_{\mathcal{I}_{\mathsf{MSI}}}$ *and* $\text{Upper}^{L^{ASP}}_{\mathcal{I}_{\mathsf{MSI}}}$ *are* $\mathsf{C}\Sigma_2^p$*-complete. The problem* $\text{Exact}^{L^{ASP}}_{\mathcal{I}_{\mathsf{MSI}}}$ *is* $\mathsf{C}_=\Sigma_2^p$*-hard. The problem* $\text{Value}^{L^{ASP}}_{\mathcal{I}_{\mathsf{MSI}}}$ *is* $\#\cdot\Pi_2^p$*-complete under subtractive reductions.*

Proposition 6.14. *The problems* $\textsc{Lower}^{L^{ASP^*}}_{\mathcal{I}_{\mathsf{MSI}^{\mathsf{C}}}}$ *and* $\textsc{Upper}^{L^{ASP^*}}_{\mathcal{I}_{\mathsf{MSI}^{\mathsf{C}}}}$ *are* $\mathsf{C}\Sigma_2^p$-*hard,* $\textsc{Exact}^{L^{ASP^*}}_{\mathcal{I}_{\mathsf{MSI}^{\mathsf{C}}}}$ *is* $\mathsf{C}_=\Sigma_2^p$-*hard.*

Proposition 6.15. *The problem* $\textsc{Lower}^{L^{ASP}}_{\mathcal{I}_{\mathsf{p}}}$ *is* Σ_3^p-*complete,* $\textsc{Upper}^{L^{ASP}}_{\mathcal{I}_{\mathsf{p}}}$ *is* Π_3^p-*complete and* $\textsc{Exact}^{L^{ASP}}_{\mathcal{I}_{\mathsf{p}}}$ *is* D_3^p-*complete.*

Appendix B. Technical Proofs of Section 7

Theorem 7.10. *Let* $\mathcal{K}$ *and* $\mathcal{G}$ *be disjoint knowledge bases. Then* $\mathcal{S}$ *is a minimal hitting set of co-*$\textsc{NRep}_{max}(\mathcal{K},\mathcal{G})$ *if and only if* $\mathcal{S} \in \textsc{Rep}_{min}(\mathcal{K},\mathcal{G})$.

Proof. This is a corollary of the following theorem and Lemma Appendix B.1 below. □

Theorem 7.11. *Let* $\mathcal{K}$ *and* $\mathcal{G}$ *be disjoint knowledge bases. Then* $\mathcal{S}$ *is a minimal hitting set of* $\textsc{Rep}_{min}(\mathcal{K},\mathcal{G})$ *if and only if* $\mathcal{G} \setminus \mathcal{S} \in \textsc{NRep}_{max}(\mathcal{K},\mathcal{G})$.

Proof. "⇒": Let $\mathcal{S}$ be a minimal hitting set of $\textsc{Rep}_{min}(\mathcal{K},\mathcal{G})$. For the sake of contradiction assume that $\mathcal{G} \setminus \mathcal{S} \notin \textsc{NRep}_{max}(\mathcal{K},\mathcal{G})$.

First assume $\mathcal{G} \setminus \mathcal{S} \notin \textsc{NRep}(\mathcal{K},\mathcal{G})$. Then, there is a set $\mathcal{S}'$ with $\mathcal{S} \subseteq \mathcal{S}'$ such that $(\mathcal{K} \cup \mathcal{G}) \setminus \mathcal{S}'$ is consistent. Due to finiteness of $\mathcal{G}$, we might assume $\mathcal{S}'$ is maximal among all subsets of $\mathcal{G}$ that render $(\mathcal{K} \cup \mathcal{G}) \setminus \mathcal{S}'$ consistent. Set $\mathcal{A} = \mathcal{G} \setminus \mathcal{S}'$. Then, $\mathcal{K} \cup \mathcal{A}$ is consistent. In particular, $\mathcal{A} \in \textsc{Rep}_{min}(\mathcal{K},\mathcal{G})$. Due to

$$\mathcal{A} \cap \mathcal{S} \subseteq \mathcal{A} \cap \mathcal{S}' = (\mathcal{G} \setminus \mathcal{S}') \cap \mathcal{S}' = \emptyset$$

we infer $\mathcal{A} \cap \mathcal{S} = \emptyset$. Thus, $\mathcal{S}$ is no hitting set of $\textsc{Rep}_{min}(\mathcal{K},\mathcal{G})$, which is a contradiction.

Now assume $\mathcal{G} \setminus \mathcal{S} \in \textsc{NRep}(\mathcal{K},\mathcal{G})$, but it is not maximal. We thus find a set $\mathcal{S}' \subsetneq \mathcal{S}$ such that $\mathcal{G} \setminus \mathcal{S}' \in \textsc{NRep}(\mathcal{K},\mathcal{G})$. Again due to finiteness we might assume maximality, i.e., $\mathcal{G} \setminus \mathcal{S}' \in \textsc{NRep}_{max}(\mathcal{K},\mathcal{G})$. We claim that $\mathcal{S}'$ is a hitting set of $\textsc{Rep}_{min}(\mathcal{K},\mathcal{G})$ as well, which contradicts minimality of $\mathcal{S}$. This can be seen as follows: Assume $\mathcal{A} \subseteq \mathcal{G}$ and $\mathcal{K} \cup \mathcal{A}$ is consistent and $\mathcal{A}$ minimal, i.e., $\mathcal{A} \in \textsc{Rep}_{min}(\mathcal{K},\mathcal{G})$. In case $\mathcal{A} \cap \mathcal{S}' = \emptyset$ holds, then $\mathcal{A} \subseteq \mathcal{G} \setminus \mathcal{S}'$. In particular, $\mathcal{K} \cup \mathcal{A}$ is consistent with

$$\mathcal{K} \subseteq \mathcal{K} \cup \mathcal{A} \subseteq \mathcal{K} \cup \mathcal{G} \setminus \mathcal{S}',$$

i.e., $\mathcal{K} \notin SI(\mathcal{K} \cup \mathcal{G} \setminus \mathcal{S}')$, which is again a contradiction.

"$\Leftarrow$": Let $\mathcal{G} \setminus \mathcal{S} \in \text{NREP}_{max}(\mathcal{K}, \mathcal{G})$. For the sake of contradiction assume that $\mathcal{S}$ is not a minimal hitting set of $\text{REP}_{min}(\mathcal{K}, \mathcal{G})$.

First assume $\mathcal{S}$ is no hitting set of $\text{REP}_{min}(\mathcal{K}, \mathcal{G})$. Hence, there is an $\mathcal{A} \in \text{REP}_{min}(\mathcal{K}, \mathcal{G})$ with $\mathcal{A} \cap \mathcal{S} = \emptyset$. We infer a contradiction as above. Since $\mathcal{A} \cap \mathcal{S} = \emptyset$ implies $\mathcal{A} \subseteq \mathcal{G} \setminus \mathcal{S}$, and thus $\mathcal{K} \cup \mathcal{A} \subseteq \mathcal{K} \cup \mathcal{G} \setminus \mathcal{S}$, we can find a consistent subset of $\mathcal{K} \cup \mathcal{G} \setminus \mathcal{S}$ which means $\mathcal{G} \setminus \mathcal{S} \notin \text{NREP}(\mathcal{K}, \mathcal{G})$.

Now assume $\mathcal{S}$ is a hitting set of $\text{REP}_{min}(\mathcal{K}, \mathcal{G})$, but not minimal. So let $\mathcal{S}' \subsetneq \mathcal{S}$ be another hitting set of $\text{REP}_{min}(\mathcal{K}, \mathcal{G})$. We claim that this implies $\mathcal{G} \setminus \mathcal{S}' \in \text{NREP}(\mathcal{K}, \mathcal{G})$ contradicting the assumed maximality of $\mathcal{G} \setminus \mathcal{S}$. This can be seen as follows: Assume there is a set $\mathcal{A}$ with $\mathcal{A} \subseteq (\mathcal{G} \setminus \mathcal{S}')$ and $\mathcal{K} \cup \mathcal{A}$ is consistent. Due to finiteness assume minimality of $\mathcal{A}$, i. e., $\mathcal{A} \in \text{REP}_{min}(\mathcal{K}, \mathcal{G})$. Now $\mathcal{A} \subseteq (\mathcal{G} \setminus \mathcal{S}')$ implies $\mathcal{A} \cap \mathcal{S}' = \emptyset$ and in particular, $\mathcal{S}'$ is no hitting set of $\text{REP}_{min}(\mathcal{K}, \mathcal{G})$, which is a contradiction. □

Proof of Theorem 7.10. Consider the following lemma (where $minHS(\mathcal{X})$ is the set of all minimal hitting sets of a set $\mathcal{X}$ of sets):

Lemma Appendix B.1. *(Berge, 1989) Let $\mathcal{X} = \{X_1, \dots, X_n\}$ be a set of sets with $X_i \nsubseteq X_j$ for $i \neq j$. Then $minHS(minHS(\mathcal{X})) = \mathcal{X}$.*

Let us make sure that Lemma Appendix B.1 still applicable, even though we consider hitting sets of *tuples* of sets. There is a simple reason why this is no issue: Since we assume $\mathcal{K} \cap \mathcal{G} = \emptyset$, consideration of tuples is simply for ease of presentation. More precisely, if $\mathcal{A} \subseteq \mathcal{G}$ and $\mathcal{D} \subseteq \mathcal{K}$, then $\mathcal{A}$ and $\mathcal{D}$ are disjoint as well and thus, there is a canonical bijection between the tuples of the form $(\mathcal{D}, \mathcal{A})$ and sets of the form $\mathcal{A} \cup \mathcal{D}$. So if $\mathcal{S} = (\mathcal{S}_\mathcal{A}, \mathcal{S}_\mathcal{D})$ with $\mathcal{S}_\mathcal{A} \subseteq \mathcal{G}$ and $\mathcal{S}_\mathcal{D} \subseteq \mathcal{K}$, then $\mathcal{S} \cap (\mathcal{D}, \mathcal{A}) \neq \emptyset$ iff $\mathcal{S}_\mathcal{A} \cap \mathcal{A} \neq \emptyset$ or $\mathcal{S}_\mathcal{D} \cap \mathcal{D} \neq \emptyset$. Due to $\mathcal{A} \cap \mathcal{D} = \emptyset$ as well as $\mathcal{S}_\mathcal{A} \cap \mathcal{S}_\mathcal{D} = \emptyset$ this is the case if and only if $(\mathcal{A} \cup \mathcal{D}) \cap (\mathcal{S}_\mathcal{A} \cup \mathcal{S}_\mathcal{A}) \neq \emptyset$. However, in the latter term no tuple is mentioned. So we may apply Lemma Appendix B.1 as before.

Now due to Theorem 7.11, $\mathcal{S}$ is a minimal hitting set of $\text{REP}_{min}(\mathcal{K}, \mathcal{G})$ if and only if $\mathcal{S} \in co\text{-NREP}_{max}(\mathcal{K}, \mathcal{G})$. Hence,

$$minHS(\text{REP}_{min}(\mathcal{K}, \mathcal{G})) = co\text{-NREP}_{max}(\mathcal{K}, \mathcal{G})$$

and thus

$$minHS(minHS(\text{REP}_{min}(\mathcal{K}, \mathcal{G}))) = minHS(co\text{-NREP}_{max}(\mathcal{K}, \mathcal{G})).$$

Now we apply Lemma Appendix B.1 to $\textsc{Rep}_{min}(\mathcal{K}, \mathcal{G})$ and obtain:

$$\textsc{Rep}_{min}(\mathcal{K}, \mathcal{G}) = minHS(co\text{-}\textsc{NRep}_{max}(\mathcal{K}, \mathcal{G})).$$

This proves Theorem 7.10. $\square$

Proposition 7.18. *Let $\mathcal{K}$ and $\mathcal{G}$ be disjoint knowledge bases. A tuple of the form $(\mathcal{D}, \emptyset)$ is in* $\textsc{bi-Rep}_{min}(\mathcal{K}, \mathcal{G})$ *if and only if $\mathcal{H} = \mathcal{K} \setminus \mathcal{D} \in C_{max}(\mathcal{K})$.*

Proof. "$\Rightarrow$": If $(\mathcal{D}, \emptyset) \in \textsc{bi-Rep}_{min}(\mathcal{K}, \mathcal{G})$, then $\mathcal{D}$ is minimal such that $\mathcal{K} \setminus \mathcal{D}$ is consistent. So, there is no superset of $\mathcal{H} = \mathcal{K} \setminus \mathcal{D}$ which is consistent. Hence $\mathcal{H} \in C_{max}(\mathcal{K})$.
"$\Leftarrow$": Let $\mathcal{H} = \mathcal{K} \setminus \mathcal{D} \in C_{max}(\mathcal{K})$. Of course, $(\mathcal{D}, \emptyset) \in \textsc{bi-Rep}(\mathcal{K}, \mathcal{G})$ since $\mathcal{K} \setminus \mathcal{D}$ is consistent. Further, there is no set $\mathcal{D}' \subsetneq \mathcal{D}$ such that $\mathcal{K} \setminus \mathcal{D}'$ is consistent. Hence $(\mathcal{D}, \emptyset)$ is necessarily minimal in $\textsc{bi-Rep}(\mathcal{K}, \mathcal{G})$, i. e., $(\mathcal{D}, \emptyset) \in \textsc{bi-Rep}_{min}(\mathcal{K}, \mathcal{G})$. $\square$

Proposition 7.19. *Let $\mathcal{K}$ and $\mathcal{G}$ be disjoint knowledge bases. A tuple of the form $(\emptyset, \mathcal{A})$ is in* $\textsc{bi-Rep}_{min}(\mathcal{K}, \mathcal{G})$ *if and only if $\mathcal{A} \in \textsc{Rep}_{min}(\mathcal{K}, \mathcal{G})$.*

Proof. "$\Rightarrow$": If $(\emptyset, \mathcal{A}) \in \textsc{bi-Rep}_{min}(\mathcal{K}, \mathcal{G})$, then $\mathcal{A}$ is minimal such that $\mathcal{K} \cup \mathcal{A}$ is consistent. Hence $\mathcal{A} \in \textsc{Rep}_{min}(\mathcal{K}, \mathcal{G})$.
"$\Leftarrow$": Let $\mathcal{A} \in \textsc{Rep}_{min}(\mathcal{K}, \mathcal{G})$. Of course, $(\emptyset, \mathcal{A}) \in \textsc{bi-Rep}(\mathcal{K}, \mathcal{G})$ since $\mathcal{K} \cup \mathcal{A}$ is consistent. Further, there is no set $\mathcal{A}' \subsetneq \mathcal{A}$ such that $\mathcal{K} \cup \mathcal{A}'$ is consistent. Hence $(\emptyset, \mathcal{A})$ is necessarily minimal in $\textsc{bi-Rep}(\mathcal{K}, \mathcal{G})$, i. e., $(\emptyset, \mathcal{A}) \in \textsc{bi-Rep}_{min}(\mathcal{K}, \mathcal{G})$. $\square$

Proposition 7.24. *Let $\mathcal{K}$ and $\mathcal{G}$ be disjoint knowledge bases of a monotonic logic. If $(\mathcal{D}, \mathcal{A}) \in \textsc{bi-NRep}_{max}(\mathcal{K}, \mathcal{G})$, then $\mathcal{A} = \mathcal{G}$. Moreover, $(\mathcal{D}, \mathcal{G}) \in \textsc{bi-NRep}_{max}(\mathcal{K}, \mathcal{G})$ if and only if $\mathcal{H} = \mathcal{K} \setminus \mathcal{D} \in SI_{min}(\mathcal{K})$.*

Proof. The first statement is clear. So let us prove the equivalence.
"$\Leftarrow$": Let $\mathcal{H} = \mathcal{K} \setminus \mathcal{D} \in I_{min}(\mathcal{K})$. Due to monotonicity of $\mathcal{K}$, $I_{min}(\mathcal{K}) = SI_{min}(\mathcal{K})$, so $\mathcal{H}$ is strongly $\mathcal{K}$-inconsistent. Moreover, adding formulas from $\mathcal{G}$ cannot render $\mathcal{H}$ consistent. Hence $\mathcal{H}$ is even strongly $(\mathcal{K} \cup \mathcal{G})$-inconsistent, i. e., $(\mathcal{G}, \mathcal{D}) \in \textsc{bi-NRep}_{max}(\mathcal{K}, \mathcal{G})$.
"$\Rightarrow$": Let $(\mathcal{G}, \mathcal{D}) \in \textsc{bi-NRep}_{max}(\mathcal{K}, \mathcal{G})$. Then $\mathcal{K} \setminus \mathcal{D}$ is strongly $(\mathcal{K} \cup \mathcal{G})$-inconsistent. Due to monotonicity, this is equivalent to inconsistency of $\mathcal{K} \setminus \mathcal{D}$. Since $\mathcal{D}$ is maximal st. $\mathcal{K} \setminus \mathcal{D}$ is inconsistent, $\mathcal{H} = \mathcal{K} \setminus \mathcal{D}$ is minimal, i. e., $\mathcal{H} \in I_{min}(\mathcal{K}) = SI_{min}(\mathcal{K})$. $\square$

Proposition 7.25. *Let $\mathcal{K}$ be a knowledge base and $\mathcal{G} = \emptyset$. A tuple of the form $(\mathcal{D}, \emptyset)$ is in* BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$ *if and only if $\mathcal{H} = \mathcal{K} \setminus \mathcal{D} \in SI_{min}(\mathcal{K})$.*

Proof. Let $(\mathcal{D}, \emptyset) \in$ BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$. By definition, $\mathcal{D}$ is maximal such that $\mathcal{K} \setminus \mathcal{D}$ is strongly $\mathcal{K}$-inconsistent. Equivalently, $\mathcal{H} = \mathcal{K} \setminus \mathcal{D}$ is minimal strongly $\mathcal{K}$-inconsistent. □

Proposition 7.26. *Let $\mathcal{K}$ and $\mathcal{G}$ be disjoint knowledge bases.*

- *If $(\mathcal{D}, \mathcal{A}) \in$* BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$*, then $\mathcal{H} = \mathcal{K} \setminus \mathcal{D} \in SI(\mathcal{K})$. In particular, there is a set $\mathcal{D}'$ with $\mathcal{D} \subseteq \mathcal{D}'$ such that $\mathcal{K} \setminus \mathcal{D}' \in SI_{min}(\mathcal{K})$.*

- *If $\mathcal{H} = \mathcal{K} \setminus \mathcal{D} \in SI_{min}(\mathcal{K})$, then there is a (not necessarily uniquely defined) $\mathcal{A} \subseteq \mathcal{G}$ such that $(\mathcal{D}, \mathcal{A}) \in$* BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$*.*

Proof.

- Let $(\mathcal{D}, \mathcal{A}) \in$ BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$. Then $\mathcal{K} \setminus \mathcal{D}$ is strongly $(\mathcal{K} \cup \mathcal{A})$-inconsistent, hence it is also strongly $\mathcal{K}$-inconsistent, so $\mathcal{H} = \mathcal{K} \setminus \mathcal{D} \in SI(\mathcal{K})$.

- Let $\mathcal{H} = \mathcal{K} \setminus \mathcal{D} \in SI_{min}(\mathcal{K})$. In particular, $\mathcal{H} \in SI(\mathcal{K})$. Chose a maximal set $\mathcal{A} \subseteq \mathcal{G}$ such that $\mathcal{H} \in SI(\mathcal{K} \cup \mathcal{A})$. By assumption, such $\mathcal{A}$ exists since $\mathcal{H} \in SI(\mathcal{K} \cup \emptyset)$ (and $\mathcal{G}$ is finite). By definition, $(\mathcal{D}, \mathcal{A}) \in$ BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$. □

Proposition 7.28. *Let $\mathcal{K}$ and $\mathcal{G}$ be disjoint knowledge bases.*

- *If $(\mathcal{D}, \mathcal{A}) \in$* BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$ *then $\mathcal{A} \in$* NREP$(\mathcal{K}, \mathcal{G})$*. In particular, there is a set $\mathcal{A}'$ with $\mathcal{A} \subseteq \mathcal{A}'$ such that $\mathcal{A}' \in$* NREP$_{max}(\mathcal{K})$*.*

- *If $\mathcal{A} \in$* NREP$_{max}(\mathcal{K}, \mathcal{G})$*, then there is a (not necessarily uniquely defined) $\mathcal{D} \subseteq \mathcal{K}$ such that $(\mathcal{D}, \mathcal{A}) \in$* BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$*.*

Proof.

- Let $(\mathcal{D}, \mathcal{A}) \in$ BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$. Then $\mathcal{K} \setminus \mathcal{D}$ is strongly $(\mathcal{K} \cup \mathcal{A})$-inconsistent, hence $\mathcal{K}$ is also strongly $(\mathcal{K} \cup \mathcal{A})$-inconsistent, so $\mathcal{A} \in$ NREP$(\mathcal{K}, \mathcal{G})$.

- Let $\mathcal{A} \in$ NREP$_{max}(\mathcal{K}, \mathcal{G})$. In particular, $\mathcal{K} \in SI(\mathcal{K} \cup \mathcal{A})$ and $\mathcal{A}$ is maximal with this property. Chose a maximal set $\mathcal{D} \subseteq \mathcal{K}$ such that $\mathcal{K} \setminus \mathcal{D} \in SI(\mathcal{K} \cup \mathcal{A})$. By assumption, such $\mathcal{D}$ exists since $\mathcal{K} \setminus \emptyset \in SI(\mathcal{K} \cup \mathcal{A})$, and $(\mathcal{D}, \mathcal{A}) \in$ BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$. □

Before proving Theorem 7.32 we establish the following technical property:

Lemma Appendix B.2. *Let $\mathcal{K}$ and $\mathcal{G}$ be disjoint knowledge bases. Let $\mathcal{A} \subseteq \mathcal{G}$ and $\mathcal{D} \subseteq \mathcal{K}$. Any set $\mathcal{H}$ with $\mathcal{K} \setminus \mathcal{D} \subseteq \mathcal{H} \subseteq \mathcal{K} \cup \mathcal{A}$ can be written as $\mathcal{H} = (\mathcal{K} \setminus \mathcal{D}') \cup \mathcal{A}'$ with $\mathcal{D}' \subseteq \mathcal{D}$ and $\mathcal{A}' \subseteq \mathcal{A}$.*

Proof. Let $\mathcal{K} \setminus \mathcal{D} \subseteq \mathcal{H} \subseteq \mathcal{K} \cup \mathcal{A}$. First observe that $\mathcal{H}$ can be written as

$$\mathcal{H} = (\mathcal{H} \cap \mathcal{K}) \dot{\cup} (\mathcal{H} \cap \mathcal{A})$$

since $\mathcal{H} \subseteq \mathcal{K} \cup \mathcal{A}$ with $\mathcal{K} \cap \mathcal{A} \subseteq \mathcal{K} \cap \mathcal{G} = \emptyset$. Clearly, $\mathcal{H} \cap \mathcal{A} \subseteq \mathcal{A}$ so we may set $\mathcal{A}' = \mathcal{H} \cap \mathcal{A}$. Now set $\mathcal{D}' = \mathcal{K} \setminus (\mathcal{H} \cap \mathcal{K})$. We have

$$\mathcal{K} \setminus \mathcal{D} \subseteq \mathcal{H} \cap \mathcal{K}$$

and thus

$$\mathcal{D} = \mathcal{K} \setminus (\mathcal{K} \setminus \mathcal{D}) \supseteq \mathcal{K} \setminus (\mathcal{H} \cap \mathcal{K}).$$

Moreover,

$$\mathcal{K} \setminus \mathcal{D}' = \mathcal{K} \setminus (\mathcal{K} \setminus (H \cap \mathcal{K})) = \mathcal{H} \cap \mathcal{K},$$

and hence we obtain

$$(\mathcal{K} \setminus \mathcal{D}') \dot{\cup} (A') = (\mathcal{H} \cap \mathcal{K}) \dot{\cup} (\mathcal{H} \cap \mathcal{A})$$

with $\mathcal{A}' \subseteq \mathcal{A}$ and $\mathcal{D}' \subseteq \mathcal{D}$. □

Proposition 7.32. *Let $\mathcal{K}$ and $\mathcal{G}$ be disjoint knowledge bases. Then $\mathcal{S}$ is a minimal hitting set of* BI-REP$_{min}(\mathcal{K}, \mathcal{G})$ *iff $\mathcal{S} \in$ co*-BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$.

Proof. "⇒": Let $\mathcal{S} = (\mathcal{S}_\mathcal{A}, \mathcal{S}_\mathcal{D})$ be a minimal hitting set of BI-REP$_{min}(\mathcal{K}, \mathcal{G})$. For the sake of contradiction assume that $(\mathcal{K} \setminus \mathcal{S}_\mathcal{A}, \mathcal{G} \setminus \mathcal{S}_\mathcal{D}) \notin$ BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$.

First assume $(\mathcal{G} \setminus \mathcal{S}_\mathcal{A}, \mathcal{K} \setminus \mathcal{S}_\mathcal{D}) \notin$ BI-NREP$(\mathcal{K}, \mathcal{G})$. Then, by definition,

$$\mathcal{K} \setminus (\mathcal{K} \setminus \mathcal{S}_\mathcal{D}) \notin SI(\mathcal{K} \cup \mathcal{G} \setminus \mathcal{S}_\mathcal{A}),$$

and thus,

$$\mathcal{S}_\mathcal{D} \notin SI(\mathcal{K} \cup \mathcal{G} \setminus \mathcal{S}_\mathcal{A}).$$

So there is a consistent set $\mathcal{H}$ with $\mathcal{S}_{\mathcal{D}} \subseteq \mathcal{H} \subseteq \mathcal{K} \cup (\mathcal{G} \setminus \mathcal{S}_{\mathcal{A}})$. Due to Lemma Appendix B.2, we find $\mathcal{D} \subseteq \mathcal{K} \setminus \mathcal{S}_{\mathcal{D}}$ and $\mathcal{A} \subseteq \mathcal{G} \setminus \mathcal{S}_{\mathcal{A}}$ with $\mathcal{H} = \mathcal{K} \setminus \mathcal{D} \cup \mathcal{A}$. Due to finiteness of both knowledge bases we might assume $(\mathcal{D}, \mathcal{A}) \in \text{BI-REP}_{min}(\mathcal{K}, \mathcal{G})$. Now $\mathcal{S}_{\mathcal{A}} \cap \mathcal{A} = \emptyset$ as well as $\mathcal{S}_{\mathcal{D}} \cap \mathcal{D} = \emptyset$ implies that $\mathcal{S} = (\mathcal{S}_{\mathcal{A}}, \mathcal{S}_{\mathcal{D}})$ is no hitting set of $\text{BI-REP}_{min}(\mathcal{K}, \mathcal{G})$, which is a contradiction.

Now assume $(\mathcal{G} \setminus \mathcal{S}_{\mathcal{A}}, \mathcal{K} \setminus \mathcal{S}_{\mathcal{D}}) \in \text{NREP}(\mathcal{K}, \mathcal{G})$, but the tuple is not maximal. We thus find a tuple $\mathcal{S}' = (\mathcal{S}_{\mathcal{A}'}, \mathcal{S}_{\mathcal{D}'}) \subseteq (\mathcal{S}_{\mathcal{A}}, \mathcal{S}_{\mathcal{D}}) = \mathcal{S}$ such that $(\mathcal{G} \setminus \mathcal{S}_{\mathcal{A}'}, \mathcal{K} \setminus \mathcal{S}_{\mathcal{D}'}) \in \text{NREP}_{max}(\mathcal{K}, \mathcal{G})$. We claim that $\mathcal{S}'$ is a hitting set of $\text{BI-REP}_{min}(\mathcal{K}, \mathcal{G})$ as well. This can bee seen as follows: Assume this is not the case, i. e., there is $(\mathcal{D}, \mathcal{A}) \in \text{BI-REP}_{min}(\mathcal{K}, \mathcal{G})$ with $\mathcal{S}_{\mathcal{A}'} \cap \mathcal{A} = \emptyset$ as well as $\mathcal{S}_{\mathcal{D}'} \cap \mathcal{D} = \emptyset$. By assumption, $\mathcal{K} \setminus \mathcal{D} \cup \mathcal{A}$ is consistent. Due to $\mathcal{S}_{\mathcal{A}'} \cap \mathcal{A} = \emptyset$ as well as $\mathcal{S}_{\mathcal{D}'} \cap \mathcal{D} = \emptyset$, we obtain $\mathcal{S}_{\mathcal{D}'} \subseteq \mathcal{K} \setminus \mathcal{D}$ and $\mathcal{A} \subseteq \mathcal{G} \setminus \mathcal{S}_{\mathcal{A}'}$, so

$$\mathcal{S}_{\mathcal{D}'} \subseteq (\mathcal{K} \setminus \mathcal{D}) \subseteq (\mathcal{K} \cup \mathcal{A}) \setminus \mathcal{D} \subseteq (\mathcal{K} \cup (\mathcal{G} \setminus \mathcal{S}_{\mathcal{A}'})) \setminus \mathcal{D} \subseteq (\mathcal{K} \cup (\mathcal{G} \setminus \mathcal{S}_{\mathcal{A}'})).$$

In particular,

$$\mathcal{S}_{\mathcal{D}'} \subseteq (\mathcal{K} \cup \mathcal{A}) \setminus \mathcal{D} \subseteq (\mathcal{K} \cup (\mathcal{G} \setminus \mathcal{S}_{\mathcal{A}'})).$$

Due to consistency of $(\mathcal{K} \cup \mathcal{A}) \setminus \mathcal{D}$ we infer that $\mathcal{S}_{\mathcal{D}'} \notin SI(\mathcal{K} \cup (\mathcal{G} \setminus \mathcal{S}_{\mathcal{A}'}))$. So by definition, $(\mathcal{G} \setminus \mathcal{S}_{\mathcal{A}'}, \mathcal{K} \setminus \mathcal{S}_{\mathcal{D}'}) \notin \text{NREP}(\mathcal{K}, \mathcal{G})$ which is a contradiction. Hence, $\mathcal{S}'$ must be a hitting set of $\text{BI-REP}_{min}(\mathcal{K}, \mathcal{G})$ which contradicts minimality of $\mathcal{S}$.

"$\Leftarrow$": Let $(\mathcal{K} \setminus \mathcal{S}_{\mathcal{A}}, \mathcal{G} \setminus \mathcal{S}_{\mathcal{D}}) \in \text{BI-NREP}_{max}(\mathcal{K}, \mathcal{G})$. For the sake of contradiction assume that $\mathcal{S} = (\mathcal{S}_{\mathcal{A}}, \mathcal{S}_{\mathcal{D}})$ is no minimal hitting set of $\text{BI-REP}_{min}(\mathcal{K}, \mathcal{G})$.

First assume that $\mathcal{S}$ is no hitting set of $\text{BI-REP}_{min}(\mathcal{K}, \mathcal{G})$. As above we find a tuple $(\mathcal{D}, \mathcal{A}) \in \text{BI-REP}_{min}(\mathcal{K}, \mathcal{G})$ with $\mathcal{S}_{\mathcal{A}} \cap \mathcal{A} = \emptyset$ as well as $\mathcal{S}_{\mathcal{D}} \cap \mathcal{D} = \emptyset$. Similarly we obtain

$$\mathcal{S}_{\mathcal{D}} \subseteq (\mathcal{K} \cup \mathcal{A}) \setminus \mathcal{D} \subseteq (\mathcal{K} \cup (\mathcal{G} \setminus \mathcal{S}_{\mathcal{A}})).$$

where $(\mathcal{K} \cup \mathcal{A}) \setminus \mathcal{D}$ is consistent. Thus, $(\mathcal{K} \setminus \mathcal{S}_{\mathcal{A}}, \mathcal{G} \setminus \mathcal{S}_{\mathcal{D}}) \notin \text{BI-NREP}(\mathcal{K}, \mathcal{G})$, which is a contradiction.

Now assume that $\mathcal{S}$ is a hitting set of $\text{BI-REP}_{min}(\mathcal{K}, \mathcal{G})$, but not minimal. Let $\mathcal{S}'$ with $\mathcal{S}' = (\mathcal{S}_{\mathcal{A}'}, \mathcal{S}_{\mathcal{D}'}) \subseteq (\mathcal{S}_{\mathcal{A}}, \mathcal{S}_{\mathcal{D}}) = \mathcal{S}$ be a minimal hitting set of $\text{BI-REP}_{min}(\mathcal{K}, \mathcal{G})$. We claim that $(\mathcal{K} \setminus \mathcal{S}_{\mathcal{A}'}, \mathcal{G} \setminus \mathcal{S}_{\mathcal{D}'}) \in \text{BI-NREP}(\mathcal{K}, \mathcal{G})$ contradicting maximality of $(\mathcal{K} \setminus \mathcal{S}_{\mathcal{A}}, \mathcal{G} \setminus \mathcal{S}_{\mathcal{D}})$. For this, assume

$$\mathcal{S}_{\mathcal{D}'} \notin SI(\mathcal{K} \cup \mathcal{G} \setminus \mathcal{S}_{\mathcal{A}'}).$$

Let $\mathcal{H}$ with $\mathcal{S}_{\mathcal{D}'} \subseteq \mathcal{H} \subseteq (\mathcal{K} \cup \mathcal{G}) \setminus \mathcal{S}_{\mathcal{A}'}$ be consistent. As above we apply Lemma Appendix B.2 to find $\mathcal{D} \subseteq \mathcal{K} \setminus \mathcal{S}_{\mathcal{D}'}$ and $\mathcal{A} \subseteq \mathcal{G} \setminus \mathcal{S}_{\mathcal{A}'}$ with $\mathcal{H} = \mathcal{K} \setminus$

$\mathcal{D} \cup \mathcal{A}$. Again due to finiteness we might assume $(\mathcal{D}, \mathcal{A}) \in$ BI-REP$_{min}(\mathcal{K}, \mathcal{G})$. Hence, $\mathcal{S}'$ is no hitting set of BI-REP$_{min}(\mathcal{K}, \mathcal{G})$, which is again a contradiction. $\square$

Now we are almost ready to prove Theorem 7.31. Before we do so, let us make sure that Lemma Appendix B.1 is still applicable, even though we consider hitting sets of *tuples* of sets. There is a simple reason why this is no issue: Since we assume $\mathcal{K} \cap \mathcal{G} = \emptyset$, consideration of tuples is simply for ease of presentation. More precisely, if $\mathcal{A} \subseteq \mathcal{G}$ and $\mathcal{D} \subseteq \mathcal{K}$, then $\mathcal{A}$ and $\mathcal{D}$ are disjoint as well and thus, there is a canonical bijection between the tuples of the form $(\mathcal{D}, \mathcal{A})$ and sets of the form $\mathcal{A} \cup \mathcal{D}$. So if $\mathcal{S} = (\mathcal{S}_{\mathcal{A}}, \mathcal{S}_{\mathcal{D}})$ with $\mathcal{S}_{\mathcal{A}} \subseteq \mathcal{G}$ and $\mathcal{S}_{\mathcal{D}} \subseteq \mathcal{K}$, then $\mathcal{S} \cap (\mathcal{D}, \mathcal{A}) \neq \emptyset$ iff $\mathcal{S}_{\mathcal{A}} \cap \mathcal{A} \neq \emptyset$ or $\mathcal{S}_{\mathcal{D}} \cap \mathcal{D} \neq \emptyset$. Due to $\mathcal{A} \cap \mathcal{D} = \emptyset$ as well as $\mathcal{S}_{\mathcal{A}} \cap \mathcal{S}_{\mathcal{D}} = \emptyset$ this is the case if and only if $(\mathcal{A} \cup \mathcal{D}) \cap (\mathcal{S}_{\mathcal{A}} \cup \mathcal{S}_{\mathcal{A}}) \neq \emptyset$. However, in the latter term no tuple is mentioned. So we may apply Lemma Appendix B.1 as before.

Proof of Theorem 7.31. By Theorem 7.32, $\mathcal{S}$ is a minimal hitting set of BI-REP$_{min}(\mathcal{K}, \mathcal{G})$ if and only if $\mathcal{S} \in$ *co*-BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$. Hence we see as above,

$$minHS(minHS(\text{BI-REP}_{min}(\mathcal{K}, \mathcal{G}))) = minHS(co\text{-BI-NREP}_{max}(\mathcal{K}, \mathcal{G})),$$

which yields

$$\text{BI-REP}_{min}(\mathcal{K}, \mathcal{G}) = minHS(co\text{-BI-NREP}_{max}(\mathcal{K}, \mathcal{G}))$$

after applying Lemma Appendix B.1 as usual. $\square$

Proposition 7.35. *Let $\mathcal{K}$ and $\mathcal{G}$ be disjoint knowledge bases. Let $C_{max}(\mathcal{K}) \neq \emptyset$, i. e., $\mathcal{K}$ possesses consistent subsets and let $SI_{min}(\mathcal{K}) \neq \emptyset$, i. e., $\mathcal{K}$ is inconsistent.. A set $\mathcal{S}_D$ is a minimal hitting set of $SI_{min}(\mathcal{K})$ if and only if $(\mathcal{S}_{\mathcal{D}}, \emptyset)$ is a minimal hitting set of co-*BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$*.*

Proof. "$\Rightarrow$": Let $\mathcal{S}$ be a minimal hitting set of $SI_{min}(\mathcal{K})$. Assume the tuple $(\overline{\mathcal{D}}, \overline{\mathcal{A}})$ is in *co*-BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$. Then there is a tuple $(\mathcal{D}, \mathcal{A}) \in$ BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$ with $\mathcal{A} = \mathcal{G} \setminus \overline{\mathcal{A}}$ and $\mathcal{D} = \mathcal{K} \setminus \overline{\mathcal{D}}$. Due to Proposition 7.26, $\mathcal{K} \setminus \mathcal{D} \in SI(\mathcal{K})$, i. e., $\overline{\mathcal{D}} \in SI(\mathcal{K})$. Due to finiteness, there is a set $\overline{\mathcal{D}}' \in SI_{min}(\mathcal{K})$ with $\overline{\mathcal{D}}' \subseteq \overline{\mathcal{D}}$. Since $\mathcal{S}_{\mathcal{D}}$ is a minimal hitting set of $SI_{min}(\mathcal{K})$, we conclude $\emptyset \neq \mathcal{S}_{\mathcal{D}} \cap \overline{\mathcal{D}}' \subseteq \mathcal{S}_{\mathcal{D}} \cap \overline{\mathcal{D}}$. Since $(\overline{\mathcal{D}}, \overline{\mathcal{A}})$ was an arbitrary tuple in *co*-BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$ we see that $(\mathcal{S}_{\mathcal{D}}, \emptyset)$ is a hitting set of *co*-BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$.

We have left to prove minimality of $(\mathcal{S}_\mathcal{D}, \emptyset)$. Again due to Proposition 7.26, for any $\mathcal{K} \setminus \mathcal{D} \in SI_{min}(\mathcal{K})$, there is a tuple of the form $(\mathcal{D}, \mathcal{A}) \in$ $\textsc{bi-Nrep}_{max}(\mathcal{K}, \mathcal{G})$ and hence a tuple of the form $(\mathcal{K}\setminus\mathcal{D}, \mathcal{G}\setminus\mathcal{A}) \in co\text{-}\textsc{bi-Nrep}_{max}(\mathcal{K}, \mathcal{G})$. So if $(\mathcal{S}_\mathcal{D}, \emptyset)$ is a hitting set of $co\text{-}\textsc{bi-Nrep}_{max}(\mathcal{K}, \mathcal{G})$, then $\mathcal{S}_\mathcal{D}$ is a hitting set of $SI_{min}(\mathcal{K})$. Since $\mathcal{S}_\mathcal{D}$ is minimal (as a hitting set of $SI_{min}(\mathcal{K})$), we conclude that $(\mathcal{S}_\mathcal{D}, \emptyset)$ is minimal (as a hitting set of $co\text{-}\textsc{bi-Nrep}_{max}(\mathcal{K}, \mathcal{G})$).

"$\Leftarrow$": If $\mathcal{H} = \mathcal{K}\setminus\mathcal{D} \in SI_{min}(\mathcal{K})$, then there is a tuple $(\mathcal{D}, \mathcal{A}) \in \textsc{bi-Nrep}_{max}(\mathcal{K}, \mathcal{G})$ due to Proposition 7.26. In particular, there is a tuple $(\mathcal{K} \setminus \mathcal{D}, \mathcal{G} \setminus \mathcal{A}) \in$ $co\text{-}\textsc{bi-Nrep}_{max}(\mathcal{K}, \mathcal{G})$. So if $(\mathcal{S}_\mathcal{D}, \emptyset)$ is a hitting set of $co\text{-}\textsc{bi-Nrep}_{max}(\mathcal{K}, \mathcal{G})$, then $\mathcal{S}_\mathcal{D}\mathcal{S}_\mathcal{D} \cap \mathcal{K} \setminus \mathcal{D} \neq \emptyset$. Hence $\mathcal{S}_\mathcal{D}$ is a hitting set of $SI_{min}(\mathcal{K})$.

Minimality is a consequence of "$\Rightarrow$": Assume $(\mathcal{S}_\mathcal{D}, \emptyset)$ is a minimal hitting set of $co\text{-}\textsc{bi-Nrep}_{max}(\mathcal{K}, \mathcal{G})$. If there is a hitting set $\mathcal{S}'_\mathcal{D} \subsetneq \mathcal{S}_\mathcal{D}$ of $SI_{min}(\mathcal{K})$, then $(\mathcal{S}'_\mathcal{D}, \emptyset)$ is also a hitting set of $co\text{-}\textsc{bi-Nrep}_{max}(\mathcal{K}, \mathcal{G})$, contradicting minimality of $(\mathcal{S}_\mathcal{D}, \emptyset)$. □

Proposition 7.36. *Let $\mathcal{K}$ and $\mathcal{G}$ be disjoint knowledge bases. Let $C_{max}(\mathcal{K}) \neq \emptyset$, i. e., $\mathcal{K}$ possesses consistent subsets and let $SI_{min}(\mathcal{K}) \neq \emptyset$, i. e., $\mathcal{K}$ is inconsistent.. Then there is no tuple $(\overline{\mathcal{D}}, \overline{\mathcal{A}}) \in co\text{-}\textsc{bi-Nrep}_{max}(\mathcal{K}, \mathcal{G})$ with $\overline{\mathcal{D}} = \emptyset$.*

Proof. Although this is a corollary of Proposition 7.35, we want to give a straightforward proof which directly illustrates why we require $\mathcal{K}$ to possess consistent subsets:

For any tuple $(\mathcal{D}, \mathcal{A}) \in \textsc{bi-Nrep}_{max}(\mathcal{K}, \mathcal{G})$ we have $\mathcal{D} \subsetneq \mathcal{K}$. Otherwise, $\mathcal{K} \setminus \mathcal{K} = \emptyset$ would be strongly $(\mathcal{K} \cup \mathcal{A})$-inconsistent. However, as $\mathcal{K}$ possesses consistent subsets, this is a contradiction. Hence, whenever $(\overline{\mathcal{D}}, \overline{\mathcal{A}}) \in co\text{-}\textsc{bi-Nrep}_{max}(\mathcal{K}, \mathcal{G})$, we can conclude $\overline{\mathcal{D}} \neq \emptyset$. □

Proposition 7.37. *Let $\mathcal{K}$ and $\mathcal{G}$ be disjoint knowledge bases. Let $\textsc{Rep}_{min}(\mathcal{K}, \mathcal{G}) \neq \emptyset$, i. e., $\mathcal{K}$ possesses addition-based repairs and let $SI_{min}(\mathcal{K}) \neq \emptyset$, i. e., $\mathcal{K}$ is inconsistent.. A set $\mathcal{S}_\mathcal{A}$ is a minimal hitting set of $co\text{-}\textsc{Nrep}_{max}(\mathcal{K}, \mathcal{G})$ if and only if $(\emptyset, \mathcal{S}_\mathcal{A})$ is a minimal hitting set of $co\text{-}\textsc{bi-Nrep}_{max}(\mathcal{K}, \mathcal{G})$.*

Proof. "$\Rightarrow$": Let $\mathcal{S}_\mathcal{A}$ be a minimal hitting set of $co\text{-}\textsc{Nrep}_{max}(\mathcal{K}, \mathcal{G})$. Assume the tuple $(\overline{\mathcal{D}}, \overline{\mathcal{A}})$ is in $co\text{-}\textsc{bi-Nrep}_{max}(\mathcal{K}, \mathcal{G})$. Then there is a tuple $(\mathcal{D}, \mathcal{A}) \in$ $\textsc{bi-Nrep}_{max}(\mathcal{K}, \mathcal{G})$ with $\mathcal{A} = \mathcal{G} \setminus \overline{\mathcal{A}}$ and $\mathcal{D} = \mathcal{K} \setminus \overline{\mathcal{D}}$. Due to Proposition 7.28, $\mathcal{K} \cup \mathcal{A} \in \textsc{Nrep}(\mathcal{K}, \mathcal{G})$. So there is a set $\mathcal{A}'$ with $\mathcal{A} \subseteq \mathcal{A}' \in \textsc{Nrep}_{max}(\mathcal{K}, \mathcal{G})$. Since $\mathcal{S}_\mathcal{A}$ is a minimal hitting set of $co\text{-}\textsc{Nrep}_{max}(\mathcal{K}, \mathcal{G})$, we conclude

$$\emptyset \neq \mathcal{S}_\mathcal{A} \cap (\mathcal{G} \setminus \mathcal{A}') \subseteq \mathcal{S}_\mathcal{A} \cap (\mathcal{G} \setminus \mathcal{A}) = \mathcal{S}_\mathcal{A} \cap (\overline{\mathcal{A}}).$$

Since $(\overline{\mathcal{D}}, \overline{\mathcal{A}})$ was an arbitrary tuple in *co*-BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$ we see that $(\emptyset, \mathcal{S}_{\mathcal{A}})$ is a hitting set of *co*-BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$.

As before, we have left to prove minimality of $(\emptyset, \mathcal{S}_{\mathcal{A}})$. Again due to Proposition 7.28, for any $\mathcal{A} \in$ NREP$_{max}(\mathcal{K}, \mathcal{G})$, there is a tuple $(\mathcal{D}, \mathcal{A}) \in$ BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$. So if $(\emptyset, \mathcal{S}_{\mathcal{A}})$ is a hitting set of *co*-BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$, then $\mathcal{S}_{\mathcal{A}}$ is a hitting set of *co*-NREP$_{max}(\mathcal{K}, \mathcal{G})$. Since $\mathcal{S}_{\mathcal{A}}$ is minimal (as a hitting set of *co*-NREP$_{max}(\mathcal{K}, \mathcal{G})$), we conclude that $(\emptyset, \mathcal{S}_{\mathcal{A}})$ is minimal (as a hitting set of *co*-BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$) as well.

"$\Leftarrow$": If $\mathcal{A} \in$ NREP$_{max}(\mathcal{K}, \mathcal{G})$, then there is a tuple $(\mathcal{D}, \mathcal{A}) \in$ BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$ due to Proposition 7.28. So, if $(\emptyset, \mathcal{S}_{\mathcal{A}})$ is a hitting set of *co*-BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$, then $\mathcal{S}_{\mathcal{A}}$ is a hitting set of *co*-NREP$_{max}(\mathcal{K}, \mathcal{G})$ as well.

Minimality is a consequence of "$\Rightarrow$" as in the proof of Proposition 7.35. Assume $(\emptyset, \mathcal{S}_{\mathcal{A}})$ is a minimal hitting set of *co*-BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$. If there is a set $\mathcal{S}'_{\mathcal{A}} \subsetneq \mathcal{S}_{\mathcal{A}}$ that is a hitting set of NREP$_{max}(\mathcal{K}, \mathcal{G})$, then $(\emptyset, \mathcal{S}'_{\mathcal{A}})$ is a hitting set of *co*-BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$ as well, contradicting minimality of $(\emptyset, \mathcal{S}_{\mathcal{A}})$. $\square$

Proposition 7.38. *Let $\mathcal{K}$ and $\mathcal{G}$ be disjoint knowledge bases. Let* REP$_{min}(\mathcal{K}, \mathcal{G}) \neq \emptyset$*, i. e., $\mathcal{G}$ possesses repairing subsets wrt. $\mathcal{K}$ and let $SI_{min}(\mathcal{K}) \neq \emptyset$, i. e., $\mathcal{K}$ is inconsistent.. Then there is no $(\overline{\mathcal{D}}, \overline{\mathcal{A}}) \in$ co-*BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$ *with $\overline{\mathcal{A}} = \emptyset$.*

Proof. For any tuple $(\mathcal{D}, \mathcal{A}) \in$ BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$ it holds that $\mathcal{A} \subsetneq \mathcal{G}$. Otherwise, $\mathcal{K}$ would be strongly $(\mathcal{K} \cup \mathcal{G})$-inconsistent. However, this contradicts REP$_{min}(\mathcal{K}, \mathcal{G}) \neq \emptyset$. So, whenever $(\overline{\mathcal{D}}, \overline{\mathcal{A}}) \in$ *co*-BI-NREP$_{max}(\mathcal{K}, \mathcal{G})$ holds, we conclude $\overline{\mathcal{A}} \neq \emptyset$. $\square$