MEASURING INCONSISTENCY WITH THE TABLEAU METHOD

JANDSON S. RIBEIRO
Artificial Intelligence Group, University of Hagen, Germany
jandson.ribeiro@fernuni-hagen.de

MATTHIAS THIMM
Artificial Intelligence Group, University of Hagen, Germany
matthias.thimm@fernuni-hagen.de

Abstract

We introduce a novel approach to measure inconsistency in knowledge bases that is based on the *Tableau Method* and derivations of contradictions from a knowledge base. This approach is purely syntactic and differs from previous approaches by neither taking minimal inconsistent sets nor non-classical semantics into account. We develop three concrete measures that take derivations of contradictions into account and investigate their compliance w. r. t. rationality postulates, expressivity, and computational complexity.

1 Introduction

An inconsistency measure $\mathcal{I}$ is a function mapping a knowledge base—e. g. a set of propositional sentences—to a non-negative real value, such that larger values indicate more severe inconsistency in the knowledge base [7, 9, 21]. Considering, e. g., the two knowledge bases $\mathcal{K}_1$ and $\mathcal{K}_2$ defined via

$$\mathcal{K}_1 = \{a, \neg a, b\} \qquad\qquad \mathcal{K}_2 = \{a \wedge b, \neg a, \neg b\}$$

one can see that both knowledge bases are inconsistent (in the classic-logical sense), but $\mathcal{K}_2$ may be judged "more inconsistent" as it contains contradictory information about both propositional atoms a and b while $\mathcal{K}_1$ has only contradictory information about a. So an inconsistency measure $\mathcal{I}$ focusing on this aspect may give $\mathcal{I}(\mathcal{K}_1) < \mathcal{I}(\mathcal{K}_2)$. The concept of a degree of inconsistency is not easily characterisable through either formal properties or a single measure. In fact, there are many proposals for desirable properties an inconsistency measure should satisfy and many proposals for inconsistency measures that satisfy certain subsets of these properties, see [21] for a survey.

One way to classify inconsistency measures is by differentiating whether they operate on the *formula level* or on the *language level*. The former category is also called the *syntactic approach* while the latter is called the *semantic approach* [9]. Measures belonging to the syntactic approach usually make use of *minimal inconsistent subsets*, i. e., subsets of the knowledge base that are inconsistent but removing any formula renders them consistent. For example, a simple measure is $\mathcal{I}_{\mathsf{MI}}$ [11], which assigns to a knowledge base simply the number of its minimal inconsistent subsets. For the knowledge bases from before we have therefore $\mathcal{I}_{\mathsf{MI}}(\mathcal{K}_1) = 1$ and $\mathcal{I}_{\mathsf{MI}}(\mathcal{K}_2) = 2$, since $\{a, \neg a\}$ is the only minimal inconsistent subset of $\mathcal{K}_1$ and $\{a \wedge b, \neg a\}$ and $\{a \wedge b, \neg b\}$ are the minimal inconsistent subsets of $\mathcal{K}_2$. Other measures also take the relationships between minimal inconsistent subsets into account [14] or exploit other notions such as maximal consistent subsets [1], but the commonality of these approaches is that they focus on conflicts between formulae of the knowledge base. On the other hand, measures belonging to the semantic approach focus on conflicts between language components. More precisely, these measures aim at identifying those atoms of the underlying language that are conflicting and they usually employ non-classical and many-valued logics as a tool for that [20]. For example, the measure $\mathcal{I}_c$ [8] assigns to a knowledge base the number of propositional atoms participating in the inconsistency using three-valued paraconsistent semantics. Without going into details, this measure gives $\mathcal{I}_c(\mathcal{K}_1) = 1$ and $\mathcal{I}_c(\mathcal{K}_2) = 2$ as well, as one resp. propositional atoms are participating in the conflicts of $\mathcal{K}_1$ resp. $\mathcal{K}_2$.

In this paper, we propose a different perspective for measuring inconsistency based on derivations of contradictions with *logical calculi*. In fact, we argue that the current distinction between *syntactic* and *semantic* approaches is mislabelled, as our new approach is purely syntactic and does not rely on notions such as minimal inconsistent subsets or maximal consistent subsets, which are actually semantically defined concepts. We consider the *Tableau Method* [17] as a prototypical logical calculus (also called *proof system*) and consider *proofs of contradiction* as a sequence of derivation rules that shows how a logical inconsistency can be derived from the knowledge base syntactically. We use such proofs as measures of inconsistency by assuming that 1.) the existence of many such proofs and 2.) the existence of short proofs indicates a larger degree of inconsistency.

To summarise, the contributions of this paper are as follows:

1. We define three inconsistency measures based on proofs of contradictions (Section 4). Our inconsistency measures explore the size and number of minimal tableaux to weigh the inconsistency within a knowledge base.

2. We analyse our measures in terms of rationality postulates (Section 5.1), expressivity (Section 5.2), and computational complexity (Section 5.3). Besides comparing our

new measures with the existing rationality postulates, we introduce a new postulate with the objective of identifying redundant information in producing inconsistency, and we show that our measures comply with that postulate. We show that our measures are maximally expressive, in the sense that it produces infinitely many values of inconsistency. As for complexity, due to open problems in the area of *proof complexity*, $\mathsf{EXPSPACE}$ is shown to be the tightest upper bound for various decision problems related to our measures.

Sections 2 and 3 provide the formal background and Section 6 concludes.

2 Preliminaries

Let At be an arbitrary fixed finite set of propositional atoms. We assume that the special symbols $\top, \bot$ (tautology and contradiction, respectively) are always contained in At, i. e., $\top, \bot \in \mathsf{At}$.

Definition 1. *Given a set of propositional atoms At, the* propositional language $\mathcal{L}(\mathsf{At})$ *corresponds to the language generated by the following grammar:*

$$\varphi := p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi;$$

where $p \in \mathsf{At}$.

As usual, $\neg$ denotes negation, $\wedge$ is conjunction, $\vee$ is disjunction. A *knowledge base* $\mathcal{K}$ w. r. t. a language $\mathcal{L}(\mathsf{At})$ is any finite subset $\mathcal{K} \subseteq \mathcal{L}(\mathsf{At})$. Let $\mathbb{K}(\mathsf{At})$ be the set of all knowledge bases w. r. t. to the language $\mathcal{L}(\mathsf{At})$. For any formula ϕ, let $\mathsf{At}(\phi) \subseteq \mathsf{At}$ be the set of atoms appearing in ϕ. When it is clear from context, we will omit At and simply write $\mathcal{L}$ and $\mathbb{K}$.

Definition 2. *Given a set of propositional atoms At, the* length *of a formula $\phi \in \mathcal{L}(\mathsf{At})$ is given by the function $\mathtt{len} : \mathcal{L}(\mathsf{At}) \to \mathbb{Z}_{\geq 0}$ inductively defined as*

- *if $\varphi \in \mathsf{At}$ then $\mathtt{len}(\varphi) = 1$;*
- $\mathtt{len}(\neg\varphi) = \mathtt{len}(\varphi) + 1$*;*
- $\mathtt{len}(\varphi \,\square\, \psi) = \mathtt{len}(\varphi) + \mathtt{len}(\psi) + 1$ *for* $\square \in \{\wedge, \vee\}$.

The size of a set A is denoted by $|A|$. An *interpretation* ω on At is a function $\omega : \mathsf{At} \to \{\mathsf{true}, \mathsf{false}\}$ with $\omega(\top) = \mathsf{true}$ and $\omega(\bot) = \mathsf{false}$. Let $\Omega(\mathsf{At})$ be the set of all interpretations on At. An interpretation ω *satisfies* an atom $a \in \mathsf{At}$, denoted as $\omega \models a$, iff $\omega(a) = \mathsf{true}$. Let $\omega \not\models \psi$ denote that ω does not satisfy a formula ψ. The relation $\models$ is inductively extended to general formulae as usual, that is,

$\omega \models \varphi \wedge \psi$ iff $\omega \models \varphi$ and $\omega \models \psi$

$\omega \models \varphi \vee \psi$ iff $\omega \models \varphi$ or $\omega \models \psi$

$\omega \models \neg\varphi$ iff $\omega \not\models \varphi$.

If $\omega \models \phi$ we also say that ω is a *model* of ϕ. Let $\mathsf{Mod}(\phi)$ denote the set of models of a formula ϕ. A formula $\phi \in \mathcal{L}(\mathsf{At})$ is *entailed* by $\psi \in \mathcal{L}(\mathsf{At})$, denoted by $\psi \models \phi$, if for all $\omega \in \Omega(\mathsf{At})$, $\omega \models \psi$ implies $\omega \models \phi$. Two formulae $\phi, \psi \in \mathcal{L}(\mathsf{At})$ are *equivalent*, denoted by $\phi \equiv \psi$, if both $\phi \models \psi$ and $\psi \models \phi$. Furthermore, two sets of formulae X_1, X_2 are *semi-extensionally equivalent* if there is a bijection $s : X_1 \to X_2$ such that for all $\alpha \in X_1$ we have $\alpha \equiv s(\alpha)$ [18]. We denote this by $X_1 \equiv^s X_2$.

3 The Tableau Method

In general, a *proof system* is a set of schematic inference rules that allows the purely syntactic transformation of formulae. Well-known proof systems are e. g. Frege's propositional calculus [5] and Gentzen-style proof systems [6]. In this section, we review the *Tableau Method* for classical propositional logics [17]. The *Tableau method* is a proof system based on refutation: given a knowledge base $\mathcal{K}$, it constructs a binary tree by applying a sequence of rules until either (i) all the branches of the tree present a contradiction or (ii) no rules can be further applied. In the first case, the knowledge base $\mathcal{K}$ is inconsistent; while in the second case, as long as there is at least one branch free of contradiction, $\mathcal{K}$ is consistent. The constructed tree is referred to as a *tableau*. In the remainder of this section, we review the *set-labelled* variant of the *Tableau Method*, where the constructed tableau is a binary tree in which each node is labelled with a set of formulae.

Definition 3. *A set-labelled tree is a tuple $T = (N, E, \lambda)$ where*

- *(N, E) is a tree, s.t N is the set of nodes, $E \subseteq N \times N$ the set of edges,*
- *$\lambda : N \to \mathbb{K}(\mathsf{At})$ is a labelling function.*

The labelling function λ maps each node of the tree to a set of formulae in $\mathcal{L}(\mathsf{At})$. Given a set-labelled tree $T = (N, E, \lambda)$, the children of a node n are given by $\mathsf{children}(n) = \{n' \in N \mid (n, n') \in E\}$, and the leaf nodes of T are given by $\mathsf{leaf}(T) = \{n \in N \mid \mathsf{children}(n) = \emptyset\}$. Moreover, the root of T is given by $\mathsf{root}(T)$.

We will postpone the formal definition of the set-labelled tableau (see Definition 5) until we have all the necessary ingredients. We start by giving an intuition of how the tableau method works. As mentioned above, a tableau, which is a set-labelled binary tree with some further constraints, is constructed by applying a set of non-deterministic *derivation rules*,

$$(\neg\neg_e)\ \frac{\mathcal{K}\cup\{\neg\neg\varphi\}}{\mathcal{K}\cup\{\neg\neg\varphi\}\cup\{\varphi\}}$$

$$(DM_\wedge)\ \frac{\mathcal{K}\cup\{\neg(\varphi\wedge\psi)\}}{\mathcal{K}\cup\{\neg(\varphi\wedge\psi)\}\cup\{\neg\varphi\vee\neg\psi\}} \qquad (DM_\vee)\frac{\mathcal{K}\cup\{\neg(\varphi\vee\psi)\}}{\mathcal{K}\cup\{\neg(\varphi\vee\psi)\}\cup\{\neg\varphi\wedge\neg\psi\}}$$

$$(\wedge_e)\frac{\mathcal{K}\cup\{\varphi\wedge\psi\}}{\mathcal{K}\cup\{\varphi\wedge\psi\}\cup\{\varphi,\psi\}}$$

$$(\vee_e)\frac{\mathcal{K}\cup\{\varphi\vee\psi\}}{\mathcal{K}\cup\{\varphi\vee\psi\}\cup\{\varphi\}\mid\mathcal{K}\cup\{\varphi\vee\psi\}\cup\{\psi\}}$$

Figure 1: Derivation rules for the *Tableau Method*.

so several tableaux can exist for a same knowledge base $\mathcal{K}$. The procedure for constructing a tableau works by first creating a tree with only the root node (called a *root tree*), which is labelled with the knowledge base $\mathcal{K}$ itself. This initial root tree is then expanded by applying one of the derivation rules depicted in Fig. 1. When applied, these rules append new nodes to one of the leaf nodes of the tree. In the derivation rules $DM_\wedge$ and $DM_\vee$, DM stands for De Morgan, as these rules correspond to the De Morgan laws. While rules $\neg\neg_e, DM_\wedge, DM_\vee$ and $\wedge_e$ append a single leaf node, rule (5) opens two branches.

Each node is labelled with a set of formulae, and therefore, there might exist more than one possible rule to be applied on such a leaf node, or even more than one choice for a same applicable rule. We define a function σ that exhibits explicitly all the possible extensions for non-branching rules, that is, rules $\neg\neg_e, DM_\wedge, DM_\vee$ and $\wedge_e$. The set of all possible extensions for the branching rule $\vee_e$ is given by the function γ below. The set of all rule names are given by $\mathcal{R}_{TB} = \{\neg\neg_e, DM_\wedge, DM_\vee, \wedge_e, \vee_e\}$.

Definition 4. *Let* $\sigma : \mathcal{R}_{TB} \times \mathbb{K}(\mathsf{At}) \to \mathbb{K}(\mathsf{At})$ *be such that*

1. $\sigma(\neg\neg_e, \mathcal{K}) = \{\mathcal{K}\cup\{\varphi\} \in \mathbb{K}(\mathsf{At}) \mid \neg\neg\varphi \in \mathcal{K}\}$
2. $\sigma(\wedge_e, \mathcal{K}) = \{\mathcal{K}\cup\{\varphi,\psi\} \in \mathbb{K}(\mathsf{At}) \mid \varphi\wedge\psi \in \mathcal{K}\}$
3. $\sigma(DM_\wedge, \mathcal{K}) = \{\mathcal{K}\cup\{\neg\varphi\vee\neg\psi\} \in \mathbb{K}(\mathsf{At}) \mid \neg(\varphi\wedge\psi) \in \mathcal{K}\}$
4. $\sigma(DM_\vee, \mathcal{K}) = \{\mathcal{K}\cup\{\neg\varphi\wedge\neg\psi\} \in \mathbb{K}(\mathsf{At}) \mid \neg(\varphi\vee\psi) \in \mathcal{K}\}$

Let $\gamma : \mathbb{K}(\mathsf{At}) \to \mathbb{K}(\mathsf{At}) \times \mathbb{K}(\mathsf{At})$ *be such that*

$$\gamma(\mathcal{K}) = \{(X,Y) \in \mathbb{K}(\mathsf{At}) \times \mathbb{K}(\mathsf{At}) \mid X = \mathcal{K}\cup\{\varphi\}, Y = \mathcal{K}\cup\{\psi\}, \textit{for some } \varphi\vee\psi \in \mathcal{K}\}$$

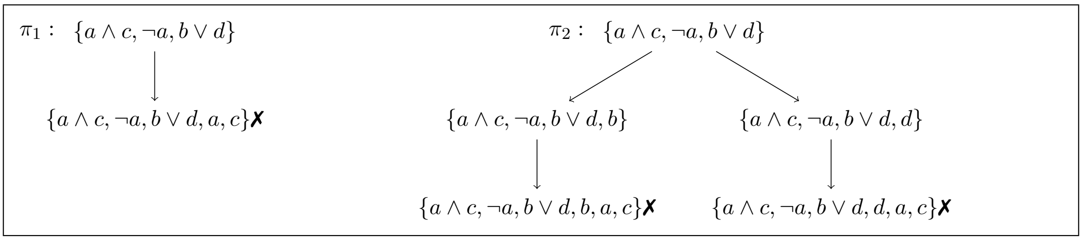

Figure 2: Example of two tableaux for the knowledge base $\mathcal{K} = \{a \wedge c, \neg a, b \vee d\}$.

Definition 5. *A* tableau *for a knowledge base $\mathcal{K} \subseteq \mathcal{L}(\mathsf{At})$ is a binary set-labelled tree (N, E, λ) such that*

- *$\lambda(r) = \mathcal{K}$, where r is the root node;*
- *for each node $n \in N$:*
  1. *$\lambda(n) \neq \lambda(n')$, for all $n' \in \mathsf{children}(n)$;*
  2. *if $\mathsf{children}(n) = \{n_1\}$ then $\lambda(n_1) \in \sigma(\varepsilon, \lambda(n))$, for a derivation rule $\varepsilon \in \mathcal{R}_{TB}$;*
  3. *if $\mathsf{children}(n) = \{n_1, n_2\}$ and $n_1 \neq n_2$ then $(\lambda(n_1), \lambda(n_2)) \in \gamma(\lambda(n))$ or $(\lambda(n_2), \lambda(n_1)) \in \gamma(\lambda(n))$.*

Conditions 1 to 3 guarantee that a tableau is generated according to the application of the rules in $\mathcal{R}_{TB}$. Condition 1 is imposed in order to avoid redundant tableaux. Specifically, the application of a rule on a node of a tableau needs to yield children nodes labelled with new formulae. This will become important since we are interested in *minimal* proofs of contradiction. The Greek letter π will be used to denote a tableau.

Example 6. *Consider the inconsistent knowledge base $\mathcal{K} = \{a \wedge c, \neg a, b \vee d\}$. Fig. 2 illustrates two tableaux π_1 and π_2 for $\mathcal{K}$. The root node of every tableau is labelled with the knowledge base itself $\mathcal{K}$. There are two possible rules to apply at the root node: (i) rule $\wedge_e$ creates a single child node with the added sub-formulae a and c (tableau π_1); (ii) rule $\vee_e$ creates two children node, one labelled with the sub-formula b with $\mathcal{K}$ and another with the sub-formula d with $\mathcal{K}$ (tableau π_2).*

If a formula α appears in the leaf node of a tableau π for a knowledge base $\mathcal{K}$, then we say that $\mathcal{K}$ *structurally derives* α, denoted by $\mathcal{K} \vdash \alpha$. For instance, in Example 6 the tableau π_1 has the formula c in its leaf node, therefore $\mathcal{K} \vdash c$.

If a node contains a formula and its negation then we say that such a node has a *clash*. More precisely, if there are formulae $\varphi, \neg\varphi \in \lambda(n)$ then n has a clash. Each leaf node of the tableaux π_1 and π_2 from Example 6 has a clash, as each leaf node has the formula a and its negation $\neg a$. If every leaf node of a tableau has a clash then such a tableau is said to be *closed*. The tableaux π_1 and π_2 from Example 6 are both closed. The set of all closed tableaux for a knowledge base $\mathcal{K}$ is given by $\mathcal{T}_\perp(\mathcal{K})$.

Theorem 7. *[17] A knowledge base $\mathcal{K} \in \mathbb{K}(\mathsf{At})$ is inconsistent iff $\mathcal{T}_\perp(\mathcal{K}) \neq \emptyset$.*

As we are interested in minimal proofs of contradiction, we introduce the notion of a closed tableau being shorter than other closed tableau.

Definition 8. *A closed tableau π is* shorter than a closed tableau π', *denoted as $\pi \preceq \pi'$, iff there is an injection $\tau : \mathsf{leaf}(\pi) \to \mathsf{leaf}(\pi')$ such that $\lambda(n) \subseteq \lambda(\tau(n))$. Given a knowledge base $\mathcal{K}$, a closed tableau $\pi \in \mathcal{T}_\perp(\mathcal{K})$ is* minimal *iff for all $\pi' \in \mathcal{T}_\perp(\mathcal{K})$, if $\pi' \preceq \pi$ then $\pi \preceq \pi'$. The set of minimal tableaux for a given knowledge base $\mathcal{K}$ is given by $\mathcal{T}_\perp^{min}(\mathcal{K})$.*

Intuitively, a closed tableau π is shorter than a tableau π' if each set of formulae that clashes (what are present in the leaf nodes of the tableaux) are subsets of the leaf nodes of π'. For instance, the tableau π_1 from Example 6 is shorter than the tableau π_2 from the same example. We say that a tableau is redundant if two different branches lead to the same clash of formulae labelled on their leaf nodes, as it occurs with the tableau π_2 from Example 6. The injection condition guarantees that redundant tableaux are identified and therefore are not among the minimal tableaux. For instance, the tableau π_1 from Example 6 is minimal, while π_2 is not minimal.

4 Measuring inconsistency via Tableaux

An inconsistency measure is a function $\mathcal{I} : \mathbb{K}(\mathsf{At}) \to \mathbb{R}^\infty_{\geq 0}$ that maps each knowledge base $\mathcal{K}$ to a non-negative real number [7, 22]. Intuitively, larger values $\mathcal{I}(\mathcal{K})$ indicate a larger degree of inconsistency in $\mathcal{K}$, while 0 is reserved to indicate the absence of inconsistency.

A closed tableau exemplifies the reasoning effort to detect the presence of an inconsistency and thus gives rise to quantitative measures of inconsistency. The following principles are our main motivation to study measures based on tableaux:

1. If there are more ways to derive inconsistency in a knowledge base $\mathcal{K}$ than there are in a knowledge base $\mathcal{K}'$, then $\mathcal{K}$ should be regarded as more inconsistent than $\mathcal{K}'$. This principle represents a form of monotonicity of inconsistency w. r. t. number of closed tableaux.

2. Smaller closed tableaux indicate a larger degree of inconsistency than larger closed tableaux. The rationale behind this principle can be motivated by the *lottery paradox* [15]: if there are many lottery tickets it is rational to assume for each ticket holder that he will not win and the less tickets there are the less rational this assumption

becomes. In the first case, the inconsistency (on the fact that one ticket will win and every ticket holder thinks he will not win) is not that much apparent as in the case of just two tickets. A tableau for the first case would include many more steps to show the inconsistency than in the second case.

Both principles capture the intuition that a knowledge base is more inconsistent if the computational effort to find an inconsistency is low. This is indeed the case if there are many ways to prove inconsistency (e. g. a random method would more likely find a proof) and these proofs are short (as the depth of the search of such an algorithm does not need to be high).

We implement the above principle in the following inconsistency measures:

Definition 9. *The three inconsistency measures are* $\mathcal{I}^{\#} : \mathbb{K}(\mathsf{At}) \rightarrow \mathbb{R}_{\geq 0}, \mathcal{I}^{\min} : \mathbb{K}(\mathsf{At}) \rightarrow \mathbb{R}_{\geq 0}$*, and* $\mathcal{I}^{\sum} : \mathbb{K}(\mathsf{At}) \rightarrow \mathbb{R}_{\geq 0}$

$$\mathcal{I}^{\#}(\mathcal{K}) = |\mathcal{T}_{\perp}^{min}(\mathcal{K})|$$

$$\mathcal{I}^{\min}(\mathcal{K}) = \begin{cases} \dfrac{1}{min\{|A| \mid A \in \mathcal{T}_{\perp}^{min}(\mathcal{K})\}} & \text{, } \textit{if } \mathcal{T}_{\perp}(\mathcal{K}) \neq \emptyset \\ 0 & \textit{otherwise.} \end{cases}$$

$$\mathcal{I}^{\sum}(\mathcal{K}) = \begin{cases} \sum_{A \in \mathcal{T}_{\perp}^{min}(\mathcal{K})} \dfrac{1}{|A|} & \textit{if } \mathcal{T}_{\perp}^{min}(\mathcal{K}) \neq \emptyset \\ 0 & \textit{otherwise} \end{cases}$$

The inconsistency measure $\mathcal{I}^{\#}$ focuses on the first principle and simply takes the number of minimal closed tableaux as the degree of inconsistency. The measure $\mathcal{I}^{\min}$ focuses on the second principle and takes the reciprocal size of a minimal closed tableau as the degree of inconsistency. Finally, the measure $\mathcal{I}^{\sum}$ combines both principles by summing up the reciprocal sizes of all minimal closed tableaux.

Example 10. *Consider the knowledge bases* $\mathcal{K}_1$ *and* $\mathcal{K}_2$ *below:*

$$\mathcal{K}_1 = \{a \wedge c, \neg a, b\} \qquad\qquad \mathcal{K}_2 = \{a \wedge b, c \wedge d, \neg a, \neg d\}.$$

Note that $\mathcal{K}_1$ *has only one minimal tableau:*

$$\pi = \begin{array}{c} \{a \wedge c, \neg a, b\} \\ | \\ \{a \wedge c, \neg a, b, a, c\} \end{array}$$

Therefore, $\mathcal{I}^{\#}(\mathcal{K}_1) = 1$*, and* $\mathcal{I}^{\min}(\mathcal{K}_1) = \mathcal{I}^{\sum}(\mathcal{K}_1) = 1/2$*. For* $\mathcal{K}_2$ *we have the following two minimal closed tableaux*

$$\pi_1 = \begin{array}{c} \{a \wedge b, c \wedge d, \neg a, \neg d\} \\ | \\ \{a \wedge b, a, b, c \wedge d, \neg a, \neg d\} \end{array} \qquad \pi_2 = \begin{array}{c} \{a \wedge b, c \wedge d, \neg a, \neg d\} \\ | \\ \{a \wedge b, c \wedge d, c, d, \neg a, \neg d\} \end{array}$$

Therefore, $\mathcal{I}^{\#}(\mathcal{K}_2) = 2$*,* $\mathcal{I}^{\min}(\mathcal{K}_2) = 1/2$ *and* $\mathcal{I}^{\sum}(\mathcal{K}_2) = 2/2 = 1$.

In general, our measures take a radically different perspective on inconsistency measurement, which is also illustrated by the fact that these measures do not conform with many postulates proposed for inconsistency measures so far (see Section 5). Our aim with these measures is to investigate a new foundation of inconsistency measurement, i. e., one based on syntactic derivations instead of semantical concepts.

5 Analysis

In this section we conduct an analytical evaluation of our measures, focussing on compliance to rationality postulates, expressivity, and computational complexity.

5.1 Rationality Postulates

Many rationality postulates have been proposed for inconsistency measures, see [21] for a survey. However, many of these postulates are disputed and there is up to now no consensus on which of these postulates are desirable and which are not, see also [3] for a discussion. In fact, there is only one postulate which can be regarded as the defining property of an inconsistency measure $\mathcal{I}$ [10]:

Consistency **(CO)** $\mathcal{I}(\mathcal{K}) = 0$ if and only if $\mathcal{K}$ is consistent

For all other postulates proposed in the literature, we can find (reasonable) proposals of inconsistency measures that violate these postulates, see [21] for an overview. We compile below the existing rationality postulates from the literature, and we investigate the compliance of our measures with such postulates. For the presentation of the postulates, we will first need the following auxiliary definitions:

Definition 11. *A set* $M \subseteq \mathcal{K}$ *is a* minimal inconsistent subset *(MI) of* $\mathcal{K}$*, if* $M \models \perp$ *and there is no* $M' \subset M$ *with* $M' \models \perp$*. Let* $\mathsf{MI}(\mathcal{K})$ *be the set of all MIs of* $\mathcal{K}$*. A formula* $\alpha \in \mathcal{K}$ *is called* free formula *if* $\alpha \notin \bigcup \mathsf{MI}(\mathcal{K})$*. Let* $\mathsf{Free}(\mathcal{K})$ *be the set of all free formulae of* $\mathcal{K}$*.*

Definition 12. *A formula $\alpha \in \mathcal{K}$ is a* safe formula *if it is consistent and $\mathsf{At}(\alpha) \cap \mathsf{At}(\mathcal{K} \setminus \{\alpha\}) = \emptyset$. Let $\mathsf{Safe}(\mathcal{K})$ be the set of all safe formulae of $\mathcal{K}$.*

Let $\mathcal{I}$ be any function $\mathcal{I} : \mathbb{K} \to \mathbb{R}^{\infty}_{\geq 0}$, $\mathcal{K}, \mathcal{K}' \in \mathbb{K}$, and $\alpha, \beta \in \mathcal{L}(\mathsf{At})$. The rationality postulates for inconsistency measure in the literature, see [20] for a survey on the subject, are:

***Normalization* (NO)** $0 \leq \mathcal{I}(\mathcal{K}) \leq 1$

***Monotony* (MO)** If $\mathcal{K} \subseteq \mathcal{K}'$ then $\mathcal{I}(\mathcal{K}) \leq \mathcal{I}(\mathcal{K}')$

***Free-formula independence* (IN)** If $\alpha \in \mathsf{Free}(\mathcal{K})$ then $\mathcal{I}(\mathcal{K}) = \mathcal{I}(\mathcal{K} \setminus \{\alpha\})$

***Dominance* (DO)** If $\alpha \not\models \perp$ and $\alpha \models \beta$ then $\mathcal{I}(\mathcal{K} \cup \{\alpha\}) \geq \mathcal{I}(\mathcal{K} \cup \{\beta\})$

***Safe-formula independence* (SI)** If $\alpha \in \mathsf{Safe}(\mathcal{K})$ then $\mathcal{I}(\mathcal{K}) = \mathcal{I}(\mathcal{K} \setminus \{\alpha\})$

***Super-Additivity* (SA)** If $\mathcal{K} \cap \mathcal{K}' = \emptyset$ then $\mathcal{I}(\mathcal{K} \cup \mathcal{K}') \geq \mathcal{I}(\mathcal{K}) + \mathcal{I}(\mathcal{K}')$

***Penalty* (PY)** If $\alpha \notin \mathsf{Free}(\mathcal{K})$ then $\mathcal{I}(\mathcal{K}) > \mathcal{I}(\mathcal{K} \setminus \{\alpha\})$

***MI-separability* (MI)** If $\mathsf{MI}(\mathcal{K} \cup \mathcal{K}') = \mathsf{MI}(\mathcal{K}) \cup \mathsf{MI}(\mathcal{K}')$ and $\mathsf{MI}(\mathcal{K}) \cap \mathsf{MI}(\mathcal{K}') = \emptyset$ then $\mathcal{I}(\mathcal{K} \cup \mathcal{K}') = \mathcal{I}(\mathcal{K}) + \mathcal{I}(\mathcal{K}')$

***MI-normalization* (MN)** If $M \in \mathsf{MI}(\mathcal{K})$ then $\mathcal{I}(M) = 1$

***Attenuation* (AT)** $M, M' \in \mathsf{MI}(\mathcal{K})$ and $|M| > |M'|$ implies $\mathcal{I}(M) < \mathcal{I}(M')$

***Equal Conflict* (EC)** $M, M' \in \mathsf{MI}(\mathcal{K})$ and $|M| = |M'|$ implies $\mathcal{I}(M) = \mathcal{I}(M')$

***Almost Consistency* (AC)** Let $M_1, M_2, \ldots$ be a sequence of minimal inconsistent sets M_i with $\lim_{i\to\infty} |M_i| = \infty$, then $\lim_{i\to\infty} \mathcal{I}(M_i) = 0$

***Contradiction* (CD)** $\mathcal{I}(\mathcal{K}) = 1$ if and only if for all $\emptyset \neq \mathcal{K}' \subseteq \mathcal{K}$, $\mathcal{K}' \models \perp$

***Free Formula Dilution* (FD)** If $\alpha \in \mathsf{Free}(\mathcal{K})$ then $\mathcal{I}(\mathcal{K}) \geq \mathcal{I}(\mathcal{K} \setminus \{\alpha\})$

***Irrelevance of Syntax* (SY)** If $\mathcal{K} \equiv^s \mathcal{K}'$ then $\mathcal{I}(\mathcal{K}) = \mathcal{I}(\mathcal{K}')$

***Exchange* (EX)** If $\mathcal{K}' \not\models \perp$ and $\mathcal{K}' \equiv \mathcal{K}''$ then $\mathcal{I}(\mathcal{K} \cup \mathcal{K}') = \mathcal{I}(\mathcal{K} \cup \mathcal{K}'')$

***Adjunction Invariance* (AI)** $\mathcal{I}(\mathcal{K} \cup \{\alpha, \beta\}) = \mathcal{I}(\mathcal{K} \cup \{\alpha \wedge \beta\})$

As mentioned above, the postulate CO addresses the basic property of an inconsistency measure to differentiate between consistent and inconsistent knowledge bases. The postulate NO expresses that the degree of inconsistency is a relative notion that is normalized in the unit interval. MO states that adding information can only increase the degree of inconsistency. IN states that adding free formulae cannot change the degree of inconsistency and DO states that substituting a formula with a semantically weaker version cannot increase the degree of inconsistency. For a discussion on the rationale of the other postulates, see [20].

It is important to stress that there is no consensus about which postulates should be satisfied or which ones should not. However, there are scenarios in which some of the postulates are clearly unsuitable. This is the case of the following postulates: IN, PY, DO, SA, MN, CD, MI, AT, EC, EX, SY and AI. We explain below why each one of such postulates is not adequate under our principles of measuring inconsistency. In fact, none of our measures satisfy these postulates.

- IN: It states that the removal of a free formula does not decrease the inconsistency degree of a knowledge base. Although this intuition might seem plausible at a first glance, it is counter-intuitive under our second principle of inconsistency degree. Let $\mathcal{K} = \{(a \vee b) \wedge (a \vee \neg b), \neg b\}$, and $\mathcal{K}' = \{(a \vee b) \wedge (a \vee \neg b), \neg b, a\}$. Observe that a is free in $\mathcal{K}'$, but the presence of a in $\mathcal{K}'$ makes it much easier to prove the inconsistency of $\mathcal{K}'$ than in $\mathcal{K}$: to prove the inconsistency of $\mathcal{K}$, one needs to take the case distinction of both disjunctive formulae $a \vee b$ and $\neg a \vee b$; while for $\mathcal{K}'$ the proof of inconsistency is much easier because only the case distinction of $\neg a \vee b$ is necessary due to the presence of a. Therefore, free formulae should indeed be considered for assessing the degree of inconsistency in a knowledge base. Therefore, under our second principle IN becomes undesirable.

- PY: this postulate is the dual of IN, removing free-formulae should strictly reduce the inconsistency degree. Analogous to our reasons against IN, as adding free formulae does not necessarily contributes to augmenting the inconsistency degree, removing them should not contribute to making it less inconsistent either.

- DO: According to this postulate, stronger formulae can only make a knowledge base more inconsistent than weaker formulae. This postulate is in conflict with our second principle of inconsistency. To illustrate this, consider the knowledge base $\mathcal{K} = \{\neg a \wedge \neg b\}$, and the formulae $\alpha = c \wedge (a \vee b) \wedge (a \vee \neg b)$, and a. Observe that $\alpha \models a$. It is much easier to prove that $\mathcal{K}_1 = \mathcal{K} \cup \{a\}$ is inconsistent than to prove that $\mathcal{K}_2 = \mathcal{K} \cup \{\alpha\}$ is inconsistent, because for the former the contradiction is evident, while for the latter one needs to consider the case distinction due to the disjunction $a \vee b$. According to our principle, the knowledge base $\mathcal{K}_1$ should be more inconsistent than $\mathcal{K}_2$, opposed to DO.

- SA: This postulate imposes a strict form of monotonicity. It states that if two knowledge bases share no formulae, then their union present an inconsistency degree equal to or higher than the sum of their individual inconsistency degrees. However, this should not be taken as a rule. According to our first principle, the inconsistency degree of a knowledge base should be directly proportional to the number of minimal tableaux. It turns out that the union of knowledge bases does not accumulate their minimal tableaux. Consider, for example, the knowledge bases $\mathcal{K}_1 = \{a \wedge (a \wedge \neg a)\}$ and $\mathcal{K}_2 = \{a, \neg a\}$. Each of them presents only one minimal proof of inconsistency. Observe that, in both knowledge bases, the cause of inconsistency is the same: a and $\neg a$. For the knowledge base $\mathcal{K}_1$, we achieve this by decomposing the conjunctions, while in $\mathcal{K}_2$, this conflict is evident. Therefore, individually, $\mathcal{K}_1$ and $\mathcal{K}_2$ present inconsistency degree of 1. Thus, according to SA, $\mathcal{K}_1 \cup \mathcal{K}_2$ must have an inconsistency degree of at least 2. However, $\mathcal{K}_1 \cup \mathcal{K}_2$ presents only one minimal proof of inconsistency as well: the explicit conflict a and $\neg a$. Therefore, in all three measures we proposed, we have that $\mathcal{K}_1 \cup \mathcal{K}_2$ presents an inconsistency degree of 1 as well. It is clear that SA does not present a good behaviour for inconsistency measurement.

- MN and CD: The postulate MN states that all minimal inconsistent sets should have the same degree of inconsistency 1, while CD states that if every formula in a knowledge base $\mathcal{K}$ is inconsistent then the inconsistency degree of $\mathcal{K}$ must be 1. Both postulates are very prohibitive, as they do not allow grading neither minimal inconsistent sets nor sets containing only inconsistent formulae. If inconsistency in a minimal inconsistent set is much more apparent than in another minimal inconsistent set, then according to our two principles, it is plausible to grade the first one as more inconsistent than the second one. This argument also applies for bases with only inconsistent formulae. Such postulates, therefore, are too fragile to give a suitable notion of rationality for assessing inconsistencies.

- MI: this postulate says that if one can partition the set of minimal inconsistent subsets of a knowledge base $\mathcal{K}$ into two sets A and B then the inconsistency degree of $\mathcal{K}$ corresponds to the sum of the inconsistency degree of the knowledge base obtained from A and obtained from B. Similar to MN, this postulate disregards that the degree of inconsistency does not depend exclusively on the minimal inconsistent subsets. As our measures resort to minimal proofs, this postulate does not pose any criteria for assessing inconsistencies.

- AT and EC: these postulates state that the degree of minimal inconsistent sets should be graded according to the number of formulae in it. The size of the minimal inconsistent set, however, is not directly connected to the effort of proving that a knowledge base is inconsistent. Indeed, smaller inconsistent sets might present minimal proofs bigger than minimal proofs from greater sets (see proof of AT in Theorem 13, for an example).

- EX, SY and AI: Two bases can be logically equivalent but present different reasons of inconsistency, therefore since we are based on the effort of reasoning to measure inconsistency it is desirable that EX, SY and AI be violated.

	CO	NO	MO	IN	DO	NM	SD	SI	SA	PY	MI	MN	AT	EC	AC	CD	FD	SY	EX	AI
$\mathcal{I}^{\#}(\mathcal{K})$	✓	✗	✗	✗	✗	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
$\mathcal{I}^{\min}(\mathcal{K})$	✓	✓	✓	✗	✗	✓	✗	✓	✗	✗	✗	✗	✗	✗	✓	✗	✓	✗	✗	✗
$\mathcal{I}^{\sum}(\mathcal{K})$	✓	✗	✗	✗	✗	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗

Table 1: Compliance of $\mathcal{I}^{\min}$ with rationality postulates for inconsistency measures.

For our measures, we obtain the following.

Theorem 13. *The compliance of the measures $\mathcal{I}^{\#}$, $\mathcal{I}^{\min}$, and $\mathcal{I}^{\sum}$ with the rationality postulates is as presented in Table 1.*

The measures $\mathcal{I}^{\#}$ and $\mathcal{I}^{\sum}$ do not comply with the MO postulate, which is satisfied by several inconsistency measures in the literature. Indeed, according to our two principles, there are cases in which it is plausible to waive MO. For instance, consider the knowledge $\mathcal{K} = \{(a \vee b) \wedge (\neg a \wedge \neg b)\}$. This knowledge base has two minimal closed tableaux: the tableaux π_2 and π_3 depicted at Fig. 3.

By adding a to $\mathcal{K}$, we obtain the knowledge base $\mathcal{K}'$, which has only one minimal closed tableau (the tableau π_1 above). Therefore, according to our first principle, the inconsistency degree of $\mathcal{K}'$ must be smaller than the inconsistency degree of $\mathcal{K}$. Although this example works as an argument against MO, we argue that there are cases in which some form of monotonicity would still be desirable. For this same example, consider the formula $a \vee c$ and the knowledge base $\mathcal{K}'' = \mathcal{K} \cup \{a \vee c\}$. Observe that $a \vee c$ does not "participate" in making $\mathcal{K}''$ inconsistent, as it does not produce any new minimal proof of inconsistency. Towards this end, according to our both principles, the inconsistency degrees of $\mathcal{K}$ and $\mathcal{K}''$ should be the same. Therefore, for this specific example, some form of monotonicity should be preserved. Indeed, for all the three inconsistency measures we defined, $\mathcal{K}$ and $\mathcal{K}''$ present the same degree of inconsistency. But then, why adding $a \vee c$ should induce a monotonic behaviour, whilst adding a should not? In fact, if we inspect $a \vee c$ and a closer, we will see that a is partially "redundant" while $a \vee c$ is not "redundant". To be more precise, $\mathcal{K} \vdash a$, but $\mathcal{K} \not\vdash a \vee c$. Let us properly define our notion of partial redundancy:

Definition 14. *A formula α is* partially-redundant *in $\mathcal{K}$ iff there is some formula φ such that $\mathcal{K} \vdash \varphi$ and $\alpha \vdash \varphi$.*

$$\pi_1 = \begin{array}{c} \mathcal{K} \cup \{a\} \\ | \\ \mathcal{K} \cup \{a, a \vee b, \neg a \wedge \neg b\} \\ | \\ \mathcal{K} \cup \{a, a \vee b, \neg a \wedge \neg b, \neg a, \neg b\} \end{array}$$

$$\pi_2 = \begin{array}{cc} \multicolumn{2}{c}{\mathcal{K}} \\ \multicolumn{2}{c}{|} \\ \multicolumn{2}{c}{\mathcal{K} \cup \{a \vee b, \neg a \wedge \neg b\}} \\ \multicolumn{2}{c}{|} \\ \multicolumn{2}{c}{\mathcal{K} \cup \{a \vee b, \neg a \wedge \neg b, \neg a, \neg b\}} \\ / & \backslash \\ \mathcal{K} \cup \{a \vee b, \neg a \wedge \neg b, \neg a, \neg b, a\} & \mathcal{K} \cup \{a \vee b, \neg a \wedge \neg b, \neg a, \neg b, b\} \end{array}$$

$$\pi_3 = \begin{array}{cc} \multicolumn{2}{c}{\mathcal{K}} \\ \multicolumn{2}{c}{|} \\ \multicolumn{2}{c}{\mathcal{K} \cup \{a \vee b, \neg a \wedge \neg b\}} \\ / & \backslash \\ \mathcal{K} \cup \{a \vee b, \neg a \wedge \neg b, a\} & \mathcal{K} \cup \{a \vee b, \neg a \wedge \neg b, b\} \\ | & | \\ \mathcal{K} \cup \{a \vee b, \neg a \wedge \neg b, a, \neg a, \neg b\} & \mathcal{K} \cup \{a \vee b, \neg a \wedge \neg b, b, \neg a, \neg b\} \end{array}$$

Figure 3: Some minimal tableaux form knowledge bases $\mathcal{K} = \{(a \vee b) \wedge (\neg a \wedge \neg b)\}$ and $\mathcal{K}' = \mathcal{K} \cup \{a\}$.

Example 15. *Consider the knowledge base $\mathcal{K} = \{a \wedge b, \neg a\}$ and the formula $\alpha = (c \vee d)$. Observe that the only common information derived from each consistent subset of $\mathcal{K}$ and α are tautologies. This means that α has no partially-redundant information with $\mathcal{K}$, that is, $\mathcal{K}$ is not partially-redundant. This is because no tableaux of $\mathcal{K}$ shares formulae with any tableaux of $\{\alpha\}$. In fact, $\mathcal{K}$ and α have each one single tableau, as illustrated in Fig. 4, and neither has a single formula in common.*

In the following, we investigate a further (and new) postulate that describe our new

$$\pi_1 = \begin{array}{c} \{a \wedge b, \neg a\} \\ | \\ \{a \wedge b, \neg a, a, b\} \end{array} \qquad \pi_2 = \begin{array}{ccc} & \{c \vee d\} & \\ / & & \backslash \\ \{c \vee d, c\} & & \{c \vee d, d\} \end{array}$$

Figure 4: The only tableaux for knowledge base $\mathcal{K}$ and formula α from Example 15.

approaches and point to their specific advantages. In particular, if we restrict the addition of information to "non-redundant" information our measures do indeed behave monotonically: This monotonicity of non-redundant information is formalised as the

Non-redundant Monotonicity (**NM**): If ϕ is not partially-redundant in $\mathcal{K}$ then $\mathcal{I}(\mathcal{K}) \leq \mathcal{I}(\mathcal{K} \cup \{\phi\})$.

The above postulate demands that adding genuinely new information to a knowledge base cannot decrease the degree of inconsistency. Our three measures comply with this demand.

Theorem 16. *The inconsistency measures $\mathcal{I}^{\min}$, $\mathcal{I}^{\#}$ and $\mathcal{I}^{\sum}$ satisfy NM.*

In the analyses above, we have shown that our measures do not comply with the postulate MI. This occurs mainly because there is no correspondence between minimal inconsistent subsets and minimal tableaux, as Example 17 and Example 18 below illustrate.

Example 17. *Consider the knowledge base $\mathcal{K} = \{a \wedge c, (\neg a \vee d) \wedge (\neg c \vee d), \neg d\}$ and the following 2 minimal tableaux of this knowledge base:*

$$\pi_1 = \begin{array}{ccc} & \mathcal{K} & \\ & | & \\ & \mathcal{K} \cup \{a, b \wedge c\} & \\ & | & \\ & \mathcal{K} \cup \{a, c, \neg a \vee d, \neg c \vee d\} & \\ / & & \backslash \\ \mathcal{K} \cup \{a, c, \neg a \vee d, \neg c \vee d, \neg a\} & & \mathcal{K} \cup \{a, c, \neg a \vee d, \neg c \vee d, d\} \end{array}$$

$$\pi_2 = \begin{array}{c} \mathcal{K} \\ | \\ \mathcal{K} \cup \{a, b \wedge c\} \\ | \\ \mathcal{K} \cup \{a, c, \neg a \vee d, \neg c \vee d\} \\ \swarrow \qquad \searrow \\ \mathcal{K} \cup \{a, c, \neg a \vee d, \neg c \vee d, \neg c\} \quad \mathcal{K} \cup \{a, c, \neg a \vee d, \neg c \vee d, d\} \end{array}$$

In Example 17, the knowledge base $\mathcal{K}$ is a minimal inconsistent set and has at least two different minimal tableaux τ_1 and τ_2.

Example 18. *Let $\mathcal{K} = \{a, \neg a, b, \neg b, a \vee b\}$. Observe that this knowledge base has two minimal inconsistent subsets which are $A_1 = \{a, \neg a\}, A_2 = \{b, \neg b\}$. However, this knowledge base has only one minimal tableau which is*

$$\pi_3 = \quad \mathcal{K}$$

In Example 18, the minimal tableaux τ_3 is associated with the minimal inconsistent subsets A_1 and A_2, since the contradictions in the leaf node, which coincides with the root node, regard both A_1 and A_2: $\{a, \neg a\}$ and $\{b, \neg b\}$. Therefore, none of the three measures that we proposed are sensible to this interpretation of the number of sources of conflict. However, we can construct a measure that iteratively removes the sources of inconsistency based on the minimal tableaux, and accumulate the values, until no inconsistency is left. For example, as both A_1 and A_2 are related to τ_3, we can remove A_1 from $\mathcal{K}$ obtaining the knowledge base $\mathcal{K}' = \mathcal{K} \setminus A_1$. We then compute the minimal tableau of $\mathcal{K}'$ which contains only one node labelled with $\mathcal{K}'$. We then remove A_2 from it obtaining a consistent knowledge base. Therefore, in the end, we assign an inconsistency value of 2 to $\mathcal{K}$: since all three measure yield value 1 on both iterations.

5.2 Expressivity

Besides rationality postulates, another (complementary) dimension of evaluating an inconsistency measure is its *expressivity* [19], that is, the number of different inconsistency values a measure can attain on some certain sets of knowledge bases. This evaluation measure has been proposed in order to be able to distinguish trivial measures such as the drastic measure—which assigns 0 to consistent and 1 to inconsistent knowledge bases but still satisfies a reasonable number of rationality postulates—from more “fine-grained” assessments of inconsistency.

Before defining expressivity characteristics we need some further definitions.

$$\mathbb{K}^v(n) = \{\mathcal{K} \in \mathbb{K} \mid |\mathsf{At}(\mathcal{K})| \leq n\}$$
$$\mathbb{K}^f(n) = \{\mathcal{K} \in \mathbb{K} \mid |\mathcal{K}| \leq n\}$$
$$\mathbb{K}^l(n) = \{\mathcal{K} \in \mathbb{K} \mid \forall \phi \in \mathcal{K} : \mathtt{len}(\phi) \leq n\}$$
$$\mathbb{K}^p(n) = \{\mathcal{K} \in \mathbb{K} \mid \forall \phi \in \mathcal{K} : |\mathsf{At}(\phi)| \leq n\}$$

Informally speaking, $\mathbb{K}^v(n)$ is the set of all knowledge bases that mention at most n different propositions, $\mathbb{K}^f(n)$ is the set of all knowledge bases that contain at most n formulae, $\mathbb{K}^l(n)$ is the set of all knowledge bases that contain only formulae with maximal length n, and $\mathbb{K}^p(n)$ is the set of all knowledge bases that contain only formulae that mention at most n different propositions each.

Definition 19. *Let $\mathcal{I}$ be an inconsistency measure and $n > 0$. Let $\alpha \in \{v, f, l, p\}$. The* α-characteristic $\mathcal{C}^\alpha(\mathcal{I}, n)$ *of $\mathcal{I}$ w. r. t. n is defined as $\mathcal{C}^\alpha(\mathcal{I}, n) = |\{\mathcal{I}(\mathcal{K}) \mid \mathcal{K} \in \mathbb{K}^\alpha(n)\}|$.*

In other words, $\mathcal{C}^\alpha(\mathcal{I}, n)$ is the number of different inconsistency values $\mathcal{I}$ assigns to knowledge bases from $\mathbb{K}^\alpha(n)$.

The following results show that our new measures are maximally expressive w. r. t. all four expressivity characteristics.

Theorem 20. *For all $n > 0$ and $\mathcal{I} \in \{\mathcal{I}^{\min}, \mathcal{I}^{\#}, \mathcal{I}^{\sum}\}$, $\mathcal{C}^v(\mathcal{I}, n) = \mathcal{C}^f(\mathcal{I}, n) = \mathcal{C}^p(\mathcal{I}, n) = \infty$.*

Theorem 21.

1. *For all $n > 1$, $\mathcal{C}^l(\mathcal{I}^{\#}, n) = \infty$.*

2. *For all $n > 3$, and $\mathcal{I} \in \{\mathcal{I}^{\min}, \mathcal{I}^{\sum}\}$, $\mathcal{C}^l(\mathcal{I}, n) = \infty$.*

All three measures are maximally expressive. All three measures present infinitely many values for knowledge bases with at least one atomic propositional symbol, or knowledge bases with at least one formula. With respect to the length of the formulae in a knowledge base, the measure $\mathcal{I}^{\#}$ presents infinitely many values for knowledge bases containing formulae with length higher than one, while for the other two measures, for length higher than 3.

5.3 Computational complexity

In the following, we will (briefly) discuss computational complexity issues of our new measures.

Following [23], we consider the following problems. Let $\mathcal{I}$ be some inconsistency measure.

EXACT$_{\mathcal{I}}$ **Input**: $\mathcal{K} \in \mathbb{K}$, $x \in \mathbb{R}^{\infty}_{\geq 0}$
Output: TRUE iff $\mathcal{I}(\mathcal{K}) = x$

UPPER$_{\mathcal{I}}$ **Input**: $\mathcal{K} \in \mathbb{K}$, $x \in \mathbb{R}^{\infty}_{\geq 0}$
Output: TRUE iff $\mathcal{I}(\mathcal{K}) \leq x$

LOWER$_{\mathcal{I}}$ **Input**: $\mathcal{K} \in \mathbb{K}$, $x \in \mathbb{R}^{\infty}_{\geq 0} \setminus \{0\}$
Output: TRUE iff $\mathcal{I}(\mathcal{K}) \geq x$

VALUE$_{\mathcal{I}}$ **Input**: $\mathcal{K} \in \mathbb{K}$
Output: The value of $\mathcal{I}(\mathcal{K})$

The computational complexity of our new measures is tightly linked to the general area of *proof complexity* [4]. As there are exponential lower bounds on the size of a minimal tableau [2, 16], we cannot expect to provide membership results of any of the above computational problems to any (deterministic or non-deterministic) complexity class within the polynomial hierarchy. The most precise statement on all our measures we can make is the following.

Theorem 22. *For $\mathcal{I} \in \{\mathcal{I}^{\#}, \mathcal{I}^{\min}, \mathcal{I}^{\#}\}$,* EXACT$_{\mathcal{I}}$*,* UPPER$_{\mathcal{I}}$*, and* LOWER$_{\mathcal{I}}$ *are in EXPSPACE, while* VALUE$_{\mathcal{I}}$ *is in FEXPSPACE (the functional variant of EXPSPACE).*

It is possible that the above bound could be improved to EXPTIME as it may not be necessary to explicitly write down every (potential) tableau (but note that whether EXPTIME$\neq$EXPSPACE is also an open question). However, without a proof system that exhibits minimal proofs of polynomial length for all contradictions, EXPTIME is a necessary lower bound. This fact establishes our three measures to be the hardest inconsistency measures among the ones investigated in [23].

6 Summary and Conclusion

In this paper, we proposed novel approaches to measure inconsistency in knowledge bases. Our approaches are based on the notion of minimal closed tableaux, and we analysed the behaviour of these novel inconsistency measures in terms of rationality postulates, expressivity and computational complexity. The central idea of our approaches is to measure inconsistency via measuring *proof complexity*, i. e. the easier it is for a reasoner to detect inconsistency, the larger the inconsistency is to be regarded.

Using tableaux methods for constructing inconsistency measurement is novel, but [13] uses a different notion of *proof* to define an inconsistency measure. There, instead of minimal tableaux a *minimal proof* is a (not necessarily consistent) subset of the knowledge base that entails some formula and inconsistency is measured by appropriately aggregating the number of proofs of complementary literals. However, this measure makes no use of proof systems in our sense and it has also been shown in [20] that it does not satisfy CO and should therefore not be regarded as a meaningful inconsistency measure. Inconsistency measures based on conflicting variables were proposed in [12]. In their measure, the inconsistency value of a knowledge base $\mathcal{K}$ corresponds to the ratio between the conflicting variables and all the variables of $\mathcal{K}$. This focus on variables makes their measure to plateau when the addition/removal of formulae does not change the amount of conflicting variables. Consider, for example, the knowledge bases $\mathcal{K} = \{a \wedge b, \neg a \vee \neg b\}$ and $\mathcal{K}' = \mathcal{K} \cup \{a\}$. As $\mathcal{K}'$ contains more conflicting sources of inconsistencies than $\mathcal{K}$ (two minimal inconsistencies sets against one minimal inconsistent set), it would be rational to assess $\mathcal{K}'$ as more inconsistent than $\mathcal{K}$. However, the measure based on conflicting variables will assess both as equally inconsistent as they contain the same number of conflicting variables. All our three measures will assess both knowledge bases differently.

Our measures provide a new completely syntactical approach to inconsistency measurement that feature maximal expressivity in differentiating inconsistent knowledge bases (see Section 5.2). However, their computational complexity is a significant challenge for their applicability. Future work is about devising (approximate) algorithmic solutions to overcome this barrier.

Acknowledgements. This research is supported by the German Research Association (DFG), project number 465447331.

References

[1] Meriem Ammoura, Yakoub Salhi, Brahim Oukacha, and Badran Raddaoui. On an MCS-based inconsistency measure. *International Journal of Approximate Reasoning*, 80:443–459, 2017.

[2] Noriko H. Arai, Toniann Pitassi, and Alasdair Urquhart. The complexity of analytic tableaux. In *Proceedings on 33rd Annual ACM Symposium on Theory of Computing*, pages 356–363. ACM, 2001.

[3] Philippe Besnard. Revisiting postulates for inconsistency measures. In *Proceedings of the 14th European Conference on Logics in Artificial Intelligence (JELIA'14)*, pages 383–396, 2014.

[4] Stephen Cook and Phuong Nguyen. *Logical foundations of proof complexity*, volume 11. Cambridge University Press Cambridge, 2010.

[5] G Frege. *Begriffsschrift.* Verlag von Louis Nebert, 1879.

[6] Gerhard Gentzen. Untersuchungen über das logische schließen. i. *Mathematische zeitschrift*, 35, 1935.

[7] John Grant. Classifications for inconsistent theories. *Notre Dame Journal of Formal Logic*, 19(3):435–444, 1978.

[8] John Grant and Anthony Hunter. Measuring consistency gain and information loss in stepwise inconsistency resolution. In W. Liu, editor, *Proceedings of the 11th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2011)*, number 6717 in Lecture Notes in Artificial Intelligence, pages 362–373. Springer-Verlag, 2011.

[9] Anthony Hunter and Sébastien Konieczny. Approaches to measuring inconsistent information. In *Inconsistency Tolerance*, volume 3300 of *Lecture Notes in Computer Science*, pages 189–234. Springer, 2004.

[10] Anthony Hunter and Sébastien Konieczny. Shapley inconsistency values. In *Proceedings of the 10th International Conference on Knowledge Representation (KR'06)*, pages 249–259. AAAI Press, 2006.

[11] Anthony Hunter and Sébastien Konieczny. Measuring inconsistency through minimal inconsistent sets. In Gerhard Brewka and Jérôme Lang, editors, *Proceedings of the Eleventh International Conference on Principles of Knowledge Representation and Reasoning (KR'2008)*, pages 358–366, Sydney, Australia, September 2008. AAAI Press, Menlo Park, California.

[12] Saïd Jabbour, Yue Ma, Badran Raddaoui, and Lakhdar Sais. Quantifying conflicts in propositional logic through prime implicates. *Int. J. Approx. Reason.*, 89:27–40, 2017.

[13] Said Jabbour and Badran Raddaoui. Measuring inconsistency through minimal proofs. In *Proceedings of the 12th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, ECSQARU'13, pages 290–301, Berlin, Heidelberg, 2013. Springer-Verlag.

[14] Said Jabbour and Lakhdar Sais. Exploiting mus structure to measure inconsistency of knowledge bases. In *Proceedings of the 22nd European Conference on Artificial Intelligence (ECAI'16)*, pages 991–998, 2016.

[15] Henry E. Kyburg. *Probability and the Logic of Rational Belief.* Wesleyan University Press, 1961.

[16] Neil V Murray and Erik Rosenthal. On the computational intractability of analytic tableau methods. *Logic Journal of the IGPL*, 2(2):205–228, 1994.

[17] Raymond M Smullyan. *First-order logic*. Courier Corporation, 1995.

[18] Matthias Thimm. Inconsistency measures for probabilistic logics. *Artificial Intelligence*, 197:1–24, April 2013.

[19] Matthias Thimm. On the expressivity of inconsistency measures. *Artificial Intelligence*, 234:120–151, February 2016.

[20] Matthias Thimm. Measuring inconsistency with many-valued logics. *International Journal of Approximate Reasoning*, 86:1–23, July 2017.

[21] Matthias Thimm. On the compliance of rationality postulates for inconsistency measures: A more or less complete picture. *Künstliche Intelligenz*, 31(1):31–39, March 2017.

[22] Matthias Thimm. On the evaluation of inconsistency measures. In John Grant and Maria Vanina Martinez, editors, *Measuring Inconsistency in Information*, volume 73 of *Studies in Logic*. College Publications, February 2018.

[23] Matthias Thimm and Johannes Peter Wallner. Some complexity results on inconsistency measurement. In *Proceedings of the 15th International Conference on Principles of Knowledge Representation and Reasoning (KR'16)*, pages 114–123, April 2016.

Received \jreceived

A Proofs for Section 5 (Analysis)

The proofs presented in this appendix include set-labelled tableaux in which the set of formulae labelled to the nodes is significantly big. For clarity, we will depict tableaux in a concise way. We will draw the root now with the whole knowledge base. However, for other nodes, instead of presenting the whole set of formulae labelled in a node, we draw only the fresh formulae added from the parent node to the child node (as illustrated in Fig. 5). The set of formulae labelled on a node n can be inferred by taking the union of the formulae from the root node to n.

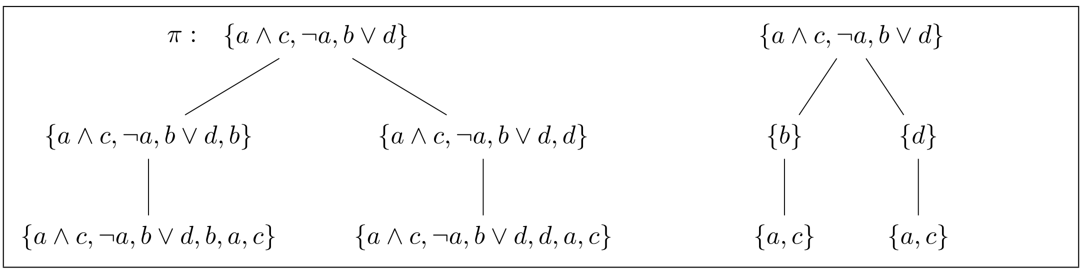

Figure 5: On the left, a closed set-labelled tableau π for the knowledge base $\mathcal{K} = \{a \wedge c, \neg a, b \vee d\}$. On the right, the concise way of representing the set-labelled tableaux π.

Lemma A.1. *If $\pi = (N, E, \lambda)$ is a tableau for a knowledge base $\mathcal{K}$ then $\mathsf{At}(\lambda(n)) = \mathsf{At}(\mathcal{K})$ for every $n \in N$.*

Proof. By induction on the level of n

Base: $level(n)$ is zero, that is, n is the root. Then $\lambda(n) = \mathcal{K}$. Thus, $\mathsf{At}(\lambda(n)) = \mathsf{At}(\mathcal{K})$.

Induction Hypothesis (IH): if $level(n') < level(n)$ then $\mathsf{At}(\lambda(n')) = \mathsf{At}(\mathcal{K})$.

Induction Step: $level(n) > 1$. Thus, n has some parent n', and either (1) $\mathsf{children}(n') = \{n\}$ or (2) $\mathsf{children}(n') = \{n, n_2\}$ with $n \neq n_2$:

(1) $\mathsf{children}(n') = \{n\}$. Then, $\lambda(n) = \lambda(n) \cup A$, where one of the following cases hold:

1. $A = \{\varphi\}$, with $\neg\neg\varphi \in \lambda(n)$. Thus, as $\mathsf{At}(\varphi) = \mathsf{At}(\neg\neg\varphi)$, we get that $\mathsf{At}(\lambda(n')) = \mathsf{At}(\lambda(n))$. Thus, it follows from HI, that $\mathsf{At}(\lambda(n)) = \mathsf{At}(\mathcal{K})$.
2. $A = \{\varphi, \psi\}$ with $\varphi \wedge \psi \in \lambda(n')$. Thus, as $\mathsf{At}(\varphi \wedge \psi) = \mathsf{At}(\varphi) \cup \mathsf{At}(\psi)$, we get that $\mathsf{At}(\lambda(n')) = \mathsf{At}(\lambda(n))$. Thus, it follows from HI, that $\mathsf{At}(\lambda(n)) = \mathsf{At}(\mathcal{K})$.

Received \jreceived

3. $A = \{\neg\varphi \vee \neg\psi\}$ with $\neg(\varphi \wedge \psi) \in \lambda(n')$. Thus, as $\mathsf{At}(\neg(\varphi \wedge \psi)) = \mathsf{At}(\neg\varphi \vee \neg\psi)$, we get that $\mathsf{At}(\lambda(n')) = \mathsf{At}(\lambda(n))$. Thus, it follows from HI, that $\mathsf{At}(\lambda(n)) = \mathsf{At}(\mathcal{K})$.
4. $A = \{\neg\varphi \wedge \neg\psi\}$ with $\neg(\varphi \vee \psi) \in \lambda(n')$. Thus, as $\mathsf{At}(\neg(\varphi \vee \psi)) = \mathsf{At}(\neg\varphi \wedge \neg\psi)$, we get that $\mathsf{At}(\lambda(n')) = \mathsf{At}(\lambda(n))$. Thus, it follows from HI, that $\mathsf{At}(\lambda(n)) = \mathsf{At}(\mathcal{K})$.

(2) $\mathsf{children}(n') = \{n, n_2\}$ with $n \neq n_2$. Thus, there is $\varphi \vee \psi \in \lambda(n')$ such that either (a) $\lambda(n) = \lambda(n') \cup \{\varphi\}$ or (b) $\lambda(n) = \lambda(n') \cup \{\psi\}$. Observe that $\mathsf{At}(\varphi) \subseteq \mathsf{At}(\varphi \vee \psi)$ and $\mathsf{At}(\psi) \subseteq \mathsf{At}(\varphi \vee \psi)$. Therefore in either cases (a or b), we get that $\mathsf{At}(n) = \mathsf{At}(n')$. Thus, it follows from HI, that $\mathsf{At}(\lambda(n)) = \mathsf{At}(\mathcal{K})$.

□

Theorem 13. *The compliance of the measures $\mathcal{I}^{\#}$, $\mathcal{I}^{\min}$, and $\mathcal{I}^{\sum}$ with the rationality postulates is as presented in Table 1.*

Proof. In the following, we denote by +X a proof that shows that property X is satisfied and by −X a proof that shows that property X is violated.

+CO Let $\mathcal{K}$ be a knowledge base. $\mathcal{K}$ is inconsistent if and only if there is a closed tableau π. Then, $\mathcal{I}^{\#}(\mathcal{K}) = 0$ if and only if $\mathcal{K}$ is inconsistent. Analogously, $\mathcal{I}^{\min}(\mathcal{K}) = 0$ if and only if $\mathcal{K}$ is inconsistent; and $\mathcal{I}^{\sum}(\mathcal{K}) = 0$ if and only if $\mathcal{K}$ is inconsistent

NO The measures $\mathcal{I}^{\#}$ and $\mathcal{I}^{\sum}$ clearly fail NO, while $\mathcal{I}^{\min}$ satisfies it.

+ By definition $\mathcal{I}^{\min}(\mathcal{K}) = 0$, if $\mathcal{K}$ is consistent, and corresponds $\mathcal{I}^{\min}(\mathcal{K}) = 1/n$, where n is the size of the minimal closed tableaux in $\mathcal{T}_{\perp}(\mathcal{K})$. Therefore, $0 \leq \mathcal{I}^{\min}(\mathcal{K}) \leq 1$.
− Consider the following knowledge base $\mathcal{K} = \{a \wedge \neg a, b \wedge \neg b, c \wedge \neg c\}$. For this knowledge base, there are only three minimal closed tableaux, all of them of size 2. Therefore, $\mathcal{I}^{\#}(\mathcal{K}) = 3$, and $\mathcal{I}^{\sum}(\mathcal{K}) = \frac{3}{2} > 1$.

MO The measures $\mathcal{I}^{\#}$ and $\mathcal{I}^{\sum}$ clearly fail MO, while $\mathcal{I}^{\min}$ satisfies it.

+ Note that if π is a closed tableau in $\mathcal{K}$ then π is also a closed tableau in $\mathcal{K}'$ for $\mathcal{K} \subseteq \mathcal{K}'$. Therefore, the length of a minimal closed tableau can only decrease when adding information, thus $\mathcal{I}^{\min}$ can only increase.
− Let $\mathcal{K} = \{(a \vee b) \wedge (\neg a \wedge \neg b), a\}$.

Received \jreceived

There is only one minimal closed tableau for $\mathcal{K}$, which is π_1 below. On the other hand, there are two minimal closed tableau for $\mathcal{K} \setminus \{a\}$, which are π_2 and π_3 below (depicted in the concise form). We have

$$\begin{array}{lcl} \mathcal{I}^{\#}(\mathcal{K}) = 1 & < & \mathcal{I}^{\#}(\mathcal{K} \setminus \{a\}) = 2 \\ \mathcal{I}^{\sum}(\mathcal{K}) = 1/3 & < & \mathcal{I}^{\sum}(\mathcal{K} \setminus \{a\}) = 1/5 + 1/6 = 11/30 \end{array}$$

$$\pi_1 = \begin{array}{c} \mathcal{K} \\ | \\ \{a \vee b, \neg a \wedge \neg b\} \\ | \\ \{\neg a, \neg b\} \end{array} \quad \pi_2 = \begin{array}{c} \mathcal{K}\setminus\{a\} \\ | \\ \{a \vee b, \neg a \wedge \neg b\} \\ | \\ \{\neg a, \neg b\} \\ \diagup \quad \diagdown \\ \{a\} \quad \{b\} \end{array}$$

$$\pi_3 = \begin{array}{c} \mathcal{K}\setminus\{a\} \\ | \\ \mathcal{K} \cup \{a \vee b, \neg a \wedge \neg b\} \\ \diagup \qquad \diagdown \\ \begin{array}{cc} \{a\} & \{b\} \\ | & | \\ \{\neg a, \neg b\} & \{\neg a, \neg b\} \end{array} \end{array}$$

–IN Consider the counterexample for MO. Recall $\mathcal{K} = \{(a \vee b) \wedge (\neg a \wedge \neg b), a\}$. Observe that a is free, and $\mathcal{I}^{\#}(\mathcal{K}) \neq \mathcal{I}^{\#}(\mathcal{K}\setminus\{a\}))$, $\mathcal{I}^{\min}(\mathcal{K}) \neq \mathcal{I}^{\min}(\mathcal{K}\setminus\{a\}))$ and $\mathcal{I}^{\sum}(\mathcal{K}) \neq \mathcal{I}^{\sum}(\mathcal{K} \setminus \{a\}))$.

–DO Let $\mathcal{K} = \{\neg a, \neg b, \neg c\}$, and formulae $\alpha = a$ and $\beta = (a \vee b) \wedge (a \vee c)$. Note that $\mathbb{K} \equiv \beta$. The knowledge base $\mathcal{K} \cup \{\alpha\}$ has only one closed tableau which has size 1 ($\pi_1 = \mathcal{K} \cup \{\alpha\}$), while $\mathcal{K} \cup \{\beta\}$ has two closed tableaux (π_1 and π_2 below), both with size 4. Thus, $\mathcal{I}^{\#}(\mathcal{K} \cup \{\alpha\}) = 1$, $\mathcal{I}^{\#}(\mathcal{K} \cup \{\beta\}) = 2$, $\mathcal{I}^{\min}(\mathcal{K} \cup \{\alpha\}) = 1$,

Received \jreceived

$\mathcal{I}^{\min}(\mathcal{K} \cup \{\beta\}) = 1/4, \mathcal{I}^{\sum}(\mathcal{K} \cup \{\alpha\}) = 1, \mathcal{I}^{\sum}(\mathcal{K} \cup \{\beta\}) = 2/4 = 1/2.$

$$\pi_2 = \begin{array}{c} \mathcal{K} \cup \{\beta\} \\ | \\ \mathcal{K} \cup \{(a \vee b), (a \vee c), \beta\} \\ \swarrow \qquad \searrow \\ \mathcal{K} \cup \{a, (a \vee b), (a \vee c), \beta\} \quad \mathcal{K} \cup \{b, (a \vee b), (a \vee c), \beta\} \end{array}$$

$$\pi_3 = \begin{array}{c} \mathcal{K} \cup \{\beta\} \\ | \\ \mathcal{K} \cup \{(a \vee b), (a \vee c), \beta\} \\ \swarrow \qquad \searrow \\ \mathcal{K} \cup \{a, (a \vee b), (a \vee c), \beta\} \quad \mathcal{K} \cup \{c, (a \vee b), (a \vee c), \beta\} \end{array}$$

+SI Let α be a safe-formula in $\mathcal{K}$. From Proposition A.9, we have that α is non-redundant with $\mathcal{K} \setminus \{\alpha\}$. Thus α is consistent and not-redundant in $\mathcal{K} \setminus \{\alpha\}$. Thus, from Theorem A.14, $\mathcal{T}_{\perp}^{min}(K) = \bigcup_{\pi \in \mathcal{T}_{\perp}^{min}(\mathcal{K} \setminus \alpha)} \pi[\alpha]$. This implies that $\mathcal{I}(\mathcal{K}) = \mathcal{I}(\mathcal{K} \setminus \{\alpha\})$, for all three measures.

−SA Let $\mathcal{K} = \{a \wedge (b \wedge \neg b)\}$ and $\mathcal{K}' = \{a, \neg a\}$. Note that both $\mathcal{K}$ and $\mathcal{K}'$ have only one tableau (π_1 below, and $\mathcal{K}'$ also has only one closed tableau which is the tableau π' with only the root node labelled with $\mathcal{K}'$ itself. Moreover, $\mathcal{K} \cup \mathcal{K}'$ has only one tableau: the tableau with only the root node. Thus,

	$\mathcal{K}$	$\mathcal{K}'$	$\mathcal{K} \cup \mathcal{K}'$	$\mathcal{I}(\mathcal{K}) + \mathcal{I}(\mathcal{K}')$
$\mathcal{I}^{\#}$	1	1	1	2
$\mathcal{I}^{\min}$	1/4	1	1	4/3
$\mathcal{I}^{\sum}$	1/4	1	1	4/3

$$\pi_1 = \begin{array}{c} \mathcal{K} \\ | \\ \mathcal{K} \cup \{a, b \wedge \neg b\} \\ | \\ \mathcal{K} \cup \{a, \neg b, b, b \wedge \neg a\} \end{array}$$

Received \jreceived

–PY . Let $\mathcal{K} = \{a, \neg a, a \wedge b\}$. Observe that $\mathsf{MI}(\mathcal{K}) = \{\{a, \neg a\}, \{a \wedge b, \neg a\}.\}$. Thus, $a \wedge b$ is not free. However, both $\mathcal{K}$ and $\mathcal{K}' = \mathcal{K} \setminus \{a \wedge b\}$ have only one minimal closed tableau each: $\mathcal{K}$ and $\mathcal{K}'$, respectively. Thus penalty is violated for all three measures.

–MI . Let $\mathcal{K} = \{\neg a, a \wedge b\}$ and $\mathcal{K}' = \{a \wedge b, (\neg a \wedge b) \wedge c\}$. Note that $\mathsf{MI}(\mathcal{K}) = \{\mathcal{K}\}$, $\mathsf{MI}(\mathcal{K}') = \{\mathcal{K}'\}$, and $\mathsf{MI}(\mathcal{K} \cup \mathcal{K}') = \{\mathcal{K}, \mathcal{K}'\}$. Thus, $\mathsf{MI}(\mathcal{K}) \cap \mathsf{MI}(\mathcal{K}') = \emptyset$ and $\mathsf{MI}(\mathcal{K}) \cup \mathsf{MI}(\mathcal{K}') = \mathsf{MI}(\mathcal{K} \cup \mathcal{K}')$. The minimal closed tableau of $\mathcal{K}$ is π_1, the minimal closed tableau of $\mathcal{K}'$ is π_2 and the minimal closed tableau of $\mathcal{K} \cup \mathcal{K}'$ is π_3. All of them are shown below. Thus,

	$\mathcal{K}$	$\mathcal{K}'$	$\mathcal{K} \cup \mathcal{K}'$	$\mathcal{I}(\mathcal{K}) + \mathcal{I}(\mathcal{K}')$
$\mathcal{I}^{\#}$	1	1	1	2
$\mathcal{I}^{\min}$	1/2	1/3	1/2	1/2 + 1/3
$\mathcal{I}^{\Sigma}$	1/2	1/3	1/2	1/2 +1/3

$$\pi_1 = \begin{array}{c} \mathcal{K} \\ | \\ \mathcal{K} \cup \{a, b\} \end{array} \qquad \pi_2 = \begin{array}{c} \mathcal{K}\text{'} \\ | \\ \mathcal{K}\text{'} \cup \{c, \neg a \wedge b\} \\ | \\ \mathcal{K}\text{'} \cup \{c, \neg a, b\} \end{array} \qquad \pi_3 = \begin{array}{c} \mathcal{K} \cup \mathcal{K}' \\ | \\ \mathcal{K} \cup \mathcal{K}' \cup \{a, b\} \end{array}$$

–MN Let $\mathcal{K} = \{\neg a \wedge (\neg b \wedge \neg c), (a \vee b) \wedge (a \vee c)\}$. Note that $\mathcal{K} \in \mathsf{MI}(\mathcal{K})$. The minimal closed tableaux of $\mathcal{K}$ are π_1 and π_2 below.

Thus, $\mathcal{I}^{\#}(\mathcal{K}) = 2$, $\mathcal{I}^{\min}(\mathcal{K}) = 1/6$ and $\mathcal{I}^{\Sigma}(\mathcal{K}) = 2 \cdot 1/6 = 1/3$.

$\pi_1 =$

$$\begin{tikzcd}[column sep=tiny]
& \mathcal{K} \arrow[d, dash] & \\
& \mathcal{K} \cup \{\neg a, \neg b \wedge \neg c\} \arrow[d, dash] & \\
& \mathcal{K} \cup \{\neg a, \neg b, \neg c, \neg b \wedge \neg c\} \arrow[d, dash] & \\
& \mathcal{K} \cup \{a \vee b, a \vee c, \neg a, \neg b, \neg c, \neg b \wedge \neg c\} \arrow[dl, dash] \arrow[dr, dash] & \\
\mathcal{K} \cup \{a, a \vee b, a \vee c, \neg a, \neg b, \neg c, \neg b \wedge \neg c\} & & \mathcal{K} \cup \{b, a \vee b, a \vee c, \neg a, \neg b, \neg c, \neg b \wedge \neg c\}
\end{tikzcd}$$

Received \jreceived

$\pi_2 =$

$$\begin{array}{c}
\mathcal{K} \\
| \\
\mathcal{K} \cup \{\neg a, \neg b \wedge \neg c\} \\
| \\
\mathcal{K} \cup \{\neg a, \neg b, \neg c, \neg b \wedge \neg c\} \\
| \\
\mathcal{K} \cup \{a \vee b, a \vee c, \neg a, \neg b, \neg c, \neg b \wedge \neg c\} \\
\diagup \qquad \diagdown \\
\mathcal{K} \cup \{a, a \vee b, a \vee c, \neg a, \neg b, \neg c, \neg b \wedge \neg c\} \qquad \mathcal{K} \cup \{c, a \vee b, a \vee c, \neg a, \neg b, \neg c, \neg b \wedge \neg c\}
\end{array}$$

−AT Let $\mathcal{K} = \{a \wedge (\neg a \wedge \neg b), a, \neg a\}$, $M = \{a \wedge (\neg a \wedge \neg b)\}$ and $M' = \{a, \neg a\}$. Observe that $M, M' \in \mathsf{MI}(\mathcal{K})$ and $|M| < |M'|$. The only closed tableau of M is π_1, and $\pi_2 = M'$ is the only proof of closed tableau of M'. Thus,

	M	M'
$\mathcal{I}^{\#}$	1	1
$\mathcal{I}^{\min}$	1/3	1
$\mathcal{I}^{\Sigma}$	1/3	1

$$\pi_1 = \begin{array}{c}
M \\
| \\
M \cup \{a, \neg a \wedge b\} \\
| \\
M \cup \{a, \neg a, b, \neg a \wedge b\}
\end{array}$$

−EC Let $\mathcal{K} = \{(a \wedge (b \wedge c)) \wedge (\neg a \vee \neg b) \wedge (\neg a \vee \neg c)\}$. It has only two closed tableaux, π_1 and π_2 below. Thus, $\mathcal{I}^{\#}(\mathcal{K}) = 2$, $\mathcal{I}^{\min}(\mathcal{K}) = 1/7$, and $\mathcal{I}^{\Sigma}(\mathcal{K}) = 1/7$

Below for clarity, we do not draw the whole sets in each node, but instead, only the fresh formulae just added.

Received \jreceived

$\pi_1 =$

$$\mathcal{K}$$
$$|$$
$$\mathcal{K} \cup \{(a \wedge (b \wedge c)), (\neg a \vee \neg b) \wedge (\neg a \vee \neg c)\}$$
$$|$$
$$\mathcal{K} \cup \{a, b \wedge c, (\neg a \vee \neg b) \wedge (\neg a \vee \neg c)\}$$
$$|$$
$$\mathcal{K} \cup \{a, b, c, (\neg a \vee \neg b) \wedge (\neg a \vee \neg c)\}$$
$$|$$
$$\mathcal{K} \cup \{a, b, c, (\neg a \vee \neg b), (\neg a \vee \neg c)\}$$
$$\diagup \qquad \diagdown$$
$$\mathcal{K} \cup \{\neg a, a, b, c, (\neg a \vee \neg b), (\neg a \vee \neg c)\} \qquad \mathcal{K} \cup \{\neg b, a, b, c, (\neg a \vee \neg b), (\neg a \vee \neg c)\}$$

$\pi_2 =$

$$\mathcal{K}$$
$$|$$
$$\mathcal{K} \cup \{(a \wedge (b \wedge c)), (\neg a \vee \neg b) \wedge (\neg a \vee \neg c)\}$$
$$|$$
$$\mathcal{K} \cup \{a, b \wedge c, (\neg a \vee \neg b) \wedge (\neg a \vee \neg c)\}$$
$$|$$
$$\mathcal{K} \cup \{a, b, c, (\neg a \vee \neg b) \wedge (\neg a \vee \neg c)\}$$
$$|$$
$$\mathcal{K} \cup \{a, b, c, (\neg a \vee \neg b), (\neg a \vee \neg c)\}$$
$$\diagup \qquad \diagdown$$
$$\mathcal{K} \cup \{\neg a, a, b, c, (\neg a \vee \neg b), (\neg a \vee \neg c)\} \qquad \mathcal{K} \cup \{\neg c, a, b, c, (\neg a \vee \neg b), (\neg a \vee \neg c)\}$$

+-AC The inconsistency measures $\mathcal{I}^{\#}$ and $\mathcal{I}^{\Sigma}$ violates AC.

– $\mathcal{I}^{\#}$. Consider the sequence M_i, $i \in \mathbb{N}$ of minimal inconsistent sets given via

$$M_i = \{a_1, \ldots, a_i, \neg a_1 \vee (\neg a_2 \vee (\ldots \vee \neg a_i) \ldots)\}$$

We have $\lim_{i \to \infty} |M_i| = \infty$. Observe that each M_i has only one minimal closed tableau. Thus, $\mathcal{I}^{\#}(M_i) = 1$, which means $\lim_{i \to \infty} \mathcal{I}^{\#}(M_i) = 1$,

Received \jreceived

– $\mathcal{I}^{\sum}$. Consider the sequence M_i, $i \in \mathbb{N}$ of minimal inconsistent sets given via

$$
\begin{aligned}
&M_1 = \{\neg a_1, a_1\}\\
&M_2 = \{\neg a_1, \neg a_2, (a_1 \vee a_2) \wedge (a_2 \vee a_1)\}\\
&M_3 = \{\neg a_1, \neg a_2, \neg a_3, (a_1 \vee a_2) \wedge (a_2 \vee a_3) \wedge (a_3 \vee a_1)\}\\
&\dots\\
&M_i = \{\neg a_1, \neg a_2, \dots, \neg a_i, (a_1 \vee a_2) \wedge (a_2 \vee a_3) \wedge \dots \wedge (a_i \vee a_1)\}\\
&\dots
\end{aligned}
$$

Each M_i has exactly i minimal closed tableau. The M_1 has one with size one, M_2 has two, each with size 4. For the following ones, we can enumerate their minimal tableaux in the following way. The M_i has $i-2$ minimal tableaux, such that their sizes correspond exactly to the size of the tableaux of M_{i-1}, while the last 2 minimal tableaux have size $i+2$. In summary, (the number between commas represents the size of each tableau).

$$
\begin{aligned}
&M_1 = 1\\
&M_2 = 2 \cdot 4\\
&M_3 = 4, 2 \cdot (3+2)\\
&M_4 = 4, (3+2), 2 \cdot (4+2)\\
&M_5 = 4, (3+2), (4+2), 2 \cdot (5+2)\\
&\dots\\
&M_i = 4, (3+2), (4+2), \dots (i-1+2), 2 \cdot (i+2)
\end{aligned}
$$

Thus,

$$
\begin{aligned}
&\mathcal{I}^{\sum}(M_1) = 1\\
&\mathcal{I}^{\sum}(M_2) = \frac{2}{4} = \frac{1}{2}\\
&\mathcal{I}^{\sum}(M_3) = \frac{1}{4} + \frac{2}{(3+2)}\\
&\quad\dots\\
&\mathcal{I}^{\sum}(M_i) = \frac{1}{4} + \frac{1}{(3+2)} + \frac{1}{(4+2)} + \dots + \frac{1}{(i-1+2)} + \frac{2}{(i+2)}
\end{aligned}
$$

Received \jreceived

Note that $\mathcal{I}^{\sum}(M_i) < \mathcal{I}^{\sum}(M_{i+1})$. Thus, $\lim\limits_{i\to\infty} \mathcal{I}^{\sum}(M_i) = \infty$.

+ $\mathcal{I}^{\min}$. Let M_i be a minimal inconsistent set, and π one of its minimal tableau. Observe that, due to the sub-formulae derivation structure of the tableau, $|\pi| \geq |M_i|$. Thus, the bigger is the set, the bigger is the tableau, which means that the smaller is its inconsistent value according to $\mathcal{I}^{\min}$. Therefore, for an infinity sequence of minimal inconsistent sets, if $\lim\limits_{i\to\infty} |M_i| = \infty$, then $\lim\limits_{i\to\infty} \mathcal{I}^{\min}(M_i) = 0$.

−CD . Let $\mathcal{K} = \{a \wedge \neg a\}$. The knowledge base has only one minimal closed tableau and its size is 2. Thus, $\mathcal{I}^{\min}(\mathcal{K}) = \mathcal{I}^{\sum}(\mathcal{K}) = 1/2$. For $\mathcal{I}^{\min}$, let $\mathcal{K}' = \{\neg a \wedge (a \vee b) \wedge (a \vee \neg b) \wedge (a \vee c) \wedge (a \vee \neg c)\}$. Observe that $\mathcal{K}'$ has two minimal closed tableaux. Therefore, $\mathcal{I}^{\#}(\mathcal{K}) = 2$.

FD − $\mathcal{I}^{\#}$ and $\mathcal{I}^{\sum}$. See counterexample for MO.

+ $\mathcal{I}^{\min}$. If follows from MO.

−SY . Let $\mathcal{K} = \{a, \neg a\}$ and $\mathcal{K}' = \{(a \vee b) \wedge (a \vee \neg b) \wedge (a \vee c) \wedge (a \vee \neg c), \neg a\}$. Note that $\mathcal{K} \equiv^s \mathcal{K}'$ Observe that $\mathcal{K}$ has only one closed tableau which is $\mathcal{K}$, while $\mathcal{K}'$ has two closed tableaux: π_1 and π_2 below. Thus, $\mathcal{I}^{\#}(\mathcal{K}) = 1$, $\mathcal{I}^{\#}(\mathcal{K}') = 2$, $\mathcal{I}^{\min}(\mathcal{K}) = 1$, $\mathcal{I}^{\min}(\mathcal{K}') = 1/7$ and $\mathcal{I}^{\sum}(\mathcal{K}) = 1$, $\mathcal{I}^{\sum}(\mathcal{K}') = 1/7$. Below for clarity, we do not draw the whole sets in each node, but instead, only the fresh formulae just added.

$$\pi_1 = \begin{array}{c}
\mathcal{K} \\
| \\
\mathcal{K} \cup \{(a \vee b) \wedge (a \vee \neg b), \} \\
| \\
\mathcal{K} \cup \{(a \vee b), (a \vee \neg b)\} \\
\swarrow \qquad \searrow \\
\mathcal{K} \cup \{a, (a \vee \neg b), (a \vee c) \wedge (a \vee \neg c)\} \qquad \mathcal{K} \cup \{b, (a \vee \neg b)\} \\
\qquad\qquad\qquad\qquad\qquad\qquad \swarrow \qquad \searrow \\
\qquad\qquad\qquad\qquad \mathcal{K} \cup \{a, b, (a \vee \neg b)\} \qquad \mathcal{K} \cup \{b, \neg b, (a \vee \neg b)\}
\end{array}$$

Received \jreceived

$$
\pi_2 = \quad
\begin{array}{c}
\mathcal{K} \\
| \\
\mathcal{K} \cup \{(a \vee b) \wedge (a \vee \neg b), (a \vee c) \wedge (a \vee \neg c)\} \\
| \\
\mathcal{K} \cup \{(a \vee b) \wedge (a \vee \neg b), (a \vee c), (a \vee \neg c)\} \\
\swarrow \qquad \searrow \\
\mathcal{K} \cup \{a, (a \vee c), (a \vee \neg c)\} \qquad \mathcal{K} \cup \{c, (a \vee c), (a \vee \neg c)\} \\
\qquad\qquad\qquad\qquad \swarrow \qquad \searrow \\
\qquad\qquad\qquad\qquad \mathcal{K} \cup \{a, c, (a \vee c), (a \vee \neg c)\} \qquad \mathcal{K} \cup \{\neg c, c, (a \vee c), (a \vee \neg c)\}
\end{array}
$$

$-\mathsf{EX}$ See counterexample for SY

$-\mathsf{AI}$ See counter-example for SY

□

The following definition will be useful for proving the following results regarding non-redundant formulae.

Definition A.2. *The sub-structural formulae of a given formula ϕ are defined inductively as*

- $\mathsf{subs}(\varphi) = \{\varphi\}$*, if φ is a literal;*
- $\mathsf{subs}(\varphi \,\Box\, \psi) = \{\varphi \wedge \psi\} \cup \mathsf{subs}(\varphi) \cup \mathsf{subs}(\psi)$*, for $\Box \in \{\wedge, \vee\}$;*
- $\mathsf{subs}(\neg(\varphi \,\Box\, \psi)) = \{\neg(\varphi \,\Box\, \psi)\} \cup \mathsf{subs}(\neg\varphi) \cup \mathsf{subs}(\neg\psi)$*, for $\Box \in \{\wedge, \vee\}$.*

Definition A.3. *Let $\pi = (N, E, \lambda)$ be a tableau for a knowledge base $\mathcal{K}$, we define $\pi[\alpha] = (N, E, \lambda')$ such that $\lambda'(n) = \lambda(n) \cup \{\alpha\}$.*

The tableau $\pi[\alpha]$ stands for a tableau that augments each node of π with the formula φ.

Proposition A.4. *For every knowledge base $\mathcal{K}$, if α is not partially-redundant in $\mathcal{K}$ and π is a tableau for $\mathcal{K}$ then $\pi[\alpha]$ is a tableau for $\mathcal{K} \cup \{\alpha\}$.*

Proof. Let $\pi = (N, E, \lambda)$ be a tableau for $\mathcal{K}$, and α a not partially-redundant formula in $\mathcal{K}$. We will show that $\pi[\alpha] = (N, E, \lambda')$ satisfies all conditions of a tableau:

- $\lambda'(r) = \mathcal{K} \cup \{\alpha\}$, where r is the root of $\pi[\alpha]$. By definition, $\lambda'(r) = \lambda(r) \cup \{\mathbb{K}\}$ and $\lambda(r) = \mathcal{K}$. Thus, $\lambda'(r) = \mathcal{K} \cup \{\alpha\}$.
- Let $n \in N$:

1. we will show $\lambda'(n) \neq \lambda'(n')$, for all $n' \in \mathsf{children}(n)$. Let $n' \in \mathsf{children}(n)$. As π is a tableau, $\lambda(n) \subset \lambda(n)$. By hypothesis, α is not partially-redundant in $\mathcal{K}$ which means that $\varphi \notin \lambda(w)$, for all $w \in N$. Therefore, $\lambda(n) \cup \{\alpha\} \subset \lambda(n') \cup \{\alpha\}$. By definition, $\lambda'(n) = \lambda(n) \cup \{\alpha\}$ and $\lambda'(n') = \lambda(n') \cup \{\alpha\}$. Therefore, $\lambda'(n) \subset \lambda'(n')$ which means that $\lambda'(n) \neq \lambda'(n')$
2. assume $\mathsf{children}(n) = \{n_1\}$. We will show that $\lambda'(n_1) \in \sigma(\varepsilon, \lambda'(n))$, for some $\varepsilon \in \mathcal{R}_{TB} \setminus \{\vee_e\}$. As π is tableau for $\mathcal{K}$, we have that $\lambda(n_1) \in \sigma(\varepsilon, \lambda(n))$ for some $\varepsilon \in \mathcal{R}_{TB} \setminus \{\vee_e\} = \{\wedge_e, \neg\neg_e, DM_\wedge, DM_\vee\}$:
   - "$\varepsilon = \neg\neg_e$". Thus,

     $$\lambda(n_1) = \lambda(n) \cup \{\varphi\}, \text{ for some } \neg\neg\varphi \in \lambda(n).$$

     By definition, $\lambda'(n_1) = \lambda(n_1) \cup \{\alpha\}$, and $\lambda'(n) = \lambda(n) \cup \{\alpha\}$ which implies that $\neg\neg\varphi \in \lambda'(n)$ and

     $$\begin{aligned} \lambda'(n_1) &= \lambda(n) \cup \{\varphi\} \cup \{\alpha\} \\ &= \lambda'(n) \cup \{\varphi\} \end{aligned}$$

     By definition, $\sigma(\neg\neg_e, \lambda'(n)) = \{\lambda'(n) \cup \{\psi\} \mid \neg\neg\psi \in \lambda'(n)\}$. Thus, as $\neg\neg\varphi \in \lambda'(n)$, we get $\lambda'(n) \cup \{\varphi\} \in \sigma(\neg\neg_e, \lambda'(n))$ which means $\lambda'(n_1) \in \sigma(\neg\neg_e, \lambda'(n))$.
   - "$\varepsilon = \wedge_e$". Thus,

     $$\lambda(n_1) = \lambda(n) \cup \{\varphi, \psi\}, \text{ for some } \varphi \wedge \psi \in \lambda(n).$$

     By definition, $\lambda'(n_1) = \lambda(n_1) \cup \{\alpha\}$, and $\lambda'(n) = \lambda(n) \cup \{\alpha\}$ which implies that $\varphi \wedge \psi \in \lambda'(n)$ and

     $$\begin{aligned} \lambda'(n_1) &= \lambda(n) \cup \{\varphi, \psi\} \cup \{\alpha\} \\ &= \lambda'(n) \cup \{\varphi, \psi\} \end{aligned}$$

     By definition, $\sigma(\wedge_e, \lambda'(n)) = \{\lambda'(n) \cup \{\varphi', \psi'\} \mid \varphi' \wedge \psi' \in \lambda'(n)\}$. Thus, as $\varphi \wedge \psi \in \lambda'(n)$, we get $\lambda'(n) \cup \{\varphi, \psi\} \in \sigma(\neg\neg_e, \lambda'(n))$ which means $\lambda'(n_1) \in \sigma(\wedge_e, \lambda'(n))$.
   - "$\varepsilon = DM_\wedge$". Thus,

     $$\lambda(n_1) = \lambda(n) \cup \{\neg\varphi \vee \neg\psi\}, \text{ for some } \neg(\varphi \wedge \psi) \in \lambda(n).$$

     By definition, $\lambda'(n_1) = \lambda(n_1) \cup \{\alpha\}$, and $\lambda'(n) = \lambda(n) \cup \{\alpha\}$ which implies that $\neg(\varphi \wedge \psi) \in \lambda'(n)$ and

     $$\lambda'(n_1) = \lambda(n) \cup \{\neg\varphi \vee \neg\psi\} \cup \{\alpha\}$$

Received \jreceived

$$= \lambda'(n) \cup \{\neg\varphi \vee \neg\psi\}$$

By definition, $\sigma(DM_\wedge, \lambda'(n)) = \{\lambda'(n) \cup \{\neg\varphi' \vee \neg\psi'\} \mid \neg(\varphi' \wedge \psi') \in \lambda'(n)\}$. Thus, as $\neg(\varphi \wedge \psi) \in \lambda'(n)$, we get $\lambda'(n) \cup \{\neg\varphi \vee \neg\psi\} \in \sigma(DM_\wedge, \lambda'(n))$ which means $\lambda'(n_1) \in \sigma(DM_\wedge, \lambda'(n))$.

– "$\varepsilon = DM_\vee$". Thus,

$$\lambda(n_1) = \lambda(n) \cup \{\neg\varphi \wedge \neg\psi\}, \text{ for some } \neg(\varphi \vee \psi) \in \lambda(n).$$

By definition, $\lambda'(n_1) = \lambda(n_1) \cup \{\alpha\}$, and $\lambda'(n) = \lambda(n) \cup \{\alpha\}$ which implies that $\neg(\varphi \vee \psi) \in \lambda'(n)$ and

$$\begin{aligned} \lambda'(n_1) &= \lambda(n) \cup \{\neg\varphi \wedge \neg\psi\} \cup \{\alpha\} \\ &= \lambda'(n) \cup \{\neg\varphi \wedge \neg\psi\} \end{aligned}$$

By definition, $\sigma(DM_\vee, \lambda'(n)) = \{\lambda'(n) \cup \{\neg\varphi' \wedge \neg\psi'\} \mid \neg(\varphi' \vee \psi') \in \lambda'(n)\}$. Thus, as $\neg(\varphi \vee \psi) \in \lambda'(n)$, we get $\lambda'(n) \cup \{\neg\varphi \wedge \neg\psi\} \in \sigma(DM_\vee, \lambda'(n))$ which means $\lambda'(n_1) \in \sigma(DM_\vee, \lambda'(n))$.

Thus, we conclude that $\lambda'(n_1) \in \sigma(\varepsilon, \lambda'(n))$, for some $\varepsilon \in \mathcal{R}_{TB} \setminus \{\vee_e\}$.

3. assume $\mathsf{children}(n) = \{n_1, n_2\}$ with $n_1 \neq n_2$. We will show that either $(\lambda'(n_1), \lambda'(n_2)) \in \gamma(\lambda'(n))$ or $(\lambda'(n_2), \lambda'(n_1)) \in \gamma(\lambda'(n))$. Since π is tableau for $\mathcal{K}$, we have that $(\lambda(n_1), \lambda(n_2)) \in \gamma(\lambda(n))$ or $(\lambda(n_2), \lambda(n_1)) \in \gamma(\lambda(n))$. Without loss of generality, let us assume that $(\lambda(n_1), \lambda(n_2)) \in \gamma(\lambda(n))$. Thus, there is some $\varphi \vee \psi \in \lambda(n)$ such that

$$\begin{aligned} &\lambda(n_1) = \lambda(n) \cup \{\varphi\} \text{ and} \\ &\lambda(n_2) = \lambda(n) \cup \{\psi\} \end{aligned}$$

By definition, $\lambda'(n) = \lambda(n) \cup \{\alpha\}$, while $\lambda'(n_1) = \lambda(n_1) \cup \{\alpha\}$ and $\lambda'(n_2) = \lambda(n_2) \cup \{\alpha\}$. Thus, $\varphi \vee \psi \in \lambda'(n)$ and

$$\begin{aligned} \lambda'(n_1) &= \lambda(n) \cup \{\varphi\} \cup \{\alpha\} \\ &= \lambda'(n) \cup \{\varphi\} \end{aligned} \qquad\qquad \begin{aligned} \lambda'(n_2) &= \lambda(n) \cup \{\psi\} \cup \{\alpha\} \\ &= \lambda'(n) \cup \{\psi\} \end{aligned}$$

By definition, $\gamma(\lambda'(n)) = \{(\lambda'(n) \cup \{\varphi'\}, \lambda'(n) \cup \{\psi'\}) \mid \varphi' \vee \psi' \in \lambda'(n)\}$. Thus, as $\varphi \vee \psi \in \lambda'(n)$, we get that

$$(\lambda'(n) \cup \{\varphi\}, \lambda'(n) \cup \{\psi\}) \in \gamma(\lambda'(n)),$$

which implies $(\lambda'(n_1), \lambda'(n_2)) \in \gamma(\lambda'(n))$. Thus, $(\lambda'(n_1), \lambda'(n_2)) \in \gamma(\lambda'(n))$ or $(\lambda'(n_2), \lambda'(n_1)) \in \gamma(\lambda'(n))$.

Received \jreceived

□

Proposition A.5. *Let $\mathcal{K}$ be a knowledge base, π a tableau for $\mathcal{K}$ and π' a tableau for $\mathcal{K} \cup \alpha$. If α is not partially-redundant in $\mathcal{K}$ then*

(a) $\lambda_{\pi[\alpha]}(n) \cap \mathsf{subs}(\alpha) = \{\alpha\}$, and

(b) if for all node n of π', $\lambda'(n) \cap \mathsf{subs}(\alpha) = \{\alpha\}$ then for every formula $\beta \in \lambda'(n) \setminus \{\alpha\}$, $(\mathsf{subs}(\beta) \cap \mathsf{subs}(\alpha) = \emptyset$,

Proof. Let $\mathcal{K}$ be a knowledge base, π a tableau for $\mathcal{K}$ and π' a tableau for $\mathcal{K} \cup \alpha$, and α a formula not partially-redundant in $\mathcal{K}$

(a) $\lambda_{\pi[\alpha]}(n) \cap \mathsf{subs}(\alpha) = \{\alpha\}$. As α is not partially-redundant in $\mathcal{K}$, we get that $\lambda(n) \cap \mathsf{subs}(\alpha) = \emptyset$. Thus,

$$\begin{aligned}\lambda_{\pi[\alpha]}(n) \cap \mathsf{subs}(\alpha) &= (\lambda(n) \cup \{\alpha\}) \cap \mathsf{subs}(\alpha)\\ &= (\lambda(n) \cap \mathsf{subs}(\alpha)) \cup (\{\alpha\} \cap \mathsf{subs}(\alpha))\\ &= \emptyset \cup \{\alpha\} = \{\alpha\}.\end{aligned}$$

1. (b) for every formula $\beta \in \lambda'(n) \setminus \{\alpha\}$, $(\mathsf{subs}(\beta) \cap \mathsf{subs}(\alpha) = \emptyset$. The proof is by induction on the level of n

Base: level of n is 0, that is, n is the root node. Thus $\lambda'(n) \setminus \{\alpha\} = \mathcal{K}$. By hypothesis, α is not redudant in $\mathcal{K}$ which means that $\mathsf{subs}(\alpha) \cap \mathsf{subs}(\beta) = \emptyset$, for all $\beta \in \mathcal{K}$.

Induction Hypothesis: for all node n' such that $level(n') < level(n)$, $\mathsf{subs}(\alpha) \cap \mathsf{subs}(\beta) = \emptyset$, for all $\beta \in \lambda'(n') \setminus \{\alpha\}$

Induction Step: $level(n) > 0$. Thus, n has a parent node n', and either (i) $\mathsf{children}(n') = \{n\}$ or (ii) $\mathsf{children}(n') = \{n, n_2\}$

(i) $\mathsf{children}(n') = \{n\}$. By the definition of Tableau, $\lambda'(n) \in \sigma(\varepsilon, \lambda'(n'))$, for some $\varepsilon \in \mathcal{R}_{TB} \setminus \{\vee_e\} = \{\wedge_e, \neg\neg_e, DM_\wedge, DM_\vee\}$:

- "$\varepsilon = \neg\neg_e$". Thus,

$$\lambda'(n) = \lambda'(n') \cup \{\varphi\}, \text{ for some } \neg\neg\varphi \in \lambda'(n')$$

which implies

$$\lambda'(n) = (\lambda'(n') \setminus \{\alpha\}) \cup (\{\varphi\} \setminus \{\alpha\}.$$

As n' is the parent of n, we have that $level(n') < level(n)$. We have two cases: either $\alpha = \varphi$ or $\alpha \neq \varphi$

Received \jreceived

• $\alpha = \varphi$. Thus, $\lambda'(n) = (\lambda'(n') \setminus \{\alpha\})$. Thus, from IH: $\mathsf{subs}(\alpha) \cap \mathsf{subs}(\beta) = \emptyset$, for all $\beta \in \lambda'(n') \setminus \{\alpha\}$, taht is, $\mathsf{subs}(\alpha) \cap \mathsf{subs}(\beta) = \emptyset$, for all $\beta \in \lambda'(n)$.

• $\alpha \neq \varphi$. Let $\beta \in \lambda'(n) \setminus \{\alpha\}$. Thus, $\beta \in \lambda'(n') \setminus \{\alpha\}$ or $\beta = \varphi$. For the former, it follows from IH tha $\mathsf{subs}(\beta) \cap \mathsf{subs}(\alpha) = \emptyset$. For the latter, recall that $\neg\neg\varphi \in \lambda'(n')$ and that from hypothesis $\lambda'(n') \cap \mathsf{subs}(\alpha) = \{\alpha\}$. Therefore, $\alpha \neq \neg\neg\varphi$ as $\alpha \neq \varphi$ and $\varphi \in \mathsf{subs}(\neg\neg\varphi)$. Thus $\neg\neg\varphi \in \lambda'(n') \setminus \{\alpha\}$, which implies from IH that $\mathsf{subs}(\neg\neg\varphi) \cap \{\alpha\} = \emptyset$. Thus, as $\varphi \in \mathsf{subs}(\neg\neg\varphi)$, we get that $\mathsf{subs}(\varphi) \cap \mathsf{subs}(\alpha) = \emptyset$. Thus, $\mathsf{subs}(\beta) \cap \mathsf{subs}(\alpha) = \emptyset$, as $\beta = \varphi$.

– the other cases are analagous.

(ii) $\mathsf{children}(n') = \{n, n_2\}$. Analogous to the $\wedge_e$ case.

□

Proposition A.6. *Let $\mathcal{K}$ be a knowledge base and α a formula which is not partially-redundant in $\mathcal{K}$. If π is a tableau for $\mathcal{K} \cup \{\alpha\}$, and for all $n \in \pi, \mathsf{subs}(\alpha) \cap \lambda(n) = \{\alpha\}$ then there is some tableau π' of $\mathcal{K}$ such that $\pi = \pi'[\alpha]$.*

Proof. Let $\mathcal{K}$ be a knowledge base, α be a formula that is not partially-redundant in $\mathcal{K}$, and $\pi = (N, E, \lambda)$ be a tableau for $\mathcal{K} \cup \{\alpha\}$ such that $\mathsf{subs}(\alpha) \cap \lambda(n) = \{\alpha\}$, for all $n \in \pi$. Let $\pi' = (N, E, \lambda')$ such that $\lambda'(n) = \lambda(n) \setminus \{\alpha\}$. We will show that (a) π' is a tableau for $\mathcal{K}$ and (b) $\pi'[\alpha] = \pi$.

(a) We will show that π' satisfy all the conditions of a tableau. Let r be the root of π', and therefore also the root of π.

- $\lambda'(r) = \mathcal{K}$. By definition, $\lambda(\pi) = \mathcal{K} \cup \{\alpha\}$ and $\lambda'(r) = \lambda(r) \setminus \{\alpha\}$. Thus, $\lambda'(r) = \mathcal{K}$.
- let $n \in N$:
  1. let $n' \in \mathsf{children}(n)$. As π is a tableau $\lambda(n) \subset \lambda(n')$. Thus, as α is labelled in both n and n', we have that $\lambda(n) \setminus \{\alpha\} \subset \lambda(n') \setminus \{\alpha\}$ which means $\lambda'(n) \subset \lambda'(n)$. Thus, $\lambda'(n) \neq \lambda'(n')$.
  2. assume $\mathsf{children}(n) = \{n_1\}$. We will show that $\lambda'(n_1) \in \sigma(\varepsilon, \lambda'(n))$, for some $\varepsilon \in \mathcal{R}_{TB} \setminus \{\vee_e\}$. As π is tableau for $\mathcal{K}$, we have that $\lambda(n_1) \in \sigma(\varepsilon, \lambda(n))$ for some $\varepsilon \in \mathcal{R}_{TB} \setminus \{\vee_e\} = \{\wedge_e, \neg\neg_e, DM_\wedge, DM_\vee\}$:
     - "$\varepsilon = \neg\neg_e$". Thus,

$$\lambda(n_1) = \lambda(n) \cup \{\varphi\}, \text{ for some } \neg\neg\varphi \in \lambda(n).$$

Received \jreceived

As π is a tableaux, $\lambda(n) \subset \lambda(n_1)$. Thus, as by hypothesis $\alpha \in \lambda(n)$, we get $\varphi \neq \alpha$. Also, observe that $\alpha \neq \neg\neg\varphi$. Otherwise, we would have that $\varphi \in \mathsf{subs}(\alpha)$, and therefore, we would get $\{\alpha, \varphi\} \subseteq \lambda(n_1) \cap \mathsf{subs}(\alpha)$, a contradiction as by hypothesis $\lambda(n_1) \cap \mathsf{subs}(\alpha) = \{\alpha\}$. Thus, we have

$$\alpha \neq \varphi \text{ and } \alpha \neq \neg\neg\varphi.$$

By definition, $\lambda'(n_1) = \lambda(n_1) \setminus \{\alpha\}$ which implies

$$\lambda'(n_1) = (\lambda(n) \cup \{\varphi\}) \setminus \{\alpha\}.$$

Thus, as $\varphi \neq \alpha$, we get

$$\lambda'(n_1) = (\lambda(n) \setminus \{\alpha\}) \cup \{\varphi\}$$

By definition, $\lambda'(n) = \lambda(n) \setminus \{\alpha\}$. Thus,

$$\lambda'(n_1) = \lambda'(n) \cup \{\varphi\}.$$

Moreover, as $\neg\neg\varphi \in \lambda(n)$ and $\alpha \neq \neg\neg\varphi$, we get that $\neg\neg\varphi \in \lambda'(n)$. By definition,

$$\sigma(\neg\neg_e, \lambda'(n)) = \{\lambda'(n) \cup \{\psi\} \mid \neg\neg\psi \in \lambda'(n)\}.$$

Thus, as $\neg\neg\varphi \in \lambda'(n)$, we get that $\lambda'(n) \cup \{\varphi\} \in \sigma(\neg\neg_e, \lambda'(n))$, which means $\lambda'(n_1) \in \sigma(\neg\neg_e, \lambda'(n))$.

- "$\varepsilon = \wedge_e''$. Thus,

$$\lambda(n_1) = \lambda(n) \cup \{\varphi, \psi\}, \text{ for some } \varphi \wedge \psi \in \lambda(n).$$

Before we proceed, we need first to show that $\alpha \neq \varphi, \alpha \neq \psi$ and $\alpha \neq \varphi \wedge \psi$. If $\alpha = \varphi \wedge \psi$ then we would have that $\varphi, \psi \in \mathsf{subs}(\alpha)$, and therefore, we would get $\{\alpha, \varphi, \psi\} \subseteq \lambda(n) \cap \mathsf{subs}(\alpha)$, a contradiction as by hypothesis $\lambda(n) \cap \mathsf{subs}(\alpha) = \{\alpha\}$. It it was that case that $\alpha = \varphi$ then we would have that $\varphi \wedge \psi, \varphi \in \lambda(n)$. This implies that $\varphi \wedge \psi \in \lambda(n) \setminus \{\alpha\}$. Note that $\mathsf{subs}(\varphi \wedge \psi) \cap \mathsf{subs}(\alpha) \neq \emptyset$. However, from Proposition A.5, we have that $\mathsf{subs}(\varphi \wedge \psi) \cap \mathsf{subs}(\alpha) = \emptyset$ a contradiction. Analogously, we get at the same contraction for $\alpha = \psi$. Therefore,

$$\alpha \neq \varphi, \alpha \neq \psi \text{ and } \alpha \neq \varphi \wedge \psi.$$

By definition, $\lambda'(n_1) = \lambda(n_1) \setminus \{\alpha\}$ which implies

$$\lambda'(n_1) = (\lambda(n) \cup \{\varphi, \psi\}) \setminus \{\alpha\}.$$

Received \jreceived

Thus, as $\alpha \neq \varphi$ and $\alpha \neq \psi$, we get

$$\lambda'(n_1) = (\lambda(n) \setminus \{\alpha\}) \cup \{\varphi, \psi\}$$

By definition, $\lambda'(n) = \lambda(n) \setminus \{\alpha\}$. Thus,

$$\lambda'(n_1) = \lambda'(n) \cup \{\varphi, \psi\}.$$

Moreover, as $\varphi \wedge \psi \in \lambda(n)$ and $\alpha \neq \varphi \wedge \psi$, we get that $\varphi \wedge \psi \in \lambda'(n)$. By definition,

$$\sigma(\wedge_e, \lambda'(n)) = \{\lambda'(n) \cup \{\varphi', \psi'\} \mid \varphi' \wedge \psi' \in \lambda'(n)\}.$$

Thus, as $\varphi \wedge \psi \in \lambda'(n)$, we get that $\lambda'(n) \cup \{\varphi, \psi\} \in \sigma(\wedge_e, \lambda'(n))$, which means $\lambda'(n_1) \in \sigma(\wedge_e, \lambda'(n))$.

– "$\varepsilon = DM_\wedge''$. Thus,

$$\lambda(n_1) = \lambda(n) \cup \{\neg\varphi \vee \neg\psi\}, \text{ for some } \neg(\varphi \wedge \psi) \in \lambda(n).$$

As π is a tableaux, $\lambda(n) \subset \lambda(n_1)$. Thus, as by hypothesis $\alpha \in \lambda(n)$, we get $\alpha \neq \neg\varphi \vee \neg\psi$. Also observe that $\alpha \neq \neg(\varphi \wedge \psi)$. Otherwise, we would have that $\neg\varphi \vee \neg\psi \in \mathsf{subs}(\alpha)$, and therefore, we would get $\{\alpha, \neg\varphi \vee \neg\psi\} \subseteq \lambda(n_1) \cap \mathsf{subs}(\alpha)$, a contradiction as by hypothesis $\lambda(n_1) \cap \mathsf{subs}(\alpha) = \{\alpha\}$. Thus, we have

$$\alpha \neq \neg(\varphi \wedge \psi) \text{ and } \alpha \neq \neg\varphi \vee \neg\psi.$$

By definition, $\lambda'(n_1) = \lambda(n_1) \setminus \{\alpha\}$ which implies

$$\lambda'(n_1) = (\lambda(n) \cup \{\neg\varphi \vee \psi\}) \setminus \{\alpha\}.$$

Thus, as $\alpha \neq \neg\varphi \vee \psi$,

$$\lambda'(n_1) = (\lambda(n) \setminus \{\alpha\}) \cup \{\neg\varphi \vee \psi\}$$

By definition, $\lambda'(n) = \lambda(n) \setminus \{\alpha\}$. Thus,

$$\lambda'(n_1) = \lambda'(n) \cup \{\neg\varphi \vee \psi\}.$$

Moreover, as $\neg(\varphi \wedge \psi) \in \lambda(n)$ and $\alpha \neq \neg(\varphi \wedge \psi)$, we get that $\neg(\varphi \wedge \psi) \in \lambda'(n)$. By definition,

$$\sigma(DM_\wedge, \lambda'(n)) = \{\lambda'(n) \cup \{\neg\varphi' \vee \neg\psi'\} \mid \neg(\varphi' \wedge \psi') \in \lambda'(n)\}.$$

Thus, as $\neg(\varphi \wedge \psi) \in \lambda'(n)$, we get $\lambda'(n) \cup \{\neg\varphi \vee \psi\} \in \sigma(DM_\wedge, \lambda'(n))$, which means $\lambda'(n_1) \in \sigma(DM_\wedge, \lambda'(n))$.

Received \jreceived

  – “$\varepsilon = DM''_\vee$. Analogous to case $\varepsilon = DM_\wedge$.

3. let $\mathsf{children}(n) = \{n_1, n_2\}$ with $n_1 \neq n_2$. We will show $(\lambda'(n_1), \lambda'(n_2)) \in \gamma(\lambda'(n))$ or $(\lambda'(n_2), \lambda'(n_1)) \in \gamma(\lambda'(n))$. Since π is tableau for $\mathcal{K}$, we have that $(\lambda(n_1), \lambda(n_2)) \in \gamma(\lambda(n))$ or $(\lambda(n_2), \lambda(n_1)) \in \gamma(\lambda(n))$. Without loss of generality, let us assume that $(\lambda(n_1), \lambda(n_2)) \in \gamma(\lambda(n))$. Thus, there is some $\varphi \vee \psi \in \lambda(n)$ such that

$$\begin{aligned} \lambda(n_1) &= \lambda(n) \cup \{\varphi\} \text{ and} \\ \lambda(n_2) &= \lambda(n) \cup \{\psi\} \end{aligned}$$

   Before we proceed, we need to show that $\alpha \neq \varphi, \alpha \neq \psi$ and $\alpha \neq \varphi \vee \psi$. If $\alpha = \varphi \vee \psi$ then we would have that $\varphi, \psi \in \mathsf{subs}(\alpha)$, and therefore, we would get $\{\alpha, \varphi, \psi\} \subseteq \lambda(n) \cap \mathsf{subs}(\alpha)$, a contradiction as by hypothesis $\lambda(n) \cap \mathsf{subs}(\alpha) = \{\alpha\}$. It it was that case that $\alpha = \varphi$ then we would have that $\varphi \vee \psi, \varphi \in \lambda(n)$. This implies that $\varphi \vee \psi \in \lambda(n) \setminus \{\alpha\}$. Note that $\mathsf{subs}(\varphi \vee \psi) \cap \mathsf{subs}(\alpha) \neq \emptyset$. However, from Proposition A.5, we have that $\mathsf{subs}(\varphi \vee \psi) \cap \mathsf{subs}(\alpha) = \emptyset$ a contradiction. Analogously, we get at the same contraction for $\alpha = \psi$. Therefore,

$$\alpha \neq \varphi, \alpha \neq \psi \text{ and } \alpha \neq \varphi \vee \psi$$

   By definition, $\lambda'(n) = \lambda(n) \setminus \{\alpha\}$, while $\lambda'(n_1) = \lambda(n_1) \setminus \{\alpha\}$ and $\lambda'(n_2) = \lambda(n_2) \setminus \{\alpha\}$. Thus, as $\alpha \neq \varphi \vee \psi$ and $\varphi \vee \psi \in \lambda(n)$ we get that $\varphi \vee \psi \in \lambda'(n)$. Moreover, as $\alpha \neq \varphi$ and $\alpha \neq \psi$, we get

$$\begin{aligned} \lambda'(n_1) &= (\lambda(n) \cup \{\varphi\}) \setminus \{\alpha\} & \qquad \lambda'(n_2) &= (\lambda(n) \cup \{\psi\}) \setminus \{\alpha\} \\ &= (\lambda(n) \setminus \{\alpha\}) \cup \{\varphi\} & &= (\lambda(n) \setminus \{\alpha\}) \cup \{\psi\} \\ &= \lambda'(n) \cup \{\varphi\} & &= \lambda'(n) \cup \{\psi\} \end{aligned}$$

   By definition, $\gamma(\lambda'(n)) = \{(\lambda'(n) \cup \{\varphi'\}, \lambda'(n) \cup \{\psi'\}) \mid \varphi' \vee \psi' \in \lambda'(n)\}$. Thus, as $\varphi \vee \psi \in \lambda'(n)$, we get that

$$(\lambda'(n) \cup \{\varphi\}, \lambda'(n) \cup \{\psi\}) \in \gamma(\lambda'(n)),$$

   which implies $(\lambda'(n_1), \lambda'(n_2)) \in \gamma(\lambda'(n))$. Thus, $(\lambda'(n_1), \lambda'(n_2)) \in \gamma(\lambda'(n))$ or $(\lambda'(n_2), \lambda'(n_1)) \in \gamma(\lambda'(n))$.

(b) We only have to show that $\lambda_{\pi'[\alpha]}(n) = \lambda(n)$, for all $n \in N$. Let $n \in N$. By definition $\lambda'(n) = \lambda(n) \setminus \{\alpha\}$, and $\lambda_{\pi'[\alpha]}(n) = \lambda'(n) \cup \{\alpha\}$. Thus, $\lambda_{\pi'[\alpha]}(n) =$

Received \jreceived

$(\lambda(n) \setminus \{\alpha\}) \cup \{\alpha\}$. By hypothesis, $\alpha \in \lambda(n)$, as $\mathsf{subs}(\alpha) \cap \lambda(n) = \{\alpha\}$. Therefore, $\lambda_{\pi'[\alpha]}(n) = \lambda(n)$.

□

Proposition A.7. *Let $\mathcal{K}$ be a knowledge base and α a formula which is not partially-redundant in $\mathcal{K}$. If π and π' are tableaux for $\mathcal{K}$ then: $\pi \preceq \pi'$ iff $\pi[\alpha] \preceq \pi'[\alpha]$*

Proof. Let $\mathcal{K}$ be a knowledge base, α be a formula that is not partially-redundant in $\mathcal{K}$, and π and π' be tableaux for $\mathcal{K}$.

"$\Rightarrow$". Let $\pi \preceq \pi'$. Thus there is an injection $\tau : \mathsf{leaf}(\pi) \to \mathsf{leaf}(\pi')$ such that

$$\lambda_\pi(n) \subseteq \lambda_{\pi'}(\tau(n)) \tag{1}$$

Observe, from the definition of $\pi[\alpha]$ and $\pi'[\alpha]$, that $\mathsf{leaf}(\pi) = \mathsf{leaf}(\pi[\alpha])$, $\mathsf{leaf}(\pi') = \mathsf{leaf}(\pi'[\alpha])$. Therefore, τ is also an injection from the leaf nodes of $\pi[\alpha]$ to the leaf nodes of $\pi'[\alpha]$. We only need to show that, $\lambda_{\pi[\alpha]}(n) \subseteq \lambda_{\pi'[\alpha]}(\tau(n))$, for all $n \in \mathsf{leaf}(\pi[\alpha])$. Let $n \in \mathsf{leaf}(\pi[\alpha])$. From Eq. (1), we have that

$$\lambda(n) \cup \{\alpha\} \subseteq \lambda'(\tau(n)) \cup \{\alpha\}.$$

By definition, $\lambda_{\pi[\alpha]}(n) = \lambda_\pi(n) \cup \{\alpha\}$ and $\lambda_{\pi'[\alpha]}(\tau(n)) = \lambda_{\pi'[\alpha]}(\tau(n)) \cup \{\alpha\}$. Therefore, $\lambda_{\pi[\alpha]}(n) \subseteq \lambda_{\pi'[\alpha]}(\tau(n))$.

- "$\Leftarrow$". Let $\pi[\alpha] \preceq \pi'[\alpha]$. Thus there is an injection $\tau : \mathsf{leaf}(\pi[\alpha]) \to \mathsf{leaf}(\pi'[\alpha])$ such that

$$\lambda_{\pi[\alpha]}(n) \subseteq \lambda_{\pi'[\alpha]}(\tau(n)) \tag{2}$$

Observe, from the definition of $\pi[\alpha]$ and $\pi'[\alpha]$, that $\mathsf{leaf}(\pi) = \mathsf{leaf}(\pi[\alpha])$, $\mathsf{leaf}(\pi') = \mathsf{leaf}(\pi'[\alpha])$. Therefore, τ is also an injection from the leaf nodes of π to the leaf nodes of π'. We only need to show that $\lambda_\pi(n) \subseteq \lambda_{\pi'}(\tau(n))$, for all $n \in \mathsf{leaf}(\pi)$. Let $n \in \mathsf{leaf}(\pi)$.

By definition, $\lambda_{\pi[\alpha]}(n) = \lambda_\pi(n) \cup \{\alpha\}$ and $\lambda_{\pi'[\alpha]}(\tau(n)) = \lambda_{\pi'}(\tau(n)) \cup \{\alpha\}$. Thus, from Eq. (2) above we get

$$\lambda_\pi(n) \cup \{\alpha\} \subseteq \lambda_{\pi'}(\tau(n)) \cup \{\alpha\} \tag{3}$$

As α is not partially-redundant in $\mathcal{K}$, we have that α is not labelled in any of the tableaux of $\mathcal{K}$. This means that $\alpha \notin \lambda_\pi(n)$ and $\alpha \notin \lambda_{\pi'}(\tau(n))$. This jointly with Eq. (3) implies

$$\lambda_\pi(n) \subseteq \lambda_{\pi'}(\tau(n)).$$

Received \jreceived

□

Proposition A.8. *If α is not partially-redundant in $\mathcal{K}$ then*

$$\left(\bigcup_{\pi \in \mathcal{T}_{\perp}^{min}(\mathcal{K})} \pi[\alpha]\right) \subseteq \mathcal{T}_{\perp}^{min}(\mathcal{K} \cup \{\alpha\}).$$

Proof. Let us suppose for contradiction that there is a $\pi \in \mathcal{T}_{\perp}^{min}(\mathcal{K})$ such that $\pi[\alpha] \notin \mathcal{T}_{\perp}^{min}(\mathcal{K} \cup \{\alpha\})$. Thus, there is a $\pi' \in \mathcal{T}_{\perp}^{min}(\mathcal{K} \cup \{\alpha\})$ such that $\pi' \prec \pi[\alpha]$. This means that there is some leaf nodes $n' \in \pi'$ and $n \in \pi[\alpha]$ such that

$$\lambda_{\pi'}(n') \subset \lambda_{\pi[\alpha]}(n). \tag{4}$$

Observe that $\alpha \in \lambda(m)$, for all node m of every tableau of $\mathcal{K} \cup \{\alpha\}$. This means that,

$$\{\alpha\} \subseteq \lambda_{\pi'}(n') \cap \mathsf{subs}(\alpha) \text{ and } \{\alpha\} \subseteq \lambda_{\pi[\alpha]}(n) \cap \mathsf{subs}(\alpha)$$

Thus, we have two cases: either (i) $\{\alpha\} = \lambda_{\pi'}(n') \cap \mathsf{subs}(\alpha)$ or (ii) $\{\alpha\} \subset \lambda_{\pi'}(n') \cap \mathsf{subs}(\alpha)$. We get a contradiction in either case:

- (i) $\{\alpha\} = \lambda_{\pi'}(n') \cap \mathsf{subs}(\alpha)$. Thus from Proposition A.6, there is a a tableau π_y for $\mathcal{K}$ such that $\pi' = \pi_y[\alpha]$. From Proposition A.7 we get that $\pi_y \prec \pi$ iff $\pi_y[\alpha] \prec \pi[\alpha]$. Thus, as $\pi' = \pi_y[\alpha]$, we get

$$\pi_y \prec \pi \text{ iff } \pi' \prec \pi[\alpha].$$

  By hypothesis, $\pi' \prec \pi[\alpha]$ which implies that $\pi_y \prec \pi$. Therefore, $\pi \notin \mathcal{T}_{\perp}^{min}(\mathcal{K})$ which is a contradiction.

- (ii) $\{\alpha\} \subset \lambda_{\pi'}(n') \cap \mathsf{subs}(\alpha)$. It follows from Eq. (4) above that $\lambda_{\pi'}(n') \cap \mathsf{subs}(\alpha) \subseteq \lambda_{\pi[\alpha]}(n) \cap \mathsf{subs}(\alpha)$. Therefore,

$$\{\alpha\} \subset \lambda_{\pi'}(n') \cap \mathsf{subs}(\alpha) \subseteq \lambda_{\pi[\alpha]}(n) \cap \mathsf{subs}(\alpha)$$

  which implies that $\{\alpha\} \subset \lambda_{\pi[\alpha]}(n) \cap \mathsf{subs}(\alpha)$. This means that $\{\alpha\} \neq \lambda_{\pi[\alpha]}(n) \cap \mathsf{subs}(\alpha)$. However, from Proposition A.5, we have that $\{\alpha\} = \lambda_{\pi[\alpha]}(n) \cap \mathsf{subs}(\alpha)$ which is a contradiction.

□

Theorem 16. *The inconsistency measures $\mathcal{I}^{\min}$, $\mathcal{I}^{\#}$ and $\mathcal{I}^{\sum}$ satisfy* **NM**.

Received \jreceived

Proof. It follows directly from MO that $\mathcal{I}^{\min}$satisfies NM. For the other two measures, we prove compliance with NM separately:

- $\mathcal{I}^{\#}$: Let $\mathcal{K}$ be a knowledge base and α a non-redundant formula with $\mathcal{K}$. Therefore, from Proposition A.8, we get that $|\mathcal{T}_{\perp}^{min}(\mathcal{K})| \leq |\mathcal{T}_{\perp}^{min}(\mathcal{K} \cup \{\alpha\})|$. Thus, $\mathcal{I}^{\#}(\mathcal{K}) \leq \mathcal{I}^{\#}(\mathcal{K} \cup \{\alpha\})$.

- $\mathcal{I}^{\sum}$: Let $\mathcal{K}$ be a knowledge base and α a non-partially-redundant formula with $\mathcal{K}$. Observe that $|\pi| = |\pi[\alpha]|$. Therefore,

$$\sum_{\pi \in \mathcal{T}_{\perp}^{min}(\mathcal{K})} \frac{1}{|\pi|} = \sum_{\pi \in \mathcal{T}_{\perp}^{min}(\mathcal{K})} \frac{1}{|\pi[\alpha]|}$$

which implies

$$\mathcal{I}^{\sum}(\mathcal{K}) = \sum_{\pi \in \mathcal{T}_{\perp}^{min}(\mathcal{K})} \frac{1}{|\pi[\alpha]|} \tag{5}$$

Let $X = \bigcup_{\pi \in \mathcal{T}_{\perp}^{min}(\mathcal{K})} \pi[\alpha]$. From Proposition A.8, we get $X \subseteq \mathcal{T}_{\perp}^{min}(\mathcal{K} \cup \{\alpha\})$. Thus, $\mathcal{T}_{\perp}^{min}(\mathcal{K} \cup \{\alpha\}) = X \cup (\mathcal{K} \cup \{\alpha\} \setminus X)$. Therefore,

$$\begin{aligned}
\mathcal{I}^{\sum}(\mathcal{K} \cup \{\alpha\}) &= \left(\sum_{\pi \in X} \frac{1}{|\pi|}\right) + \left(\sum_{\pi \in X \setminus \mathcal{T}_{\perp}^{min}(\mathcal{K} \cup \{\alpha\})} \frac{1}{|\pi|}\right) \\
&= \left(\sum_{\pi \in \mathcal{T}_{\perp}^{min}(\mathcal{K})} \frac{1}{|\pi[\alpha]|}\right) + \left(\sum_{\pi \in X \setminus \mathcal{T}_{\perp}^{min}(\mathcal{K} \cup \{\alpha\})} \frac{1}{|\pi|}\right).
\end{aligned}$$

Thus, from Eq. (5), we get

$$\mathcal{I}^{\sum}(\mathcal{K} \cup \{\alpha\}) = \mathcal{I}^{\sum}(\mathcal{K}) + \left(\sum_{\pi \in X \setminus \mathcal{T}_{\perp}^{min}(\mathcal{K} \cup \{\alpha\})} \frac{1}{|\pi|}\right).$$

Thus, $\mathcal{I}^{\sum}(\mathcal{K} \cup \{\alpha\}) \geq \mathcal{I}^{\sum}(\mathcal{K})$.

□

Proposition A.9. *If a formula α is safe within $\mathcal{K}$ then α is not partially-redundant with $\mathcal{K} \setminus \{\alpha\}$.*

Received \jreceived

Proof. Let π be a tableau for $\mathcal{K} \setminus \{\alpha\}$, and π' a tableau for $\{\alpha\}$, we will show that there is no formula φ that is labelled in both π and π'. From Lemma A.1, we have that $\mathsf{At}(\varphi) \subseteq \mathsf{At}(\mathcal{K}\setminus\{\alpha\})$ and $\mathsf{At}(\psi) \subseteq \mathsf{At}(\alpha)$, for all φ that appears in π and all $\psi \in \pi'$. Thus, as α is safe with $\mathcal{K}$, we have that $\mathsf{At}(\mathcal{K}\setminus\{\alpha\})\cap\mathsf{At}(\alpha) = \emptyset$, which means $\mathsf{At}(\varphi)\cap\mathsf{At}(\psi) = \emptyset$. Therefore, there is no common formula between π and π', that is, α is not-partially-redundant. $\square$

Proposition A.10. *If a formula α is not partially-redundant with a knowledge base $\mathcal{K}$ and α is consistent then α is safe in $\mathcal{K} \cup \{\alpha\}$.*

Proof. Let us suppose for contradiction that for some knowledge base $\mathcal{K}$ there is a consistent formula α that is not partially-redundant with $\mathcal{K}$, but it is not safe in $\mathcal{K}\cup\{\alpha\}$. First, observe that each propositional atom in α appears in some tableau of α. By hypothesis, α is not safe in $\mathcal{K} \cup \{\alpha\}$, which means there is a formula $\varphi \in \mathcal{K}$ that shares some atomic proposition p with α, that is $p \in \mathsf{At}(\varphi) \cap \mathsf{At}(\alpha)$. But then p appears in some tableau of $\mathcal{K}$ and in some tableau of α which means that α is partially-redundant with $\mathcal{K}$. This contradicts our hypothesis. Therefore, α is safe. $\square$

To prove compliance of our measures with the postulate SI, we will need some extra constructions. First, given a tableau π for a knowledge base $\mathcal{K}$, and a node n of π, we denote by $\mathsf{subT}(n)$ all the nodes of the subtree rooted on n. A node n has two children, say n_1 and n_2, only when such children were obtained by applying the disjunction rule $DM_\vee$, that is, $\lambda(n_1) \setminus \lambda(n_) = \{\varphi\}$, $\lambda(n_2) \setminus \lambda(n) = \{\psi\}$, and either $\varphi \vee \psi \in \lambda(n)$ or $\psi \vee \varphi \in \lambda(n)$. We say that such a node n is a disjunctive node. In addition, if $\mathsf{At}(\varphi \vee \psi) \cap \mathsf{At}(\alpha) \neq \emptyset$ then we say that such a disjunctive node n is α-connected. Given a tableau π for $\mathcal{K}$, let $\pi[\backslash\alpha] = (N, E, \lambda)$ be a sub-labelled tree of π, such that for each α-connected disjunctive node n of π, we remove exactly one of the sub-trees rooted on one of the two children of n. Given a $\pi[\backslash\alpha]$, we define the function $f_{\pi[\alpha]} : N \to 2^N$ where $f(n) = \{n' \in N \mid (\lambda(n) \setminus \mathsf{forms}(\alpha)) \cup \alpha\}$. Imagine that we re-label each node of the tableau by removing any formula that shares some atomic proposition with α, except α itself. By doing so, some nodes might present the same new label. The function $f_{\pi[\alpha]}$ identifies such nodes whose new labels collapse. The image of $f_{\pi[\alpha]}$ is denoted by $\mathsf{Img}(f_{\pi[\alpha]})$.

We define the collapsed sub-labelled tree of $\pi[\backslash\alpha] = (N', E', \lambda')$ as the labelled tree $\tilde{\pi}[\backslash\alpha] = (\tilde{N}, \tilde{E}, \tilde{\lambda})$, where

- $\tilde{N} = \mathsf{Img}(f_{\pi[\alpha]})$;
- $\tilde{E} = \{(A, B) \in \tilde{N} \times \tilde{N} \mid A \neq B, (n, n') \in E',$ for some $n \in A$ and $n' \in B\}$;
- $\tilde{\lambda}(A) = (\lambda(n') \setminus \mathsf{forms}(\alpha)) \cup \{\alpha\}$, for some $n' \in A$

Lemma A.11. *If n is a disjunctive α-connected node, and n' is a child of n then $(\lambda(n') \setminus \mathsf{forms}(\alpha)) \cup \{\alpha\} = (\lambda(n) \setminus \mathsf{forms}(\alpha)) \cup \{\alpha\}$.*

Received \jreceived

Proof. As n is a disjunctive node and n' is a child of it, we get that $\lambda(n') = \lambda(n) \cup \{\varphi\}$. As n is α-connected, we get that $\varphi \in \mathsf{forms}(\alpha)$. Thus,

$$(\lambda(n) \cup \{\varphi\}) \setminus \mathsf{forms}(\alpha) = \lambda(n) \setminus \mathsf{forms}(\alpha)$$
$$\lambda(n') \setminus \mathsf{forms}(\alpha) = \lambda(n) \setminus \mathsf{forms}(\alpha)$$
$$(\lambda(n') \setminus \mathsf{forms}(\alpha)) \cup \{\alpha\} = (\lambda(n) \setminus \mathsf{forms}(\alpha)) \cup \{\alpha\}$$

□

Proposition A.12. *If π is a tableau of a knowledge base $\mathcal{K}$, and $\alpha \in \mathcal{K}$ is safe then the collapsed sub-labelled tree $\tilde{\pi}[\backslash\alpha] = (\tilde{N}, \tilde{E}, \tilde{\lambda})$, is a tableau of $\mathcal{K}$, for every $\pi[\backslash\alpha]$.*

Proof. Let us show that each condition of the tableau is satisfied:

- $\tilde{\lambda}(r) = \mathcal{K}$. By definition, $\tilde{\lambda}(r) = (\lambda(r) \setminus \mathsf{forms}(\alpha)) \cup \alpha$ and $\lambda(r) = \mathcal{K}$. Thus, $\tilde{\lambda}(r) = (\mathcal{K} \setminus \mathsf{forms}(\alpha)) \cup \alpha$. By hypothesis, α is safe in $\mathcal{K}$, therefore, $((\mathcal{K}) \setminus \mathsf{forms}(\alpha)) = \mathcal{K} \setminus \{\alpha\}$. This implies that $\tilde{\lambda}(r) = (\mathcal{K} \setminus \alpha) \cup \alpha = \mathcal{K}$.
- let $A \in \tilde{N}$:
  1. $\tilde{\lambda}(A) \neq \tilde{\lambda}(A')$, for all children A' of A. By definition, $A \neq B$.
  2. if $\mathsf{children}(A) = \{A_1\}$, then $\lambda(A_1) \in \sigma(\varepsilon, \lambda(A))$, for some derivation rule $\epsilon \in \mathcal{R}_{TB}$. As A' is single child of A, we have that there are some $n \in A$ and $n' \in A'$ such that n' is child of n in π. Let us fix such a n and n'. As A_1 is child of A, we have that $A \neq A_1$ which implies that

     $$(\lambda(n) \setminus \mathsf{forms}(\alpha)) \cup \{\alpha\} \neq (\lambda(n') \setminus \mathsf{forms}(\alpha)) \cup \{\alpha\}.$$

     Thus, from the contrapositive of Lemma A.11, we have that n is not a disjunctive α-connected node. By definition, $\tilde{\lambda}(A) = (\lambda(n) \setminus \mathsf{forms}(\alpha)) \cup \{\alpha\}$ and $\tilde{\lambda}(A_1) = (\lambda(n') \setminus \mathsf{forms}(\alpha)) \cup \{\alpha\}$. Therefore,

     $$\tilde{\lambda}(A) \neq \tilde{\lambda}(A_1).$$

     Therefore, n has only one single node which means that $\lambda(n') \in \sigma(\varepsilon, \lambda(n))$, for some derivation rule $\epsilon \in \mathcal{R}_{TB}$:
     - $\lambda(n') = \lambda(n) \cup \{\varphi\}$, with $\neg\neg\varphi \in \lambda(n)$. Observe that if $\varphi \in \mathsf{forms}(\alpha)$, then we would have $(\lambda(n) \setminus \mathsf{forms}(\alpha)) \cup \{\alpha\} = (\lambda(n') \setminus \mathsf{forms}(\alpha)) \cup \{\alpha\}$. But we have from above that this is not the case, therefore $\varphi \notin \mathsf{forms}(\alpha)$, which means $\{\varphi\} \setminus \mathsf{forms}(\alpha) = \{\varphi\}$. This implies that,

       $$\lambda(n') \setminus \mathsf{forms}(\alpha) = (\lambda(n) \cup \{\varphi\}) \setminus \mathsf{forms}(\alpha)$$

Received \jreceived

$$= (\lambda(n) \setminus \mathsf{forms}(\alpha)) \cup \{\varphi\}$$

Thus,

$$\begin{aligned} \lambda(n') \setminus \mathsf{forms}(\alpha) \cup \{\alpha\} &= (\lambda(n) \setminus \mathsf{forms}(\alpha)) \cup \{\alpha\} \cup \{\varphi\} \\ \tilde{\lambda}(A_1) &= \tilde{\lambda}(A_1) \cup \{\varphi\}. \end{aligned}$$

Thus, $\tilde{\lambda}(A_1) \in \sigma(DM_\wedge, \lambda(A))$.

– the other cases are analogous.

3. if $\mathsf{children}(A) = \{A_1, A_2\}$ with $A_1 \neq A_2$ then there are nodes $n \in A$, $n_1 \in A_1$ and $n_2 \in A_2$ such that both n_1 and n_2 are children of n is π. The proof is analogous to item 2 above.

□

Proposition A.13. *Let π be a tableau for a knowledge base $\mathcal{K}$. If α is safe in $\mathcal{K}$ and there is a node n such that $\mathsf{At}(\lambda(n) \setminus \{\alpha\}) \neq \emptyset$ then π is not minimal.*

Proof. The idea is simple, let us take a collapsed tableau $\tilde{\pi}$ of π. As $\mathsf{At}(\lambda(n) \setminus \{\alpha\}) \neq \emptyset$, there is some formula $\beta \in \lambda(n)$ such that $\beta \in \mathsf{forms}(\alpha)$ and $\beta \neq \alpha$. Consider the following injection $g : \mathsf{leaf}(\tilde{\pi}) \to \mathsf{leaf}(\pi)$ with $g(A) = m \in A$ such that m is a leaf. By definition, $\tilde{\lambda}(A) = \lambda(m) \setminus \mathsf{forms}(\alpha) \cup \{\alpha\} \subseteq \lambda(m)$. Therefore, $\tilde{\pi} \preceq \pi$. As both π and $\tilde{\pi}$ are tableaux, we have that all leaf nodes reachable from n in π contains β. Let m be one of such leaf nodes reachable from n. Thus, $\tilde{\lambda}(A) = \lambda(m) \setminus \mathsf{forms}(\alpha) \cup \{\alpha\}$. As $\beta \in \mathsf{forms}(\alpha)$, $\beta \neq \alpha$ and $\beta \in \lambda(m)$, we get that $\lambda(m) \setminus \mathsf{forms}(\alpha) \cup \{\alpha\} \subset \lambda(m)$. This means $\tilde{\lambda}(A) \subset \lambda(m)$. Therefore, $\pi \not\preceq \tilde{\pi}$. Thus, $\tilde{\pi} \prec \pi$ which means π is not minimal. □

Theorem A.14. *If α is safe in $\mathcal{K}$ then*

$$\mathcal{T}_\perp^{min}(K) = \bigcup_{\pi \in \mathcal{T}_\perp^{min}(\mathcal{K} \setminus \alpha)} \pi[\alpha]$$

Proof. As α is safe in $\mathcal{K}$,we have that α is consistent and non-partially-redundant in $\mathcal{K} \setminus \{\alpha\}$ which implies from Proposition A.8 that $\bigcup_{\pi \in \mathcal{T}_\perp^{min}(\mathcal{K}\setminus\alpha)\pi[\alpha]} \subset \mathcal{T}_\perp^{min}(K)$. Let $X = \bigcup_{\pi \in \mathcal{T}_\perp^{min}(\mathcal{K}\setminus\alpha)} \pi[\alpha]$. Thus, $\mathcal{T}_\perp^{min}(\mathcal{K}) = X \cup (\mathcal{T}_\perp^{min}(\mathcal{K}) \setminus X)$. As α is safe in $\mathcal{K}$, it follows from Lemma A.1 that for every tableau $\pi \in X$, and each node n of π: $\mathsf{At}(\lambda(n)) \cap \mathsf{At}(\alpha) = \emptyset$. Therefore, from Proposition A.13, we get that $(\mathcal{T}_\perp^{min}(\mathcal{K}) \setminus X) = \emptyset$. Therefore, $\mathcal{T}_\perp^{min}(\mathcal{K}) = X$. □

Theorem 20. *For all $n > 0$ and $\mathcal{I} \in \{\mathcal{I}^{\min}, \mathcal{I}^{\#}, \mathcal{I}^{\Sigma}\}$, $\mathcal{C}^v(\mathcal{I}, n) = \mathcal{C}^f(\mathcal{I}, n) = \mathcal{C}^p(\mathcal{I}, n) = \infty$.*

Received \jreceived

Proof. We will have to split the proof for $\mathcal{I}^{\#}$ from $\mathcal{I}^{\min}$ and $\mathcal{I}^{\Sigma}$, for each item 1 and 2.

- $\mathcal{I}^{\#}$. Let us consider the following formulae

$$\alpha_i = (\bigwedge_1^i a)$$

And for $i \in \mathbb{N}$, consider the family of knowledge bases $\mathcal{K}_i$ defined via

$$\mathcal{K}_i = \{(\alpha_1 \wedge \neg\alpha_1) \wedge (\alpha_2 \wedge \neg\alpha_2) \wedge \cdots \wedge (\alpha_i \wedge \neg\alpha_i)\}$$

For example,

$$\begin{aligned}
&\mathcal{K}_1 = \{(a \wedge \neg a)\} \\
&\mathcal{K}_2 = \{(a \wedge \neg a) \wedge \big((a \wedge a) \wedge \neg(a \wedge a)\big)\} \\
&\mathcal{K}_3 = \{(a \wedge \neg a) \wedge \big((a \wedge a) \wedge \neg(a \wedge a)\big) \wedge \big((a \wedge a \wedge a) \wedge \neg(a \wedge a \wedge a)\big)\}
\end{aligned}$$

Each $\mathcal{K}_i$ has exactly i minimal closed tableaux. To see this, observe that we can apply rule $\wedge_e$ to obtain one of the conjunctions $\alpha_j \wedge \neg\alpha_j$, for $1 \leqslant j \leqslant i$. Then we can apply rule $\wedge_e$ again to get a clash. This generates a minimal closed tableau. As we have i conjunctions $\alpha_j \wedge \neg\alpha_j$, we obtain i minmal closed tableaux.

Thus, $\mathcal{I}^{\#}(\mathcal{K}_i) = i$, for all $i > 0$. This means that each $\{\mathcal{I}^{\#}(\mathcal{K}_i) \mid i > 0\}$ is an infinite set. Also note that $|\mathcal{K}_i| = 1$, $\mathsf{At}(\mathcal{K}_i) = \{a\}$, and for all $\varphi \in \mathcal{K}_i$, $\mathsf{At}(\varphi) = \{a\}$. Therefore, for $n > 0$, $\mathcal{C}^v(\mathcal{I}^{\#}, n) = \mathcal{C}^f(\mathcal{I}^{\#}, n) = \mathcal{C}^p(\mathcal{I}^{\#}, n) = \infty$.

- $\mathcal{I}^{\min}, \mathcal{I}^{\Sigma}$. Consider the following family of knowledge bases

$$\mathcal{K}_i^+ = \{\alpha_i^+ \wedge \neg a\},$$

where

$$\begin{aligned}
\alpha_1^+ &= a \\
\alpha_{i+1}^+ &= a \vee (\alpha_i^+)
\end{aligned}$$

For example,

$$\mathcal{K}_1 = \{a \wedge \neg a\}$$

Received \jreceived

$$\mathcal{K}_2 = \{(a \vee a) \wedge \neg a\}$$
$$\mathcal{K}_3 = \{(a \vee (a \vee a)) \wedge \neg a\}$$

Each $\mathcal{K}_i$ has only one minimal closed tableau, and its size is $2i$, thus $\mathcal{I}^{\min}(\mathcal{K}_i) = \mathcal{I}^{\sum}(\mathcal{K}_i) = \frac{1}{2i}$. This implies that for all $i, j > 0$, if $i \neq j$ then $\mathcal{I}^{\min}(\mathcal{K}_i) \neq \mathcal{I}^{\min}(\mathcal{K}_j)$ and $\mathcal{I}^{\sum}(\mathcal{K}_i) \neq \mathcal{I}^{\sum}(\mathcal{K}_j)$. Thus, the sets $\{\mathcal{I}^{\min}(\mathcal{K}_i) \mid i > 0\}$ and $\{\mathcal{I}^{\sum}(\mathcal{K}_i) \mid i > 0\}$ are infinite sets. Also note that $|\mathcal{K}_i| = 1$, $\mathsf{At}(\mathcal{K}_i) = \{a\}$, and for all $\varphi \in \mathcal{K}_i$, $\mathsf{At}(\varphi) = \{a\}$. Therefore, for $n > 0$, $\mathcal{C}^v(\mathcal{I}, n) = \mathcal{C}^f(\mathcal{I}, n) = \mathcal{C}^p(\mathcal{I}, n) = \infty$, for $\mathcal{I} \in \{\mathcal{I}^{\min}, \mathcal{I}^{\sum}\}$.

□

Theorem 21.

1. *For all $n > 1$, $\mathcal{C}^l(\mathcal{I}^{\#}, n) = \infty$.*
2. *For all $n > 3$, and $\mathcal{I} \in \{\mathcal{I}^{\min}, \mathcal{I}^{\sum}\}$, $\mathcal{C}^l(\mathcal{I}, n) = \infty$.*

Proof. 1. Consider the following family of knowledge bases

$$\mathcal{K}_1 = \{a_1, \neg a_1\}$$
$$\mathcal{K}_{i+1} = \mathcal{K}_i \cup \mathcal{K}\{a_{i+1}, \neg a_{i+1}\}$$

Each $\mathcal{K}_i$ has exactly i minimal closed tableaux. Thus, $\mathcal{I}^{\#}(\mathcal{K}_i) = i$, for all $i > 0$. Observe that for all $i > 0$, and $\varphi \in \mathcal{K}_i$, $|\varphi| \leq 2$. Thus, the set $\{\mathcal{I}^{\#}(\mathcal{K}_i) \mid i > 0\}$ is infinite which implies that $\mathcal{C}^l(\mathcal{I}^{\#}, n) = \infty$, for all $n > 1$.

2. Consider the following family of knowledge bases

$$\begin{aligned}
\mathcal{K}_1^+ &= \{a_1, \neg a_1\} \\
\mathcal{K}_2^+ &= \{a_1, \neg a_1 \vee a_2, \neg a_2\} \\
\mathcal{K}_3^+ &= \{a_1, \neg a_1 \vee a_2, \neg a_2 \vee a_3, \neg a_3\} \\
&\cdots \\
\mathcal{K}_{i+1}^+ &= \{a_1, \neg a_1 \vee a_2, \neg a_2 \vee a_3, \ldots, \neg a_i \vee a_{i+1}, \neg a_{i+1}\}
\end{aligned}$$

For, $i > 0$, each $\mathcal{K}_i$ has exactly one minimal closed tableau π, and it is size is $|\pi_i| = 2i + 1$. Thus, $\mathcal{I}^{\min}(\mathcal{K}_i) = \mathcal{I}^{\sum}(\mathcal{K}_i) = \frac{1}{2i+1}$. Observe that, if $i \neq j$, then $\mathcal{I}^{\min}(\mathcal{K}_i) \neq \mathcal{I}^{\min}(\mathcal{K}_j)$ and $\mathcal{I}^{\sum}(\mathcal{K}_i) \neq \mathcal{I}^{\sum}(\mathcal{K}_j)$. Therefore, the set $\{\mathcal{I}(\mathcal{K}_i) \mid i > 0\}$ is infinite, for every $\mathcal{I} \in \{\mathcal{I}^{\min}, \mathcal{I}^{\sum}\}$. Also note that for all $i > 0$, and $\varphi \in \mathcal{K}_i$, $|\varphi| \leq 4$. Thus, $\mathcal{C}^l(\mathcal{I}, n) = \infty$, for all $n > 3$, and $\mathcal{I} \in \{\mathcal{I}^{\min}, \mathcal{I}^{\sum}\}$.

□

Received \jreceived

Theorem 22. *For $\mathcal{I} \in \{\mathcal{I}^{\#}, \mathcal{I}^{\min}, \mathcal{I}^{\#}\}$,* EXACT$_{\mathcal{I}}$*,* UPPER$_{\mathcal{I}}$*, and* LOWER$_{\mathcal{I}}$ *are in EXPSPACE, while* VALUE$_{\mathcal{I}}$ *is in FEXPSPACE (the functional variant of EXPSPACE).*

Proof. First, we show that VALUE$_{\mathcal{I}}$ is in FEXPSPACE. From this, we prove that the other problems are in EXPSPACE.

- VALUE$_{\mathcal{I}}$ is in FEXPSPACE, for all $\mathcal{I} \in \{\mathcal{I}^{\#}, \mathcal{I}^{\min}, \mathcal{I}^{\#}\}$.

  Given a knowledge base $\mathcal{K}$, we will show first how one can compute $\mathcal{I}(\mathcal{K})$, for all $\mathcal{I} \in \{\mathcal{I}^{\#}, \mathcal{I}^{\min}, \mathcal{I}^{\#}\}$. The idea is simple, we enumerate all tableaux, and we mark all minimal tableaux, thereafter we count and check the size of each minimal tableaux. First, note that we do not allow two nodes on the same branch of a tableau to have the same label (if the application of a rule repeats some label on the branch, we ignore this application and look for another rule application). As $\mathcal{K}$ is finite and each formula is finite, at each derivation step there is only a finite number of possible derivations and the number of possibilities reduces in the following derivation step. Therefore, the procedure eventually finishes. Each branch has at most linear size on the sum of the sizes of the formulae in $\mathcal{K}$, while a tableau can have exponential size on the sum of the sizes of the formulae in $\mathcal{K}$. And we have an exponential number of tableaux on the size of the sum of the sizes of the formulae in $\mathcal{K}$. To determine $\mathcal{I}(\mathcal{K})$, for any $\mathcal{I} \in \{\mathcal{I}^{\#}, \mathcal{I}^{\min}, \mathcal{I}^{\#}\}$, we (1) enumerate all such tableaux, (2) check which ones of them are minimal, and (3) for $\mathcal{I}^{\#}(\mathcal{K})$, we count the number of such minimal tableaux. For $\mathcal{I}^{\min}$, we visit each minimal tableaux, keeping the size of the minimal tableau visited so far. The value of $\mathcal{I}^{\min}(\mathcal{K})$ corresponds to the value obtained when we finish visiting all minimal tableaux. For $\mathcal{I}^{\sum}$, the process is analogous, we just need to keep a counter that is incremented every time that we find a minimal tableau with the same size as the least tableau so far computed. However, if a smaller tableau is found, then we reset this counter to one. At the end of the procedure, we obtain the correct value of $\mathcal{I}^{\sum}(\mathcal{K})$. This strategy takes an exponential space, since we have an exponential number of tableaux (as explained above), and each of them has at most exponential size.

- The problems LOWER, UPPER and EXACT are easily solved by using the TM that computes VALUE$_{\mathcal{I}}$. In the input $\mathcal{K} \in \mathbb{K}, x \in \mathbb{R}^{\infty}_{\geq 0} \setminus \{0\}$, simulate the TM M that solves VALUE$_{\mathcal{K}}$, that we presented above. To compute LOWER, UPPER and EXACT, we only need to compare x with the value returned by M.

□

Received \jreceived