On Quasi-Inconsistency and its Complexity

Carl Corea, Matthias Thimm

University of Koblenz-Landau

Abstract

We address the issue of analyzing *potential* inconsistencies in knowledge bases. This refers to knowledge bases that contain rules which will always be activated together, and the knowledge base will become inconsistent, should these rules be activated. We investigate this problem in the context of the industrial use-case of business rule management, where it is often required that sets of (only) rules are analyzed for potential inconsistencies, e. g., during business rule modelling. To this aim, we introduce the notion of quasi-inconsistency, which is a formalization of the above-mentioned problem of potential inconsistencies. We put a specific focus on the analysis of computational complexity of some involved problems and show that many of them are intractable.

1. Introduction

Inconsistency is a core problem in knowledge representation and reasoning, and usually refers to a knowledge base containing multiple pieces of information which cannot hold at the same time. For example, consider the following logic program K_1 (we will formalize syntax and semantics later)

$$K_1 = \{a; \quad b \leftarrow a; \quad \neg b \leftarrow a\}.$$

K_1 is inconsistent in the classic-logical sense, as the conclusions $b, \neg b$ cannot hold at the same time.

Handling such inconsistencies is not only subject of research within the wider area of Knowledge Representation and Reasoning (KR), but also a problem often faced in practice, for example, in the field of *Business Rule Management* (BRM), cf. [1, 2] for an overview or [3, 4, 5] for some recent

[1]{ccorea,thimm}@uni-koblenz.de

works. Informally speaking, business rules are declarative statements about a domain of interest, which are used to model (external) regulations in order to govern compliant company activities. Inconsistencies in business rule bases are a challenge within BRM, as they impede using business rules for their intended purpose of reasoning about allowed company behavior [6].

Consider the exemplary rule base $\mathcal{B}_1$ defined via

$$\mathcal{B}_1 = \{\mathit{creditworthy} \leftarrow \mathit{newCustomer};\ \ \neg\mathit{creditworthy} \leftarrow \mathit{newCustomer}\}.$$

In contrast to $\mathcal{K}_1$, we see that this business rule base contains only rules, but no facts (such as a in $\mathcal{K}_1$). This is very common in a BRM setting, as business rules are modelled at design-time (and case-dependent facts will only be known during process execution). Yet, the company has to warrant a "consistent“ rule-base during modelling. In this regard, we see that $\mathcal{B}_1$ is *consistent* in the classic-logical sense, as no non-trivial inference is possible without facts. Existing approaches to inconsistency-tolerant reasoning [7] and inconsistency measurement [8] would therefore not indicate any issue in this case. Yet, from a business rule management perspective, it does not make sense to have two rules which will a) always be activated together, but b) are contradictory, should they be activated. Intuitively, the rules in $\mathcal{B}_1$ would become useless for compliance reasoning, given we would encounter a fact *newCustomer* during run-time. Thus, methods are needed to analyze such *potential* inconsistencies in knowledge bases. This could be useful for companies, in order to handle modelling errors in the business logic and facilitate correct compliance reasoning.

In order to address the scenario discussed above, we introduce the notion of *quasi-inconsistency*. The intuition is that a rule base, i. e., a knowledge base containing only rules, is quasi-inconsistent, if there are rules that will always be activated together but yield inconsistent conclusions. This notion is a generalisation of the notion of *incoherence* from description logics and similar formalisms [9, 10, 11]. In this work, however, we consider a general rule-based knowledge representation formalism in order to theoretically investigate this notion in a broad manner (Section 2). We then define the notion of quasi-inconsistency (Section 3) and examine the complexity of several computational tasks involving quasi-inconsistency (Section 4). We conclude with a discussion in Section 5 where we also point out the relationship to the notion of incoherence.

2. Preliminaries

We consider a general but simple monotonic rule-based knowledge representation formalism, similar to logic programs with classical negation but without default negation [12]. Let $\mathcal{A}$ be a set of propositional atoms and $\mathcal{L}$ the corresponding set of literals, i. e., $\mathcal{L} = \{a, \neg a \mid a \in \mathcal{A}\}$ where $\neg$ is interpreted as classical negation. We abbreviate $\overline{\neg a} = a$ and $\overline{a} = \neg a$ for an atom a.

A rule r has the form

$$r: \quad l_0 \leftarrow l_1, \ldots, l_m. \tag{1}$$

with $l_0, \ldots, l_m \in \mathcal{L}$ and $m > 1$ (note that we require rules to have a non-empty premise on purpose). We abbreviate $head(r) = l_0$ and $body(r) = \{l_1, \ldots, l_m\}$. Let $\mathcal{R}_\mathcal{L}$ denote the set of all rules. A *rule base* $\mathcal{B}$ is a set of rules, i. e., $\mathcal{B} \subseteq \mathcal{R}_\mathcal{L}$, and a set of facts (=literals) $F \subseteq \mathcal{L}$ is also called *fact base*. For X being a rule base or fact base, let $\mathcal{A}(X)$ denote the set of atoms appearing in X.

An important fragment of $\mathcal{R}_\mathcal{L}$ is the set of *acyclic* rules bases, i. e., rule bases without any cycles in the rules. More concretely, the *dependency graph* $G_\mathcal{B}$ of a rule base $\mathcal{B}$ is a directed graph $G_\mathcal{B} = (\mathcal{B}, E_\mathcal{B})$ where $(r_1, r_2) \in E_\mathcal{B}$ for $r_1, r_2 \in \mathcal{B}$ iff $head(r_1) \in body(r_2)$. Let $\mathcal{R}_\mathcal{L}^{\text{acyclic}} \subseteq \mathcal{R}_\mathcal{L}$ be the set of those rule bases $\mathcal{B} \in \mathcal{R}_\mathcal{L}$ where $G_\mathcal{B}$ is acyclic. Computationally, rule bases in $\mathcal{R}^{\text{acyclic}}$ are easier to analyse than general rule bases, as we will see in the remainder of the paper. However, we will also consider the case of rule bases that may indeed contain cycles.

Example 1. *Consider the rule base $\mathcal{B}_2$ defined via*

$$\mathcal{B}_2 = \{b \leftarrow a; \quad c \leftarrow b; \quad d \leftarrow c\}.$$

$\mathcal{B}_2$ is acyclic, where $G_{\mathcal{B}_2} = (\mathcal{B}_2, E_{\mathcal{B}_2})$ with

$$\begin{aligned} E_{\mathcal{B}_2} = \{&(b \leftarrow a; \quad c \leftarrow b), \\ &(c \leftarrow b; \quad d \leftarrow c)\} \end{aligned}$$

A set $M \subseteq \mathcal{L}$ of literals is *closed* wrt. a rule base $\mathcal{B}$ and a fact base F, iff $F \subseteq M$ and for every rule of the form in (1), if $l_1, \ldots, l_m \in M$ then $l_0 \in M$. The *F-minimal model* M of a rule base $\mathcal{B}$, denoted by $\min_F(\mathcal{B})$ is the smallest (wrt. set inclusion) closed set of literals. A set M of literals is called *consistent* if it does not contain both a and $\neg a$ for an atom a. A rule base $\mathcal{B}$ is called *F-consistent* if its F-minimal model is consistent. If not, we call $\mathcal{B}$ F-inconsistent. Some obvious properties of F-consistency are summarised in the following result (given without proof).

Proposition 1. *Let $\mathcal{B}$ be a rule base and F a set of facts.*

1. *$\mathcal{B}$ is $\emptyset$-consistent.*
2. *$\mathcal{B}$ is $\mathcal{L}$-inconsistent.*
3. *If F is inconsistent then $\mathcal{B}$ is F-inconsistent.*
4. *If $\mathcal{B}$ is F-inconsistent then $\mathcal{B}$ is F'-inconsistent for every $F \subseteq F'$.*
5. *If $\mathcal{B}$ is F-consistent then $\mathcal{B}$ is F'-consistent for every $F' \subseteq F$.*

Due to property 3.) and 4.) from above we strengthen the notion of F-inconsistency as follows. We say that a rule base $\mathcal{B}$ is *minimally F-inconsistent* if 1.) F is consistent, 2.) $\mathcal{B}$ is *F-inconsistent*, and 3.) for every $F' \subsetneq F$, $\mathcal{B}$ is *F'-consistent.*

Example 2. *Consider the rule base $\mathcal{B}_3$ and fact base F_1, defined via*

$$\mathcal{B}_3 = \{b \leftarrow a; \quad c \leftarrow b; \quad \neg c \leftarrow a\}$$
$$F_1 = \{a\}$$

Then we have F_1 is a consistent set of facts, $\mathcal{B}_3$ is $\{a\}$-inconsistent, and $\mathcal{B}_3$ is $\emptyset$-consistent, thus, $\mathcal{B}_3$ is minimally $\{a\}$-inconsistent.

3. Quasi-Inconsistency

When considering knowledge bases consisting only of rules, we see from Proposition 1 that every rule base is consistent in the classic-logical sense, i. e., $\emptyset$-consistent. In this work, we are, however, interested in cases where there exists a set of facts F, s.t. a rule base becomes F-inconsistent. That is, we are interested in cases where a rule base will become inconsistent, should certain facts be introduced. In industrial application scenarios such as business rule management, addressing this problem is of high interest, as the modelling of business rules happens at an earlier point in time, independent of facts, i. e., it cannot be foreseen which combination of (case-dependent) facts will occur "later" during run-time. Here, it becomes important for companies to analyze whether there exist *potential* inconsistencies in the rule base, such that experts can improve operations and counteract potential compliance breaches during process execution.

Example 3. *Consider the rule base $\mathcal{B}_4$ defined via*

$$\mathcal{B}_4 = \{e \leftarrow a; \quad \neg e \leftarrow c; \quad c \leftarrow a; \quad e \leftarrow b\}$$

Observe that $\mathcal{B}_4$ is minimally $\{a\}$-inconsistent and minimally $\{b,c\}$-inconsistent. However, the $\{a\}$-, resp. $\{b,c\}$-inconsistency entail different types of problems:

The $\{b,c\}$-inconsistency says that whenever b and c are added to the rule base, we activate both $\neg e \leftarrow c$ and $e \leftarrow b$, which yields an inconsistent conclusion. However, there can of course be cases where only b, or only c occurs, thus it is possible to activate the rules individually and draw meaningful conclusions from the rule base. It may even be the case that b and c will never appear together during run-time, because there is some extrinsic constraint not allowing the two to appear together (for example b could be "The client is under-age" and c could be "The client is older than 60 years"; although these two statements are not direct complements of each other, they cannot be true at the same time). We therefore denote the $\{b,c\}$-inconsistency as a potential issue *and we do not consider dealing with these in this paper.*

On the other hand, the $\{a\}$-inconsistency says that whenever the rule $e \leftarrow a$ is activated, we automatically derive both e and $\neg e$ (because of the rules $c \leftarrow a$ and $\neg e \leftarrow c$). To clarify, the rules $e \leftarrow a$ and $c \leftarrow a$ can either not be activated at all, or, they will always be activated together but in this case yield inconsistent conclusions. Thus, these two rules cannot be used for any meaningful reasoning. We therefore denote the $\{a\}$-inconsistency as an (actual) issue.

Intuitively, it is important to resolve *(actual) issues* in the scope of business rule management, as they clearly indicate a modelling error in the set of business rules.For that purpose, we will now introduce the notion of *quasi-inconsistency*. Informally, we say that a rule base is *quasi-inconsistent* if it contains rules that will always be activated together and yield inconsistent conclusions. For that we need some further notation.

Definition 1 (Rule Set activation)**.** *A set of facts X* activates *a finite set of rules R iff there is a sequence $< r_1, \ldots, r_n >$ with $\{r_1, \ldots, r_n\} = R$ such that*

1. *$body(r_1) \subseteq X$*
2. *for all $i = 2, \ldots, n$ we have $body(r_i) \subseteq \{head(r_1), \ldots, head(r_{i-1})\} \cup X$*

A set of facts X minimally (w.r.t. set-inclusion) activates *a set of rules R iff X activates R and there is no proper subset of X that activates R. If X (minimally) activates R we also say that X is a* (minimal) activation set *of R. Let $\mathsf{ActSets}(R)$ be the set of minimal activation sets of R.*

Intuitively, X is a set of facts sufficient for deriving all conclusions of rules in R. We summarise some obvious properties of activation sets in the following result (given without proof).

Proposition 2. *Let $\mathcal{B}$ be a rule base and X, X' sets of facts.*

1. *If X activates $\mathcal{B}$ and $X \subseteq X'$ then X' activates $\mathcal{B}$.*
2. *If $\mathcal{B} \in \mathcal{R}_{\mathcal{L}}^{acyclic}$ then $\mathsf{ActSets}(\mathcal{B}) = \{X_{\mathcal{B}}\}$ with*

$$X_{\mathcal{B}} = \bigcup_{r \in \mathcal{B}} body(r) \setminus \bigcup_{r \in \mathcal{B}} \{head(r)\}$$

In particular, note that 2.) means that acyclic rule bases have uniquely determined activation sets that have a simple characterisation (and can be computed in polynomial time).

Example 4. *We recall the rule base $\mathcal{B}_2$*

$$\mathcal{B}_2 = \{b \leftarrow a; \quad c \leftarrow b; \quad d \leftarrow c\}.$$

For each individual rule, its activation set consists simply of the body of the rule, i. e., $\{a\}$ is an activation set of $\{b \leftarrow a\}$. Furthermore, the set $\{a\}$ also activates the entire set $\mathcal{B}_2$.

Observe that a cyclic rule base may have multiple (minimal) activation sets.

Example 5. *Consider the rule base $\mathcal{B}_c = \{a \leftarrow b; \quad b \leftarrow a\}$. Then both $\{a\}$ and $\{b\}$ minimally activate $\mathcal{B}_c$.*

In general, $\mathsf{ActSets}(R)$ may become exponential in size. Consider, e. g., $R_i = \{a_1 \leftarrow b_1; \quad b_1 \leftarrow a_1; \quad \ldots; \quad a_i \leftarrow b_i; \quad b_i \leftarrow a_i\}$ with $|\mathsf{ActSets}(R_i)| = 2^i$ for $i > 0$.

We are now ready to define *quasi-inconsistency* as follows.

Definition 2 (Quasi-Inconsistency)**.** *Let $R_1, R_2 \subseteq \mathcal{R}_{\mathcal{L}}$ be rule bases and X_1, X_2 be consistent sets of literals. A tuple (R_1, X_1, R_2, X_2) is called an* issue *iff*

1. *$X_1 \subseteq X_2$.*
2. *X_1 minimally activates R_1.*
3. *X_2 minimally activates R_2.*
4. *R_1 is X_1-consistent and R_2 is X_2-consistent.*
5. *$R_1 \cup R_2$ is X_2-inconsistent.*

A tuple (R_1, X_1, R_2, X_2) *is called a* minimal issue *iff there are no* R'_1, R'_2, X'_1, X'_2 *with* $R'_1 \subseteq R_1$ *and* $R'_2 \subseteq R_2$ *(one of these set inclusions being proper) such that* (R'_1, X'_1, R'_2, X'_2) *is an issue.*

A rule base $\mathcal{B}$ *is* quasi-inconsistent *iff there is an issue* (R_1, X_1, R_2, X_2) *with* $R_1, R_2 \subseteq \mathcal{B}$*. Then we also say that* (R_1, X_1, R_2, X_2) *is an issue of* $\mathcal{B}$*. Let* $\mathsf{Issues}(\mathcal{B})$*,* $\mathsf{MinIssues}(\mathcal{B})$ *be the set of all (minimal) issues of* $\mathcal{B}$*, respectively.*

In other words, an issue (R_1, X_1, R_2, X_2) describes a case where the activation of one set of rules R_1 implies the activation of a second set of rules R_2 and both sets together derive an inconsistency (while being consistent on their own). Note that for acyclic rule bases we can write an issue (R_1, X_1, R_2, X_2) simply as (R_1, R_2) as the minimal activation sets X_1 and X_2 are uniquely determined, cf. Proposition 2.

Example 6. *Consider the following rule bases* $\mathcal{B}_5$*–*$\mathcal{B}_7$*, defined via*

$$\begin{aligned}
\mathcal{B}_5 &= \{c \leftarrow a; \quad \neg c \leftarrow a, b\} \\
\mathcal{B}_6 &= \{c \leftarrow a, b; \quad \neg c \leftarrow a, d; \quad d \leftarrow b\} \\
\mathcal{B}_7 &= \{c \leftarrow a, f; \quad \neg c \leftarrow h, d; \quad d \leftarrow b; \quad f \leftarrow b; \quad h \leftarrow a\}
\end{aligned}$$

Then for

$$\begin{aligned}
t_1 &= (\{c \leftarrow a\}, \{a\}, \{\neg c \leftarrow a, b\}, \{a, b\}) \\
t_2 &= (\{c \leftarrow a, b\}, \{a, b\}, \{d \leftarrow b; \neg c \leftarrow a, d\}, \{a, b\}) \\
t_3 &= (\{f \leftarrow b; \quad c \leftarrow a, f\}, \{a, b\}, \\
&\qquad \{d \leftarrow b; \quad h \leftarrow a; \quad \neg c \leftarrow h, d\}, \{a, b\})
\end{aligned}$$

we have that t_1 *is an issue of* $\mathcal{B}_5$*,* t_2 *is an issue of* $\mathcal{B}_6$*, and* t_3 *is an issue of* $\mathcal{B}_7$*. Hence, all these rule bases are quasi-inconsistent (while being classically consistent).*

Example 7. *We recall the business rule base* $\mathcal{B}_1$ *from the introduction*

$$\begin{aligned}
\mathcal{B}_1 =& \{\mathit{creditworthy} \leftarrow \mathit{newCustomer}; \\
& \neg \mathit{creditworthy} \leftarrow \mathit{newCustomer}\}.
\end{aligned}$$

Then we have that

$$\begin{aligned}
t_4 =& (\{\mathit{creditworthy} \leftarrow \mathit{newCustomer}\}, \{\mathit{newCustomer}\}, \\
& \{\neg \mathit{creditworthy} \leftarrow \mathit{newCustomer}\}, \{\mathit{newCustomer}\})
\end{aligned}$$

is a minimal issue of $\mathcal{B}_1$*, i. e.,* $\mathcal{B}_1$ *is quasi-inconsistent (despite being classically consistent).*

We conclude this section with a result summarising some general properties of quasi-inconsistency (proofs are straightforward and omitted).

Proposition 3. *Let $\mathcal{B}_1, \mathcal{B}_2$ be rule bases.*

1. *If $\mathcal{B}_1$ is quasi-inconsistent and $\mathcal{B}_1 \subseteq \mathcal{B}_2$ then $\mathcal{B}_2$ is quasi-inconsistent.*
2. *If $\mathcal{B}_1 \subseteq \mathcal{B}_2$ then $\mathsf{Issues}(\mathcal{B}_1) \subseteq \mathsf{Issues}(\mathcal{B}_2)$.*
3. *If $\mathcal{B}_1 \subseteq \mathcal{B}_2$ then $\mathsf{MinIssues}(\mathcal{B}_1) \subseteq \mathsf{MinIssues}(\mathcal{B}_2)$.*

The means to detect quasi-inconsistency proposed in this work heavily rely on the notion of (minimal) issues. Especially as we envisage to apply our results to support an industrial use-case, the actual computation of issues needs to be addressed. In the following, we therefore investigate the computational complexity of various problems related to quasi-inconsistency.

4. Computational complexity

We assume familiarity with basic concepts of computational complexity and basic complexity classes such as P and NP, see [13] for an introduction.

We start with analysing the complexity of verification tasks pertaining to issues.

Lemma 1. *Let $\mathcal{B}$ be a rule base, $R_1, R_2 \subseteq \mathcal{B}$, and X_1, X_2 consistent sets of literals. Checking whether (R_1, X_1, R_2, X_2) is an issue can be done in polynomial time.*

Proof. We go through the properties of an issues step by step (compare with Definition 2):

1. Checking $X_1 \subseteq X_2$ is obviously polynomial.
2. Checking that X_1 activates R_1 is simple forward propagation (check every rule whether it can be activated with X_1 alone; if yes add the head of that rule to X_1 and continue). In order to check that X_1 minimally activates R_1 it suffices to check whether $X_1' \subseteq X_1$ where X_1' has exactly one fact less than X_1 does not activate R_1 (for all such X_1', which are exactly $|X_1|$ many)
3. Checking that X_2 activates R_2 is analogous.
4. Checking $X_1 \cup R_1 \not\models \perp$ is simple forward propagation and checking whether the set of derived literals is consistent; analogous for $X_2 \cup R_2 \not\models \perp$.
5. Checking $X_1 \cup R_1 \cup X_2 \cup R_2 \models \perp$ is analogous. □

A bit surprisingly maybe, even verifying minimal issues can be done in polynomial time.

Lemma 2. *Let $\mathcal{B}$ be a rule base, $R_1, R_2 \subseteq \mathcal{B}$, and X_1, X_2 consistent sets of literals. Checking whether (R_1, X_1, R_2, X_2) is a minimal issue can be done in polynomial time.*

Proof. Checking whether (R_1, X_1, R_2, X_2) is an issue can be done as in Lemma 1. For minimality, we check for each $R'_1 \subseteq R_1, R'_2 \subseteq R_2$, where exactly one of R'_1, R'_2 contains one rule less, whether $X_1 \cup R'_1 \cup X_2 \cup R'_2 \models \perp$. If that is the case, (R_1, X_1, R_2, X_2) cannot be a minimal issue. Note that there are only polynomially many tuples $(|R_1| + |R_2|)$ to check, each check being polynomial, cf. the proof of Lemma 1. $\square$

Let us now turn to our central notion of quasi-inconsistency and the task to check whether a rule base is quasi-inconsistent:

DEC-QI	**Input**:	rule base $\mathcal{B}$
	Output:	TRUE iff $\mathcal{B}$ is quasi-inconsistent

It turns out that DEC-QI is intractable even for acyclic rule bases.

Proposition 4. DEC-QI *is NP-complete. This remains true even for $\mathcal{R}_{\mathcal{L}}^{acyclic}$.*

Proof. In order to show NP-membership consider the following non-deterministic algorithm. On input $\mathcal{B}$, first guess sets of rules R_1, R_2 with $R_1, R_2 \subseteq \mathcal{B}$ and consistent sets of literals X_1, X_2. If $t = (R_1, X_1, R_2, X_2)$ is an issue of $\mathcal{B}$, return TRUE, otherwise return FALSE. Observe that this check can be done in polynomial time according to Lemma 1. This shows DEC-QI $\in \mathsf{NP}$.

In order to show NP-hardness we reduce the classical satisfiability problem SAT to DEC-QI. Let Φ be an instance of SAT over the signature A (=set of atoms), i. e., $\Phi = C_1 \wedge \ldots \wedge C_n$ where each $C_i = l_{i,1} \vee \ldots \vee l_{i,n(i)}$ with literals $l_{i,1}, \ldots, l_{i,n(i)}$ over A for $i = 1, \ldots, n$ (recall that a literal is either an atom $a \in A$ or its negation $\neg a$). The question is whether Φ is satisfiable, i. e., whether we can find an interpretation $I : A \rightarrow \{\mathsf{T}, \mathsf{F}\}$ s. t. for all $i = 1, \ldots, n$ there is a $k \in \{1, \ldots, n(i)\}$ with $I(l) = \mathsf{T}$ (if $l = l_{i,k}$ is an atom) or $I(l') = \mathsf{F}$ (if $\neg l' = l_{i,k}$ is a negated atom). On input Φ we construct a rule base $\mathcal{B}_\Phi$ as follows. For each clause C_i, $i = 1, \ldots, n$, we create two new atoms α_i, α'_i (both with the informal meaning that α_i/α'_i is derivable in $\mathcal{B}_\Phi$ if C_i is satisfied). For each clause $C_i = l_{1,1} \vee \ldots \vee l_{1,n(i)}$ we construct $2n(i)$ rules of the form

$$\mathcal{B}_i = \{\alpha_i \leftarrow l_{i,1}; \quad \ldots \quad ; \quad \alpha_i \leftarrow l_{i,n(i)}; \quad \alpha'_i \leftarrow l_{i,1}; \quad \ldots \quad ; \quad \alpha'_i \leftarrow l_{i,n(i)}\}$$

Then we create yet another new atom π and define $\mathcal{B}_\Phi$ to be composed of the above rules and two further rules:

$$\mathcal{B}_\Phi = \mathcal{B}_1 \cup \ldots \cup \mathcal{B}_n \cup \{\pi \leftarrow \alpha_1, \ldots, \alpha_n; \quad \neg\pi \leftarrow \alpha'_1, \ldots, \alpha'_n\}$$

We claim that Φ is satisfiable iff $\mathcal{B}_\Phi$ is quasi-inconsistent. We first show that satisfiability of Φ implies quasi-inconsistency of $\mathcal{B}_\Phi$. Let I be an interpretation that satisfies Φ. Without loss of generality, for each clause C_i let $l_{1,i}$ be the literal that is satisfied by I (can be easily achieved by reordering the literals in each clause). Define $X = \{l_{1,1}, \ldots, l_{1,n}\}$ to be the set of all these literals. Furthermore, define

$$\begin{aligned}
R &= \{\alpha_1 \leftarrow l_{1,1}; \quad \ldots; \quad \alpha_n \leftarrow l_{1,n}\} \\
R' &= \{\alpha'_1 \leftarrow l_{1,1}; \quad \ldots; \quad \alpha'_n \leftarrow l_{1,n}\} \\
R_1 &= R \cup \{\pi \leftarrow \alpha_1, \ldots, \alpha_n\} \\
R_2 &= R' \cup \{\neg\pi \leftarrow \alpha'_1, \ldots, \alpha'_n\}
\end{aligned}$$

By construction, X activates R and R'. Let $X' \subseteq X$ s. t. X' minimally activates R resp R'. Then X' also minimally activates both R_1 and R_2. It follows that (R_1, X_1, R_2, X_1) is an issue of $\mathcal{B}_\Phi$ showing that the latter is quasi-inconsistent. For the other direction, assume that $\mathcal{B}_\Phi$ is quasi-inconsistent and let (R_1, X_1, R_2, X_1) be an issue of $\mathcal{B}_\Phi$. Observe that the only rules able to derive contradictory claims in $\mathcal{B}_\Phi$ are the two rules with heads π and $\neg\pi$, respectively. So one of these rules must be in R_1 and the other in R_2 (if they both would be in one of them this would violate condition 4 of Definition 2). Assume $\pi \leftarrow \alpha_1, \ldots, \alpha_n \in R_1$, in order to activate this rule, one rule for each α_i, $i = 1, \ldots, n$ needs to be present in R_1. Moreover, for each such rule, one of the literals of the corresponding clause must be present in an activation set X_1. From these literals, an interpretation I can be constructed satisfying all clauses C_i, $i = 1, \ldots, n$ in analogy to the reverse direction before (note that this interpretation is partial, not all propositions need to occur in $X_1 \cup X_2$; however, the truth value of the remaining propositions is irrelevant and can be set arbitrary).

Finally, observe that $\mathcal{B}_\Phi$ is of polynomial size wrt. Φ. This gives a polynomial-time reduction from SAT to DEC-QI, showing that the latter is NP-hard. The reader can easily verify that the construction above yields an acyclic rule base, also showing NP-hardness for this special case. □

Moreover, Given a rule $r \in \mathcal{B}$, checking whether r is contained in at least one minimal issue is also intractable.

Lemma 3. *Let $\mathcal{B}$ be a rule base and $r \in \mathcal{B}$. Checking whether there is a minimal issue (R_1, X_1, R_2, X_2) with $r \in R_1 \cup R_2$ is* NP*-complete.*

Proof. For NP-membership, we guess a tuple (R_1, X_1, R_2, X_2) with $r \in R_1 \cup R_2$ and check in polynomial time using Lemma 2 whether (R_1, X_1, R_2, X_2) is a minimal issue.

For NP-hardness, we use the exact same reduction as in the proof of Proposition 4 and ask whether the rule $\pi \leftarrow \alpha_1, \ldots, \alpha_n$ is contained in a minimal issue. This is exactly the case iff the input CNF formula Φ is satisfiable. □

We now consider some counting problems related to issues:

#ISSUES	**Input**:	rule base $\mathcal{B}$
	Output:	$\lvert \mathsf{Issues}(\mathcal{B}) \rvert$
#MINISSUES	**Input**:	rule base $\mathcal{B}$
	Output:	$\lvert \mathsf{MinIssues}(\mathcal{B}) \rvert$

Using a similar reduction as in the proof of Proposition 4 we can show the following results.

Proposition 5. #ISSUES *and* #MINISSUES *are* #P*-complete.*[2]

Proof. For #P-membership, Lemmas 1 and 2 already showed that checking whether a given tuple (R_1, X_1, R_2, X_2) is a (minimal) issue can be decided in polynomial time. It follows that #ISSUES and #MINISSUES are in #P.

For showing hardness, we reduce the problem #1-3-SAT to our problems that has been shown to be #P-complete in [14]. Given a formula Φ over $A = \{a_1, \ldots, a_m\}$ in 3-CNF, i.e., a formula of the form $\Phi = (l_{1,1} \vee l_{1,2} \vee l_{1,3}) \wedge \ldots \wedge (l_{n,1} \vee l_{n,2} \vee l_{n,3})$ (with exactly 3 literals per clause), we ask for the number of those interpretations $I : A \to \{\mathsf{T}, \mathsf{F}\}$ s.t. for all $i = 1, \ldots, n$ there is *exactly one* $k \in \{1, \ldots, n(i)\}$ with $I(l) = \mathsf{T}$ (if $l = l_{i,k}$ is an atom) or $I(l') = \mathsf{F}$ (if $\neg l' = l_{i,k}$ is a negated atom). We call an interpretation satisfying this condition *1-3-model* of Φ. On input Φ we construct a rule base $\mathcal{B}_\Phi$ as follows. For each clause C_i, $i = 1, \ldots, n$, we create two new atoms α_i, α'_i (both with the informal meaning that α_i/α'_i is derivable in $\mathcal{B}_\Phi$ if C_i is satisfied). For each clause $C_i = l_{i,1} \vee l_{i,2} \vee l_{i,3}$ we construct six rules

[2] #P is the complexity class of counting problems where the problem of deciding whether a particular element has to be counted is in P.

of the form

$$\begin{aligned}
\mathcal{B}_i = \{&\alpha_i \leftarrow l_{i,1}, \overline{l_{i,2}}, \overline{l_{i,3}};\\
&\alpha'_i \leftarrow l_{i,1}, \overline{l_{i,2}}, \overline{l_{i,3}};\\
&\alpha_i \leftarrow \overline{l_{i,1}}, l_{i,2}, \overline{l_{i,3}};\\
&\alpha'_i \leftarrow \overline{l_{i,1}}, l_{i,2}, \overline{l_{i,3}};\\
&\alpha_i \leftarrow \overline{l_{i,1}}, \overline{l_{i,2}}, l_{i,3};\\
&\alpha'_i \leftarrow \overline{l_{i,1}}, \overline{l_{i,2}}, l_{i,3}\}
\end{aligned}$$

Moreover, we create new atoms $\delta_1, \dots, \delta_m$ and construct the following rules

$$\mathcal{B}_A = \{\delta_i \leftarrow a_i; \quad \delta_i \leftarrow \neg a_i \mid i = 1, \dots, m\}$$

Then we create yet another new atom π and define $\mathcal{B}_\Phi$ to be composed of the above rules and two further rules:

$$\begin{aligned}
\mathcal{B}_\Phi = \mathcal{B}_1 \cup \ldots \cup \mathcal{B}_n \cup \mathcal{B}_A \cup \{&\pi \leftarrow \alpha_1, \dots, \alpha_n, \delta_1, \dots, \delta_m;\\
&\neg\pi \leftarrow \alpha'_1, \dots, \alpha'_n, \delta_1, \dots, \delta_m\}
\end{aligned}$$

We now claim that the number of 1-3-models of Φ is exactly the number of issues of $\mathcal{B}_\Phi$, which is exactly the number of minimal issues of $\mathcal{B}_\Phi$.

Let I be a 1-3-model if Φ. Define (X, R_1, X, R_2) via

$$\begin{aligned}
X &= \{a \mid a \in \mathcal{A}, I(a) = \mathsf{T}\} \cup \{\neg a \mid a \in \mathcal{A}, I(a) = \mathsf{F}\}\\
R_A &= \{\delta_i \leftarrow a_i \mid a_i \in X\} \cup \{\delta_i \leftarrow \neg a_i \mid \neg a_i \in X\}\\
R_1 &= R_A \cup \{\pi \leftarrow \alpha_1, \dots, \alpha_n, \delta_1, \dots, \delta_m\}\\
&\quad \cup \{\alpha_i \leftarrow l_{i,1}, \overline{l_{i,2}}, \overline{l_{i,3}} \mid I(l_{i,1}) = \mathsf{T}, I(l_{i,2}) = \mathsf{F}, I(l_{i,3}) = \mathsf{F}, i = 1, \dots, n\}\\
&\quad \cup \{\alpha_i \leftarrow \overline{l_{i,1}}, l_{i,2}, \overline{l_{i,3}} \mid I(l_{i,1}) = \mathsf{F}, I(l_{i,2}) = \mathsf{T}, I(l_{i,3}) = \mathsf{F}, i = 1, \dots, n\}\\
&\quad \cup \{\alpha_i \leftarrow \overline{l_{i,1}}, \overline{l_{i,2}}, l_{i,3} \mid I(l_{i,1}) = \mathsf{F}, I(l_{i,2}) = \mathsf{F}, I(l_{i,3}) = \mathsf{T}, i = 1, \dots, n\}\\
R_2 &= R_A \cup \{\neg\pi \leftarrow \alpha'_1, \dots, \alpha'_n, \delta_1, \dots, \delta_m\}\\
&\quad \cup \{\alpha'_i \leftarrow l_{i,1}, \overline{l_{i,2}}, \overline{l_{i,3}} \mid I(l_{i,1}) = \mathsf{T}, I(l_{i,2}) = \mathsf{F}, I(l_{i,3}) = \mathsf{F}, i = 1, \dots, n\}\\
&\quad \cup \{\alpha'_i \leftarrow \overline{l_{i,1}}, l_{i,2}, \overline{l_{i,3}} \mid I(l_{i,1}) = \mathsf{F}, I(l_{i,2}) = \mathsf{T}, I(l_{i,3}) = \mathsf{F}, i = 1, \dots, n\}\\
&\quad \cup \{\alpha'_i \leftarrow \overline{l_{i,1}}, \overline{l_{i,2}}, l_{i,3} \mid I(l_{i,1}) = \mathsf{F}, I(l_{i,2}) = \mathsf{F}, I(l_{i,3}) = \mathsf{T}, i = 1, \dots, n\}
\end{aligned}$$

Observe that for each clause C_i, $i = 1, \dots, n$ both R_1 and R_2 contain exactly one rule with head α_i and α'_1, respectively, and that exact rule is activated by X. It follows that X minimally activates both R_1 and R_2. Furthermore,

$X \cup R_1 \not\models \perp$, $X \cup R_2 \not\models \perp$, and $X \cup R_1 \cup R_2 \models \perp$ and therefore (X, R_1, X, R_2) is an issue of B_Φ. (X, R_1, X, R_2) is also a minimal issue as every rule in R_1 and R_2 is needed to entail π and $\neg\pi$, respectively.

Conversely, let (X_1, R_1, X_1, R_2) be any issue. As the only derivable conflict in R_Φ is between π and $\neg\pi$, the corresponding rules must be present in R_1 and R_2, respectively. Assume $\pi \leftarrow \alpha_1, \ldots, \alpha_n, \delta_1, \ldots, \delta_m \in R_1$ then there has to be at least one rule from $\{\delta_i \leftarrow a_i;\quad \delta_i \leftarrow \neg a_i\}$ for each $i = 1, \ldots, m$ in R_1. As X must be consistent not both rules can be activated, so there is exactly one of these rules in R_1. It also follows that X_1 is exactly the union of the premises of that rules. Due to $X_1 \subseteq X_2$ and X_2 must be consistent it follows $X_1 = X_2$ (the addition of any literal makes X_1 inconsistent). For each α_i there must be at least one of the three rules with head α_i in R_1, otherwise α_i could not be derived. As at most one of these rules can be activated by X_1, there is exactly one of the three rules in R_1 (which is also activated, otherwise (X_1, R_1, X_1, R_2) would not be an issue). The same applies to R_2 and the rules with head α'. Now observe that (X_1, R_1, X_1, R_2) is also a minimal issue as every rule in R_1 and R_2 is needed to entail π and $\neg\pi$, respectively. Moreover, define now an interpretation I via $I(a) = \mathsf{T}$ if $a \in X_1$ and $I(a) = \mathsf{F}$ if $\neg a \in X_1$. As X_1 activates each rule with head α_i for $i = 1, \ldots, n$, I satisfies exactly one literal of each clause C_i. It follows that I is a 1-3-model of Φ.

It follows that each 1-3-model of Φ corresponds exactly to one (minimal) issue of $\mathcal{B}_\Phi$. Therefore, their number is exactly the same. As $\mathcal{B}_\Phi$ is of polynomial size wrt. Φ we have shown that both #ISSUES and #MINISSUES are #P-hard. $\square$

5. Discussion and Conclusion

In this paper, we addressed the problem of potential inconsistencies in rule bases. As motivated in the introduction, this use-case is very common in domains such as business rules management, where rules that will always be activated together (but yield inconsistent conclusions) relate to actual modelling errors and need to be resolved by experts. To this aim, we introduced the notion of quasi-inconsistency and discussed various aspects related to the computational complexity on this matter.

Our notion of quasi-inconsistency is a generalisation of incoherence [9, 10, 11]. The notion of incoherence refers to the problem of unsatisfiable concepts in, e. g., description logics. For example, consider the description logic statements $A \sqsubseteq B$ (every A is a B), $A \sqsubseteq C$ (every A is a C) and $B \sqcap C \sqsubseteq \perp$ (there is nothing that is both B and C). Here, the concept A

is unsatisfiable, because in the presence of an individual that is of concept A, we derive an inconsistency. Phrased in our (propositional) language, an unsatisfiable concept corresponds to a (minimal) issue (R_1, X_1, R_2, X_2) where $X_1 = X_2$ is a singleton set. Therefore, quasi-inconsistency covers a wider spectrum of phenomena than that of incoherence.

Our results can be used to provide companies with an initial analysis of potential inconsistencies in business rule bases, and thus promotes inconsistency handling in the scope of business rule management. While our investigation on computational complexity showed that many problems are intractable (in the worst case), it has to be noted that this in line with results from classical inconsistency measurement [15], which investigates similar problems as we do. Actually, there are many classical inconsistency measures where decisions variants of computing their value, are higher up the polynomial hierarchy than "merely" NP. Indeed, the NP-completeness result of Proposition 4 and the $\#\mathsf{P}$-result of Proposition 5 allows us to use mature satisfiability solving and model counting techniques [16] for our problems. While this work focuses on an initial detection of potential inconsistencies, due to the close relation to the field of inconsistency measurement, it seems promising to also investigate means for assessing the *severity* of potential inconsistencies, which is part of future work.

Acknowledgements

This research is part of the research project "Handling Inconsistencies in Business Process Modeling", which is funded by the German Research Association (reference number: DE1983/9-1).

References

[1] I. Graham, Business rules management and service oriented architecture: a pattern language, John wiley & sons, 2007.

[2] M. L. Nelson, R. Sen, Business rules management in healthcare: A lifecycle approach, Decision Support Systems 57 (2014) 387–394.

[3] K. Batoulis, M. Weske, Disambiguation of DMN decision tables, in: Business Information Systems - 21st International Conference, 2018, pp. 236–249.

[4] K. Smit, M. Zoet, M. Berkhout, A verification framework for business rules management in the dutch government context, International Journal On Advances in Systems and Measurements 12 (2019) 101–112.

[5] C. Corea, P. Delfmann, Quasi-inconsistency in declarative process models, in: International Conference on Business Process Management, Springer, 2019, pp. 20–35.

[6] M. Hashmi, G. Governatori, H.-P. Lam, M. T. Wynn, Are we done with business process compliance: state of the art and challenges ahead, Knowledge and Information Systems 57 (1) (2018) 79–133.

[7] A. Hunter, L. Bertossi, T. Schaub (Eds.), Inconsistency Tolerance, Springer, 2005.

[8] M. Thimm, Inconsistency measurement, in: Proceedings of the 13th International Conference on Scalable Uncertainty Management (SUM'19), 2019, pp. 9–23.

[9] C. A. D. Deagustini, M. V. Martinez, M. A. Falappa, G. R. Simari, How does incoherence affect inconsistency-tolerant semantics for datalog±?, Annals of Mathematics and Artificial Intelligence 82 (2018) 43–68.

[10] G. Flouris, Z. Huang, J. Z. Pan, D. Plexousakis, H. Wache, Inconsistencies, negations and changes in ontologies, in: Proceedings of the 21st National Conference on Artificial Intelligence (AAAI'20), AAAI Press, 2006, pp. 1295–1300.

[11] G. Qi, A. Hunter, Measuring incoherence in description logic-based ontologies, in: Proceedings of the 6th international The semantic web and 2nd Asian conference on Asian semantic web conference, ISWC'07/ASWC'07, Springer-Verlag, Berlin, Heidelberg, 2007, pp. 381–394.

[12] M. Gelfond, V. Lifschitz, Classical negation in logic programs and disjunctive databases, New Generation Computing 9 (1991) 365–385.

[13] C. Papadimitriou, Computational Complexity, Addison-Wesley, 1994.

[14] N. Creignou, M. Hermann, On #p-completeness of some counting problems, Tech. rep., INRIA (1993).

[15] M. Thimm, J. P. Wallner, On the complexity of inconsistency measurement, Artificial Intelligence 275 (2019) 411–456.

[16] A. Biere, M. Heule, H. van Maaren, T. Walsh (Eds.), Handbook of Satisfiability, Vol. 185 of Frontiers in Artificial Intelligence and Applications, IOS Press, 2009.