Available online at www.sciencedirect.com

ScienceDirect

Artificial Intelligence 172 (2008) 1064–1093

Artificial Intelligence

www.elsevier.com/locate/artint

Analysing inconsistent first-order knowledgebases

John Grant [a,b], Anthony Hunter [c,*]

[a] *Department of Mathematics, Towson University, Towson, MD 21252, USA*
[b] *Department of Computer Science, University of Maryland, College Park, MD 20742, USA*
[c] *Department of Computer Science, University College London, Gower Street, London WC1E 6BT, UK*

Received 31 December 2006; received in revised form 14 November 2007; accepted 14 November 2007

Available online 22 November 2007

Abstract

It is well-known that knowledgebases may contain inconsistencies. We provide a framework of measures, based on a first-order four-valued logic, to quantify the inconsistency of a knowledgebase. This allows for the comparison of the inconsistency of diverse knowledgebases that have been represented as sets of first-order logic formulae. We motivate the approach by considering some examples of knowledgebases for representing and reasoning with ontological knowledge and with temporal knowledge. Analysing ontological knowledge (including the statements about which concepts are subconcepts of other concepts, and which concepts are disjoint) can be problematical when there is a lack of knowledge about the instances that may populate the concepts, and analysing temporal knowledge (such as temporal integrity constraints) can be problematical when considering infinite linear time lines isomorphic to the natural numbers or the real numbers or more complex structures such as branching time lines. We address these difficulties by providing algebraic measures of inconsistency in first-order knowledgebases.

Keywords: Measuring inconsistency; Paraconsistent logics; Inconsistency tolerance; Analysing inconsistency; Conflict resolution

1. Introduction

The need for handling inconsistencies in knowledgebases has been well recognised in recent years. Inconsistencies may arise for various reasons such as when information sources are merged or in the presence of integrity constraints. The use of first-order logic becomes problematical because a single (local) inconsistency leads to the (global) inconsistency of the entire knowledgebase. Paraconsistent logics allow for local inconsistency without global inconsistency. Paraconsistent reasoning is important in handling inconsistent information, and there have been a number of proposals for paraconsistent logics, such as Da Costa's C_ω logics [11], developments of C systems [9], Priest's three-valued logic LPm [33], Belnap's four-valued logic [5], and versions of Belnap's four-valued logic restricted to minimal models [1], for reasoning with inconsistent information. Further approaches, such as techniques for analysing and querying inconsistent databases and knowledgebases [2,3,12,31], techniques for merging knowledgebases [4,7,27,28], and analytical techniques for inconsistent software specifications [19], have been proposed (for reviews of some applications see

* Corresponding author.
E-mail addresses: jgrant@towson.edu (J. Grant), a.hunter@cs.ucl.ac.uk (A. Hunter).

0004-3702/$ – see front matter
doi:10.1016/j.artint.2007.11.006

[6,16]). Whilst these methods provide potentially valuable ways of using inconsistent knowledgebases, they do not provide an adequate way of summarising the nature of the inconsistencies.

Our interest in this paper is in providing a measure for the inconsistency of a knowledgebase represented as a set of first-order logic formulae. By providing such a measure we can compare different knowledgebases and evaluate their quality of information. If given the opportunity to choose between different knowledgebases, we may try to choose the one that is least inconsistent.

Four-valued paraconsistent logics have been used as the basis of an approach to measuring inconsistency in knowledgebases [14,20,21]. In this, each inconsistent set of formulae is reflected in the four-valued models for the set, and then the inconsistency is measured in the models. This approach to measuring inconsistency has already been seen as a useful tool in analysing a diverse range of information types including news reports [23], integrity constraints [14], ontologies [32], software specifications [8,30], and ecommerce protocols [10]. However, this approach of measuring inconsistency has been restricted to either a propositional language or a language with predicates but without function symbols.

In this paper, we present a framework for measuring inconsistency for a full first-order language, together with examples in analysing ontological and temporal knowledge. Dealing with a full first-order language is potentially important in diverse applications (such as reasoning about specifications [13]), but it does also raise issues with regard to analysing arbitrarily large, including infinite, domains. To address these issues, our framework provides algebraic measures of inconsistency in first-order knowledgebases.

2. Overview of our approach

In this section, we provide an informal overview of our approach together with some examples to motivate and illustrate our approach. We start by recalling that many diverse applications in computer science require the ability to represent and reason with knowledge in a form that is more expressive than propositional logic. Furthermore, in many applications, there is a need to analyse inconsistency arising in knowledge.

To illustrate the need for systems and/or users to analyse inconsistency, consider diverse applications such as tools for analysing formal software specifications (where parts of the specifications may have come from different sources), systems for disambiguation in natural language processing (where there are conflicting syntactic, semantic, or pragmatic parses of the text/speech being parsed), and tools for developing ontologies based on description logics (where there may be multiple ontologies perhaps from multiple sources that need to be combined by an ontology engineer into a single coherent and consistent ontology). In these examples, and in many other potential applications, there is either the need for an automatic system to analyse the degree of inconsistency arising in the available knowledge, or there is the need for a system to provide a user (such as a software or knowledge engineer) with an assessment of the degree of inconsistency arising in the available knowledge. Once the system/user has access to an assessment of the degree of inconsistency, the system/user can make a more intelligent and better informed decision on the course of action to take on the inconsistency.

In this paper we assume a knowledgebase is a set of formulae of classical first-order logic. We impose no restrictions on this. It can include function symbols, variable symbols, and quantifier symbols. And of course, a knowledgebase can be inconsistent, and indeed, any formula in a knowledgebase may be inconsistent.

Our approach to measuring inconsistency in a knowledgebase is to consider the "four-valued models" of it. Each of these models is based on what we call a bistructure, which essentially is a pair of classical interpretations: One of these interpretations is used for the satisfaction of positive literals (i.e. the atoms), and the other is used for the satisfaction of negative literals. So in a bistructure, both an atom and its negation, or neither, can be satisfied. This gives a four-valued semantics, so that an atom may be regarded as being exactly one of "true" or "false" or "both true and false" or "neither true nor false" in a bistructure. The semantics for more complex formulae is given by a generalisation of Belnap's four valued logic, which is a paraconsistent logic that we call tolerant logic. For our purposes, this semantics is simple and the set of models for any knowledgebase is always nonempty.

Given a bistructure, we apply a simple measure of inconsistency, denoted Inc, that gives the proportion of the tuples in the bistructure that are in conflict. The amount of conflict in a bistructure is the number of tuples that are both true and false. This is normalised by the total number of tuples that are possible in the interpretations (which is a function of the size of the domain), so we get a value in the [0, 1] interval. For example, if we have a bistructure with just one monadic relation R and two domain objects a_1 and a_2, and the first classical interpretation has both $\langle a_1\rangle$ and $\langle a_2\rangle$ (for

R), and the second classical interpretation has $\langle a_1 \rangle$ (for $\neg R$), then there is conflict with respect to the tuple $\langle a_1 \rangle$ and so the proportion of tuples in conflict is $1/2$. Note, this measure is not restricted to Herbrand interpretations.

We then generalise this measure of inconsistency to sets of bistructures. In order to set up our framework, and consider various properties, we deal with sets of bistructures in general. But in practice, if we want to analyse a knowledgebase, we consider the set of models for the knowledgebase.

For a knowledgebase, since our measure of inconsistency of a model is dependent on the domain size, we consider the models for each domain size in turn. For each domain size, we find the minimum degree of inconsistency in a model from the models of this size, using a function denoted MicroInc, and then we summarise this value obtained for each size in the form of a ratio of univariate polynomial functions (i.e. a rational function) where the variable is the cardinality of the domain. The polynomial that is the numerator gives the minimum number of tuples in conflict for the models of domain size n, and the polynomial that is the denominator gives the maximum number of tuples in the models of domain size n. By representing the degree of inconsistency in the form of such a rational function, we have a concise summary of the nature of the inconsistency for any domain size. Furthermore, it provides a direct way of comparing knowledgebases in terms of their respective rational functions.

To illustrate our approach, we now look at examples of knowledgebases to show some of the key aspects of measuring inconsistency. We start with some simple examples based on pairs of formulae taken from the following list of formulae (A1)–(A7). For each pair we consider, for example (A2, A3), imagine there are two agents who have provided the formulae, and these two agents need to jointly provide a formula that they both can agree on. Perhaps the agents need to do this as part of a requirements capture process for some software system for which the agents are stakeholders. So each formula represents the requirements of one of the agents. By measuring the inconsistency of the union of the two formulae, we get a measure of how divergent the two agents are in their positions. Furthermore, if the agents are negotiating, they may withdraw one or both formulae, and replace them with formulae that are less inconsistent. Such a negotiation may be undertaken with the aim of finding a pair of formulae that are consistent together.

(A1) $P(a)$
(A2) $\neg P(a)$
(A3) $\forall x.P(x)$
(A4) $\exists x.\neg P(x)$
(A5) $\neg\exists x.P(x)$
(A6) $\forall x.\neg P(x)$
(A7) $\forall x.\neg P(f(x))$

For the pair (A1, A2), we may describe this as an "atomic conflict" (A1 says one domain object is in the interpretation for P whereas A2 says that one domain object is in the interpretation for the negation of P). For the pairs (A2, A3) and (A3, A4), we have something similar to the case for (A1, A2), in that there is at least one domain object in conflict. So, for each of the pairs (A1, A2), (A2, A3), and (A3, A4), if our knowledgebase contains just the two formulae in the pair, then we will calculate the measure of inconsistency as the rational function $1/n$, and in the limit, as n goes to infinity, the degree of inconsistency is 0.

For the pairs (A3, A5) and (A3, A6), we have more significant inconsistency. In the models, all domain objects are in conflict in each pair. So, for each of (A3, A5) and (A3, A6), if the knowledgebase contains just the two formulae in the pair, then we will calculate the measure of inconsistency as the rational function n/n, and in the limit, the degree of inconsistency is 1.

For the pair (A3, A7), we have something similar to the pair (A3, A6), but here we also need to consider the function symbol f in the right formula. If we consider the models with the fewest conflicts (which we will see are the models we want to base our measures on), then the interpretation of the function symbol should be a constant function, i.e. there is $c \in D$ such that for all $d \in D$, $f(d) = c$. In this case, there is one domain object in conflict, namely the c just mentioned. So, for (A3, A7), if the knowledgebase contains just the two formulae in the pair, then we will calculate the measure of inconsistency as the rational function $1/n$, and in the limit, the degree of inconsistency is 0.

We now consider examples of ontological knowledge. Whilst description logics are now the leading approach to formalising ontological knowledge, the basic description logics are actually subsystems of classical logic; so it is straightforward to present examples of ontological knowledge in the form of classical logic. For this, we adopt the following conventions: (1) A concept P is represented by a monadic predicate $P(x)$; (2) An individual c that is a member of a concept P is represented by a ground predicate $P(c)$; (3) The relationship that the concept Q is a subconcept of the concept P is represented by $\forall x.Q(x) \rightarrow P(x)$; and (4) The relationship that the concept Q is disjoint with the concept P is represented by $\forall x.Q(x) \rightarrow \neg P(x) \wedge \forall x.P(x) \rightarrow \neg Q(x)$.

We see a key advantage of our approach for analysing an ontology (when it is under development) if we consider the need to analyse the structure (i.e. the concepts and their inter-relationships) without knowing about the instances that may populate it. For example, for a medical records ontology, the ontology engineer should have obtained knowledge that for example the concept *heart surgery* is a subclass of the concept *surgery*, and that *male* and *female* are disjoint concepts, but the ontology engineer cannot be expected to have a list of all the patients of the hospital in the future. Hence, when the ontology is being developed, the number of instances that will be in the ontology is unknown. Our approach directly deals with this issue, since we can consider an arbitrarily large number of instances, which we do by considering an arbitrary-sized domain. In other words, our measure of inconsistency, captured by a rational function, is a representation of the inconsistency for each domain size.

We start by considering the formulae (B1)–(B4). These formulae are not inconsistent. They say that S is a subconcept of R, Q is a subconcept of P, Q and S are disjoint, and P and R are disjoint. However, if we also have the assumption (B5) that says that there is an instance in S and Q, then we do have an inconsistency.

(B1) $\forall x.S(x) \rightarrow R(x)$
(B2) $\forall x.(Q(x) \rightarrow \neg S(x) \wedge S(x) \rightarrow \neg Q(x))$
(B3) $\forall x.Q(x) \rightarrow P(x)$
(B4) $\forall x.(P(x) \rightarrow \neg R(x) \wedge R(x) \rightarrow \neg P(x))$
(B5) $\exists x.S(x) \wedge Q(x)$

So without knowing anything about the actual membership of these concepts, we can analyse the inconsistency in this ontological knowledge. For (B1)–(B5), we will see in Example 15, that the rational function is $1/n$. This means that as the size of the domain increases, the inconsistency is diluted, and in the limit, the degree of inconsistency is reduced to zero.

As another example of ontological knowledge, consider (C1)–(C7) which are inconsistent. For this, we will see (in Example 17) that the degree of inconsistency is given by the rational function $1/3$. So as the size of the domain increases, the inconsistency is not diluted, and in the limit, the degree of inconsistency is $1/3$. Furthermore, if we compare (B1)–(B5) and (C1)–(C7), for $n > 3$, the rational function for (C1)–(C7) is always greater than that for (B1)–(B5), and so we can regard (C1)–(C7) as more inconsistent than (B1)–(B5).

(C1) $\forall x.S(x) \rightarrow R(x)$
(C2) $\forall x.Q(x) \rightarrow P(x)$
(C3) $\forall x.(Q(x) \rightarrow \neg S(x) \wedge S(x) \rightarrow \neg Q(x))$
(C4) $\forall x.(P(x) \rightarrow \neg R(x) \wedge R(x) \rightarrow \neg P(x))$
(C5) $\forall x.(T(x) \rightarrow \neg U(x) \wedge U(x) \rightarrow \neg T(x))$
(C6) $\forall x.(T(x) \rightarrow U(x) \wedge U(x) \rightarrow T(x))$
(C7) $\forall x.T(x) \vee U(x)$

We now turn to temporal knowledge. The following set of formulae (D1)–(D3) is consistent in classical logic but the set is only satisfied by an infinite model such as one based on the sequence of the natural numbers. Such a set of formulae may appear as part of a specification for time-stamping locutions in a dialogue protocol between two interacting agents.

(D1) $\forall x, \exists y.R(x, y)$
(D2) $\forall x, y.(R(x, y) \rightarrow \neg R(y, x))$
(D3) $\forall x, y, z.(R(x, y) \wedge R(y, z) \rightarrow R(x, z))$

Now suppose the formula (D4) is added to (D1)–(D3); then the set is inconsistent. However, in a sense, the conflict is extremely small, and so any bistructure for it is “overwhelmingly consistent”. In contrast, if the following formula (D5) is added to (D1)–(D3), then the set is inconsistent, and in a sense, the conflict is extremely large, and so any bistructure for it is “substantially inconsistent”.

(D4) $R(1,1)$
(D5) $\forall x.R(x,x)$

In our framework, we will provide a degree of inconsistency to account for inconsistency in infinite models (allowing us for example to differentiate between the inconsistency in the set $\{$D1, D2, D3, D4$\}$, and the set $\{$D1, D2, D3, D5$\}$), and explore some of the relationships between them, as well as with the degree of inconsistency for finite models. We will also provide a measure of the consistency of an infinite model which provides an alternative dimension for analysing an infinite model.

Since the proposal in this paper is the first proposal for measuring inconsistency in full first-order logic (including infinite models), our approach offers considerable advantages for applications in artificial intelligence and computer science involving first-order knowledge. A number of other proposals have been made for measuring the degree of information in the presence of inconsistency [24,26,29,37], and for measuring the degree of inconsistency in information [14,15,17,18,20–22,24–26,36]. All these proposals are based on propositional logic, apart from [15] and [14], with the former primarily investigating the mathematical structure of various inconsistency measures, while the latter is based on a restricted form of first-order logic. Furthermore, there are six key improvements in this new paper over the [14] paper:

1. In [14] we only considered a restricted first-order language with universal and existential quantification but without function symbols (apart from constant symbols) whereas in this paper we consider full first-order logic, and therefore in this paper we are able to handle a wider range of knowledgebases.
2. In [14] we only considered Herbrand interpretations, whereas in this paper, we consider any first-order interpretation, and therefore we drop some constraints that are inappropriate for some applications.
3. In [14] we only considered finite interpretations, whereas in this paper, we consider both finite and infinite interpretations, and therefore in this paper we are able to handle a wider range of knowledgebases.
4. In [14] we used quasi-classical logic to find the four-valued models of a knowledgebase, whereas in this paper we use a first-order version of Belnap’s logic, and as a result we have a simpler logic for finding the models for a knowledgebase.
5. In [14] the measure of inconsistency for a knowledgebase was summarised by a sequence of numbers $\langle n_1, n_2, n_3, \ldots\rangle$ (where n_1 is the measure for a domain of cardinality 1, n_2 is the measure for a domain of cardinality 2, and so on), whereas in this paper, the measure of inconsistency for a knowledgebase is summarised by a ratio of univariate polynomial functions (i.e. a rational function).
6. In [14] we did not consider limit behaviour of measures, whereas in this paper we provide a characterisation of measures in terms of limit behaviour.

In the following sections, we formalise our approach a follows. In Section 3 we review the basic definitions for the language and semantics of first-order logic that we require. In Section 4 we present a first-order version of the semantics for Belnap’s four-valued logic, called tolerant logic, that we will use to find the models for a knowledgebase. In Section 5 we consider some classes of interpretations that allow us to restrict the models considered for a knowledgebase (e.g. for temporal knowledge, we may wish to restrict consideration to models with elements corresponding to the real numbers). In Section 6 we consider measures for a finite set of finite models (which we call a bounded frame), for a set of finite models that includes a model of every domain size (which we call an unbounded frame), and for a set of models that includes an infinite model (which we call an infinite frame). In Section 7 we consider a framework for measuring consistency which is a counterpart to our framework for measuring inconsistency. Finally, in Section 8 we show that the semantics for tolerant logic subsumes the semantics for Belnap’s logic.

3. Basic definitions for a first-order logic

In this section, we provide some basic definitions and notation that are used for presenting a *first-order logic* (FOL).

The language for FOL contains logical symbols: connectives $\{\neg, \vee, \wedge, \rightarrow\}$, quantifiers $\{\forall, \exists\}$, punctuation symbols (parentheses, comma, and period), and an infinite set of variables. A specific language $\mathcal{L}$ is determined by its predicate, constant, and function symbols; these we consider the nonlogical symbols that must be provided in a language. We assume that the number of predicate and function symbols is finite. We sometimes write $P(n)$ to indicate that P is an n-ary predicate symbol.

We adopt the following conventions for our notation.

- Uppercase letters like P and R for predicate symbols.
- Lowercase letters like f, g, and h, perhaps with subscript, for function symbols.
- Lowercase letters like a, b, c, and d, perhaps with subscript, for constant symbols.
- Lowercase letters like x and y, perhaps with subscript, for variable symbols.
- Lowercase letters like t and s, perhaps with subscript, for terms.

We assume the usual classical definitions for a language including definitions for a free variable, a bound variable, and a ground formula. An atom is of the form $P(t_1, \ldots, t_n)$, where $t_1, \ldots, t_n$ are terms. As usual, a literal is either an atom or the negation of an atom. The set of formulae is defined by the usual inductive definitions for classical logic. We use the Greek letters α, β, γ for literals, ϕ for a clause (a disjunction of literals), ψ for a conjunction of clauses, and θ for any formula. For a language $\mathcal{L}$, the set of formulae that can be formed by the usual inductive definitions is denoted $\mathsf{Formulae}(\mathcal{L})$. We will usually not specify $\mathcal{L}$ and assume that given a knowledgebase Δ, $\mathcal{L}$ is the language that contains exactly the nonlogical (i.e. constant, function, and predicate) symbols that appear in Δ.

We now consider the classical semantics for FOL.

Definition 1. A *classical structure* for the language $\mathcal{L}$ is a pair (D, I), where D is a nonempty set called the *domain* and I is a function called an *interpretation* that makes assignments to the symbols of $\mathcal{L}$ as follows:

1. For every constant symbol c, $I(c) \in D$.
2. For every function symbol f of arity $n > 0$, $I(f): D^n \mapsto D$ is an n-ary function.
3. For every predicate symbol P of arity $n > 0$, $I(P) \subseteq D^n$ is an n-ary relation.

We handle variables in FOL formulae using the standard notion of an assignment.

Definition 2. Let (D, I) be a classical structure, and let V be the set of variable symbols in $\mathcal{L}$. An *assignment* A for (D, I) is a function $A: V \mapsto D$. Given an assignment A, an *x-variant assignment* A' is the same as A except perhaps in the assignment for the variable x.

Whilst the definitions for language and interpretations considered in this section are those of classical logic, we will use them for a paraconsistent logic in the next section.

In order to consider properties of our framework, we also require the classical consequence relation, denoted $\vdash$. We assume $\perp$ is shorthand for any classically inconsistent formula. However, in order to simplify the presentation we assume that $\perp$ is not in the language $\mathcal{L}$. For a knowledgebase $\Delta \subseteq \mathsf{Formulae}(\mathcal{L})$, as a shorthand we write $\Delta \vdash \perp$ to indicate that Δ is inconsistent in classical logic; otherwise we write $\Delta \nvdash \perp$.

4. Tolerant logic

We now present the definitions for *tolerant logic* which is a first-order four-valued logic. The language for tolerant logic is that of FOL. However, the semantics is different; that is why tolerant logic supports paraconsistent reasoning.

The notion of a bistructure in tolerant logic is based on the notion of a classical interpretation. The basic difference is that for tolerant logic we use a pair of classical interpretations to give a tolerant interpretation.

Definition 3. A *bistructure* is a tuple (D, I^+, I^-) where (D, I^+) and (D, I^-) are classical interpretations, and for all constant symbols c, $I^+(c) = I^-(c)$, and for all function symbols f, $I^+(f) = I^-(f)$.

The above definition ensures that in a bistructure (D, I^+, I^-) both the classical interpretations I^+ and I^- use the same domain object for each constant symbol, and the same function in the domain for each function symbol. Therefore the classical interpretations I^+ and I^- in a bistructure can only differ in their assignment to predicate symbols. As a result, we can use I^+ as the interpretation for positive literals and I^- as the interpretation for negative literals. This is formalised in the definition for decoupled satisfaction.

Definition 4. For a bistructure $E = (D, I^+, I^-)$ and an assignment A, we define a satisfiability relation, $\models_d$, called *decoupled satisfaction* for literals in $\mathcal{L}$ as follows:

$$(E, A) \models_d P(t_1, \dots, t_n) \quad \text{iff} \quad \langle I^*(t_1), \dots, I^*(t_n)\rangle \in I^+(P)$$
$$(E, A) \models_d \neg P(t_1, \dots, t_n) \quad \text{iff} \quad \langle I^*(t_1), \dots, I^*(t_n)\rangle \in I^-(P)$$

where for $1 \leqslant i \leqslant n$,

- if t_i is a variable, then $I^*(t_i) = I^+(A(t_i)) = I^-(A(t_i))$
- if t_i is a constant, then $I^*(t_i) = I^+(t_i) = I^-(t_i)$
- if t_i is of the form $f(s_1, \dots, s_m)$, then $I^*(t_i) = I^+(f)(I^*(s_1), \dots, I^*(s_m)) = I^-(f)(I^*(s_1), \dots, I^*(s_m))$

Since we allow both an atom and its complement to be satisfiable, we have decoupled, at the level of the structure, the link between an atom and its complement. In contrast, if a classical structure satisfies a literal, then it is forced to not satisfy the complement of the literal. This decoupling gives the basis for a semantics for paraconsistent reasoning.

In the following definition for satisfiability for arbitrary formulae, we provide a partial coupling (i.e. a coupling that is weaker than in classical logic) for a formula and its complement. In the propositional case, this definition of satisfiability coincides with that of Belnap's four-valued logic [5], which is a propositional logic that has a four-valued lattice-theoretic interpretation of connectives (see Theorem 5).

Definition 5. Let E be a bistructure and let A be an assignment. The *satisfiability relation*, denoted $\models$, is defined by induction on the length of a formula as follows where α is a literal, and θ, θ_1, and θ_2 are arbitrary formulae.

$$(E, A) \models \alpha \quad \text{iff} \quad (E, A) \models_d \alpha$$
$$(E, A) \models \theta_1 \vee \theta_2 \quad \text{iff} \quad (E, A) \models \theta_1 \text{ or } (E, A) \models \theta_2$$
$$(E, A) \models \theta_1 \wedge \theta_2 \quad \text{iff} \quad (E, A) \models \theta_1 \text{ and } (E, A) \models \theta_2$$
$$(E, A) \models \theta_1 \rightarrow \theta_2 \quad \text{iff} \quad (E, A) \models \neg\theta_1 \text{ or } (E, A) \models \theta_2$$
$$(E, A) \models \neg\neg\theta \quad \text{iff} \quad (E, A) \models \theta$$
$$(E, A) \models \neg(\theta_1 \vee \theta_2) \quad \text{iff} \quad (E, A) \models \neg\theta_1 \text{ and } (E, A) \models \neg\theta_2$$
$$(E, A) \models \neg(\theta_1 \wedge \theta_2) \quad \text{iff} \quad (E, A) \models \neg\theta_1 \text{ or } (E, A) \models \neg\theta_2$$
$$(E, A) \models \neg(\theta_1 \rightarrow \theta_2) \quad \text{iff} \quad (E, A) \models \theta_1 \text{ and } (E, A) \models \neg\theta_2$$
$$(E, A) \models \exists x.\theta \quad \text{iff} \quad \text{for some } x\text{-variant assignment } A', (E, A') \models \theta$$
$$(E, A) \models \forall x.\theta \quad \text{iff} \quad \text{for all } x\text{-variant assignments } A', (E, A') \models \theta$$
$$(E, A) \models \neg\exists x.\theta \quad \text{iff} \quad (E, A) \models \forall x.\neg\theta$$
$$(E, A) \models \neg\forall x.\theta \quad \text{iff} \quad (E, A) \models \exists x.\neg\theta$$

In Definition 5, the first condition defines satisfaction for literals, the second to fourth conditions define satisfaction for conjunction, disjunction, and implication, respectively, the fifth to eighth conditions define satisfaction for negation, and the ninth to twelfth conditions define satisfaction for quantification. We extend satisfaction to a bistructure in the next definition.

Definition 6. Let E be a bistructure and θ an arbitrary formula.

$$E \models \theta \quad \text{iff} \quad \text{for all assignments } A, (E, A) \models \theta$$

Example 1. Let $\mathcal{L}$ contain the predicate symbols $P(2)$ and $Q(1)$, the function symbol $f(1)$, and the constant symbols c_1, c_2, and c_3. Let $E = (D, I^+, I^-)$ be such that $D = \{d_1, d_2, d_3\}$ and the interpretations I^+ and I^- are as follows.

$$I^+(c_1) = I^+(c_2) = d_1, \quad I^+(c_3) = d_3$$
$$I^+(f)(d_1) = d_2, \quad I^+(f)(d_2) = d_3, \quad I^+(f)(d_3) = d_3$$
$$I^+(P) = \{\langle d_1, d_3\rangle, \langle d_3, d_3\rangle\}, \quad I^-(P) = \{\langle d_1, d_3\rangle\}$$
$$I^+(Q) = \{\langle d_2\rangle\}, \quad I^-(Q) = \{\}$$

Here, we see that $\langle d_1, d_3\rangle$ is in both $I^+(P)$ and $I^-(P)$. Since, $I^+(c_1) = d_1$ and $I^+(c_3) = d_3$, we have by Definition 3 that $I^-(c_1) = d_1$ and $I^-(c_3) = d_3$. Hence, for all assignments A, we get $E \models P(c_1, c_3)$ and $E \models \neg P(c_1, c_3)$. In contrast, we see that $\langle d_3, d_3\rangle$ is in $I^+(P)$ but not in $I^-(P)$. Hence, for all assignments A, we get $E \models P(c_3, c_3)$ but not $E \models \neg P(c_3, c_3)$. Similarly, $\langle d_2\rangle$ is in $I^+(Q)$ but not in $I^-(Q)$, and $I^+(f)(d_1) = d_2$, and $I^+(c_1) = d_1$. Hence, for all assignments A, we get $E \models Q(f(c_1))$ but not $E \models \neg Q(f(c_1))$.

Next we define the concept of a model.

Definition 7. Let Δ be a set of formulae and let E be a bistructure. E is a *model* of Δ iff for all $\theta \in \Delta$, $E \models \theta$.

In the next section, we will consider classes of models for tolerant logic and then we will return to studying Tolerant Logic, in Section 8, where we will show how tolerant logic generalises Belnap's logic.

5. Classes of models for knowledgebases

We start with the class of all models for a knowledgebase (Definition 8) and then consider subclasses (Definitions 9–13) that will allow us to focus our analysis of inconsistency in knowledgebases using appropriate assumptions without having to add extra formulae to a knowledgebase or add further constraints on the semantics.

Definition 8. For a set of formulae Δ, $\mathsf{Models}(\Delta) = \{E \mid E \models \theta \text{ for all } \theta \in \Delta\}$.

Next we consider a definition that gives the models that satisfy the unique names assumption (UNA), meaning that different constant symbols are assigned to different objects in the domain. In other words, each constant is treated as a unique name.

Definition 9. Let Δ be a set of formulae.

$$\mathsf{UNAModels}(\Delta) = \{(D, I^+, I^-) \in \mathsf{Models}(\Delta) \mid \text{for all } c, c' \text{ if } c \neq c' \text{ then } I^+(c) \neq I^+(c')\}$$

Example 2. Let $E = (D, I^+, I^-)$ be such that $D = \{d_1, d_2, d_3\}$ and the interpretations I^+ and I^- are as follows.

$$I^+(c_1) = d_1, \quad I^-(c_2) = d_2, \quad I^+(c_3) = d_3$$
$$I^+(P) = \{\langle d_1, d_3\rangle, \langle d_3, d_3\rangle\}, \quad I^-(P) = \{\langle d_2, d_3\rangle\}$$

If $\Delta = \{P(c_1, c_3), \neg P(c_2, c_3), P(c_3, c_3)\}$, then $E \in \mathsf{UNAModels}(\Delta)$.

As an illustration of the utility of measures of inconsistency, we will consider in the next section some examples of knowledgebases that define sets and subsets of concepts (i.e. a form of ontological knowledge). For this, we will use the following class of models.

Definition 10. Let Δ be a set of formulae.

$$\begin{aligned}\mathsf{ConceptModels}(\Delta) = \big\{E \in \mathsf{Models}(\Delta) \mid\ & \\ & \text{for each formula of the form } \forall x.\alpha \to \beta \in \Delta, \\ & \text{for each assignment } A, \\ & \text{if } (E, A) \models \alpha \text{ then } (E, A) \models \beta\big\}\end{aligned}$$

Example 3. Let $E = (D, I^+, I^-)$ be such that $D = \{d_1, d_2, d_3\}$ and the interpretations I^+ and I^- are as follows.

$$\begin{aligned}& I^+(c_1) = d_1, \quad I^+(c_2) = d_2, \quad I^+(c_3) = d_3 \\ & I^+(P) = \big\{\langle d_1\rangle, \langle d_2\rangle\big\}, \quad I^-(P) = \big\{\langle d_3\rangle\big\} \\ & I^+(Q) = \big\{\langle d_1\rangle, \langle d_2\rangle, \langle d_3\rangle\big\}, \quad I^-(Q) = \{\}\end{aligned}$$

If $\Delta = \{P(c_1), \forall x.P(x) \to Q(x)\}$, then $E \in \mathsf{ConceptModels}(\Delta)$. Now consider E' which is the same as E except that $I^+(P) = \{\langle d_1\rangle\} = I^-(P)$ and $I^+(Q) = \{\langle d_2\rangle, \langle d_3\rangle\}$. So $E' \in \mathsf{Models}(\Delta) \setminus \mathsf{ConceptModels}(\Delta)$.

We will also consider the measurement of inconsistency in temporal knowledge. To facilitate this, we consider models that conform to particular flows of time. Often temporal knowledge is represented using linear time lines, isomorphic to some or all of the natural numbers or the real numbers, or more complex structures such as branching time lines. These structures raise particular difficulties for analysis in the case of inconsistency.

For modelling time flows, we need a predicate $t_1 \preceq t_2$ where t_1 is before t_2 in the flow of time. The language $\mathcal{L}$ may also contain additional predicate symbols as needed for the application. We also assume the following languages for use with time flows: $\mathcal{L}_k$ is the language that includes the $\preceq$ relation and the constant symbols for the sequence of natural numbers from 1 to k; $\mathcal{L}_p$ is the language that includes the $\preceq$ relation and the constant symbols for the natural numbers (positive integers); and $\mathcal{L}_i$ is the language that includes the $\preceq$ relation and the constant symbols for the integers.

Definition 11. Let $\Delta \subseteq \mathsf{Formulae}(\mathcal{L}_k)$ be a set of formulae.

$$\begin{aligned}\mathsf{FPModels}(\Delta) = \big\{\big(D, I^+, I^-\big) \in \mathsf{Models}(\Delta) \mid\ & D \text{ is the sequence of natural numbers from 1 to } k \\ & \text{and } \forall n \in \{1, \ldots, k\}(I^+(n) = n) \\ & \text{and } \forall c(I^+(c) \in \{1, \ldots, k\}) \\ & \text{and there is a predicate symbol } \preceq \\ & \quad \text{s.t. } I^+(\preceq) \text{ is the usual ordering over } \{1, \ldots, k\} \\ & \quad \text{and } I^-(\preceq) = \big(\{1, \ldots, k\} \times \{1, \ldots, k\}\big) \setminus I^+(\preceq)\big\}\end{aligned}$$

Definition 12. Let $\Delta \subseteq \mathsf{Formulae}(\mathcal{L}_p)$ be a set of formulae.

$$\begin{aligned}\mathsf{CPModels}(\Delta) = \big\{\big(D, I^+, I^-\big) \in \mathsf{Models}(\Delta) \mid\ & D = \mathbb{N} \\ & \text{and } \forall n \in \mathbb{N}(I^+(n) = n) \\ & \text{and } \forall c(I^+(c) \in \mathbb{N}) \\ & \text{and there is a predicate symbol } \preceq \\ & \quad \text{s.t. } I^+(\preceq) \text{ is the usual ordering over } \mathbb{N} \\ & \quad \text{and } I^-(\preceq) = \mathbb{N}^2 \setminus I^+(\preceq)\big\}\end{aligned}$$

Definition 13. Let $\Delta \subseteq \mathsf{Formulae}(\mathcal{L}_i)$ be a set of formulae.

$$\begin{aligned}\mathsf{CIModels}(\Delta) = \big\{\big(D, I^+, I^-\big) \in \mathsf{Models}(\Delta) \mid\ & D = \mathbb{Z} \\ & \text{and } \forall n \in \mathbb{Z}(I^+(n) = n) \\ & \text{and } \forall c(I^+(c) \in \mathbb{Z}) \\ & \text{and there is a predicate symbol } \preceq \\ & \quad \text{s.t. } I^+(\preceq) \text{ is the usual ordering over } \mathbb{Z} \\ & \quad \text{and } I^-(\preceq) = \mathbb{Z}^2 \setminus I^+(\preceq)\big\}\end{aligned}$$

We can regard $\mathsf{FPModels}(\Delta)$ as the models of Δ that are finite linear time models isomorphic to a subset of the natural numbers (and so FP stands for finite positive integer models), $\mathsf{CPModels}(\Delta)$ as the models of Δ that are linear time models isomorphic to the natural numbers (and so CP stands for countable positive integer models), and $\mathsf{CIModels}(\Delta)$ as the models of Δ that are linear time models isomorphic to the integers (and so CI stands for countable integer models).

In the following examples, we use the usual symbols for numbers for illustrating the elements of the domain and for use as constant symbols in the language. It may be desirable in some situations, to use a different symbol for a number in the domain and a number in the language, so that the difference between them is explicit.

Example 4. Let $E = (D, I^+, I^-)$ be such that $D = \{1, 2, 3, 4, 5, 6, 7\}$ and the interpretations I^+ and I^- are as follows.

$$I^+(1) = 1, \quad I^+(2) = 2, \quad I^+(3) = 3, \quad I^+(4) = 4, \quad I^+(5) = 5, \quad I^+(6) = 6, \quad I^+(7) = 7$$
$$I^+(P) = \{\langle 1\rangle, \langle 3\rangle, \langle 5\rangle, \langle 7\rangle\}$$
$$I^-(P) = \{\langle 2\rangle, \langle 4\rangle, \langle 6\rangle\}$$
$$I^+(\preceq) \text{ is the usual ordering over } \{1, \ldots, 7\}$$
$$I^-(\preceq) = (\{1, \ldots, 7\} \times \{1, \ldots, 7\}) \setminus I^+(\preceq)$$

If $\Delta = \{P(1), \neg P(2), P(3)\}$, then $E \in \mathsf{FPModels}(\Delta)$.

Example 5. Let $E = (D, I^+, I^-)$ be such that $D = \mathbb{N}$ and the interpretations I^+ and I^- are as follows.

$$I^+(n) = n, \quad \text{for all } n \in \mathbb{N}$$
$$I^+(P) = \{\langle n\rangle \mid n \in \mathbb{N}\}$$
$$I^-(P) = \{\langle n\rangle \mid n \in \mathbb{N}\}$$
$$I^+(\preceq) \text{ is the usual ordering over } \mathbb{N}$$
$$I^-(\preceq) = \mathbb{N}^2 \setminus I^+(\preceq)$$

If $\Delta = \{\forall x.P(x), \forall x.\neg P(x)\}$, then $E \in \mathsf{CPModels}(\Delta)$.

We can define further classes of models as required to capture for example continuous flows of time isomorphic to the real numbers and branching flows of time. We may also consider further constraints such as domain closure axioms (e.g. [35]). Whilst we have not considered equality in this paper, it is possible to either adapt the definition for tolerant logic to support an equality relation in the semantics (in which case, it may be appropriate to assume that for any knowledgebase, the equality relation is never both true and false), or a form of quasi-equality is introduced by axiomatisation (as proposed in [14]).

For every set of formulae Δ, $\mathsf{Models}(\Delta)$ is nonempty. Furthermore, for every $n \in \mathbb{N}$, and for every Δ, there is a model $E \in \mathsf{Models}(\Delta)$ such that $E = (D, I^+, I^-)$ and $|D| = n$. Even if the formulae in Δ involve many constant symbols, there may be an interpretation that assigns the same element in the domain to some or all of these constant symbols. It is only when we deal with special classes of models such as $\mathsf{UNAModels}(\Delta)$ that we eliminate these possibilities.

6. Framework for measuring inconsistency

For a bistructure $E = (D, I^+, I^-)$, let $\mathsf{Domain}(E) = D$. In general, we can consider two disjoint possibilities for $|\mathsf{Domain}(E)|$ for any bistructure E: These are that $|\mathsf{Domain}(E)|$ is finite or that $|\mathsf{Domain}(E)|$ is infinite. In the following, we will provide a framework that measures inconsistency for both these cases.

For the rest of the paper, a set of bistructures is called a *frame*. We adopt the following nomenclature for describing a frame Φ.

- Φ is a *bounded* frame iff $\exists n \in \mathbb{N}$ such that $\forall E \in \Phi$ $|\mathsf{Domain}(E)| \leqslant n$.
- Φ is an *unbounded* frame iff $\exists m \in \mathbb{N}\ \forall n \in \mathbb{N}$ ($n \geqslant m$ implies $\exists E \in \Phi$ $|\mathsf{Domain}(E)| = n$).
- Φ is an *infinite* frame iff $\exists E \in \Phi$ such that $|\mathsf{Domain}(E)| \geqslant \aleph_0$.

An example of a bounded frame is given by a singleton set containing just the bistructure given in Example 4. An example of an unbounded frame is given by $\mathsf{Models}(\Delta)$ when $\Delta = \{\forall x, y.P(x, y)\}$. An example of an infinite frame is given by a singleton set containing just the bistructure given in Example 5.

Obviously, if Φ is a bounded frame, then Φ is not an unbounded frame, and Φ is not an infinite frame. But it is possible that Φ is both an unbounded frame and an infinite frame.

For a frame Φ, any $\Phi' \subseteq \Phi$ is called a subframe. Obviously, if Φ is unbounded, there are subframes of Φ that are bounded. If Φ is an infinite frame, there may be a subframe that is unbounded, and there may be a subframe that is bounded. To support consideration of subframes, we draw on the following two subsidiary definitions.

$$\mathsf{Finite}(\Phi) = \{E \in \Phi \mid |\mathsf{Domain}(E)| \in \mathbb{N}\}$$
$$\mathsf{Infinite}(\Phi) = \{E \in \Phi \mid |\mathsf{Domain}(E)| \geqslant \aleph_0\}$$

Some of the key definitions in the rest of this paper will be based on analysing the bistructures in a frame. We introduce the notions of bounded, unbounded, and infinite frames to provide a general way of presenting our framework for measuring inconsistency. We will give definitions for a measure for inconsistency in a bistructure (Definition 16), a measure for inconsistency in a bounded frame (Definition 17), a measure for inconsistency in an unbounded frame (Definition 20), and a measure for inconsistency in an infinite frame (Definition 25). Normally, we expect each frame to be a set of models for a knowledgebase Δ, such as $\mathsf{Models}(\Delta)$, $\mathsf{UNAModels}(\Delta)$, $\mathsf{ConceptModels}(\Delta)$, or $\mathsf{CPModels}(\Delta)$. But since there are many possible classes of models that we could consider (Section 5 only considers some of the possible classes), it is simpler and more general to define our framework of measures in terms of frames rather than directly in terms of particular classes of models for knowledgebases.

6.1. Measuring inconsistency in a bistructure

We start by considering how to measure the inconsistency of a bistructure. We assume that we are given a language $\mathcal{L}$ and $E = (D, I^+, I^-)$ is a bistructure for $\mathcal{L}$. The nonlogical symbols considered in the definitions and examples are assumed to be in $\mathcal{L}$.

Definition 14. Let $E = (D, I^+, I^-)$ be a bistructure, and let Π be a set of predicate symbols.

$$\mathsf{CollisionCount}(\Pi, E) = \sum_{P_i \in \Pi} |\mathsf{Collision}(P_i, E)|$$

where $\mathsf{Collision}(P_i, E) = \{\langle d_1, \ldots, d_n\rangle \in D^n \mid \langle d_1, \ldots, d_n\rangle \in I^+(P_i) \cap I^-(P_i)\}$.

The following definition gives an upper bound on $\mathsf{CollisionCount}$ given a set of predicate symbols and a bistructure.

Definition 15. Let $E = (D, I^+, I^-)$ be a bistructure, and let Π be a set of predicate symbols.

$$\mathsf{UniverseCount}(\Pi, E) = \sum_{P_i \in \Pi} |\mathsf{Universe}(P_i, E)|$$

where $\mathsf{Universe}(P_i, E) = \{P_i(d_1, \ldots, d_n) \mid d_1, \ldots, d_n \in D$ and P_i is arity $n\}$.

Example 6. Let $E = (D, I^+, I^-)$ such that $D = \{1, 2, 3\}$ and

$$I^+(P) = \{\langle 1\rangle, \langle 2\rangle\} \quad I^+(Q) = \{\langle 1, 2\rangle, \langle 1, 3\rangle, \langle 2, 3\rangle\}$$
$$I^-(P) = \{\langle 3\rangle\} \quad I^-(Q) = \{\langle 1, 2\rangle, \langle 1, 3\rangle, \langle 3, 2\rangle\}$$

Hence, $\mathsf{CollisionCount}(\{P, Q\}, E) = 2$ since,

$$\mathsf{Collision}(P, E) = \{\}$$
$$\mathsf{Collision}(Q, E) = \{\langle 1, 2\rangle, \langle 1, 3\rangle\}$$

and $\mathsf{UniverseCount}(\{P, Q\}, E) = 12$ since,

$$\mathsf{Universe}(P, E) = \{\langle 1\rangle, \langle 2\rangle, \langle 3\rangle\}$$
$$\mathsf{Universe}(Q, E) = \{\langle 1, 1\rangle, \langle 1, 2\rangle, \langle 1, 3\rangle, \langle 2, 1\rangle, \langle 2, 2\rangle, \langle 2, 3\rangle, \langle 3, 1\rangle, \langle 3, 2\rangle, \langle 3, 3\rangle\}$$

In the definition of $\mathsf{UniverseCount}(\Pi, E)$, it does not make sense to let $\Pi = \emptyset$, and so for the rest of this paper, we assume that Π is always nonempty. Having Π as a parameter allows us to focus on particular subsets of predicate symbols during an analysis of a model or set of models. This is valuable, since when considering arbitrarily large domains, the value obtained by CollisionCount for some predicate symbols may "drown out" the CollisionCount for other predicate symbols.

We bring together the measure of CollisionCount and UniverseCount in the following definition for the measure of inconsistency for a set of predicate symbols in a bistructure with a finite domain.

Definition 16. Let Π be a set of predicate symbols and let E be a bistructure. The *bistructure degree of inconsistency* of (Π, E), denoted $\mathsf{Inc}(\Pi, E)$, is defined as follows: If $0 < \mathsf{UniverseCount}(\Pi, E) < \infty$, then

$$\mathsf{Inc}(\Pi, E) = \frac{\mathsf{CollisionCount}(\Pi, E)}{\mathsf{UniverseCount}(\Pi, E)}$$

otherwise $\mathsf{Inc}(\Pi, E) = 0$.

Example 7. Continuing Example 6, $\mathsf{Inc}(\Pi, E) = 2/12 = 1/6$, where $\Pi = \{P, Q\}$.

The following are some simple observations concerning the Inc function.

- For any Π and E, $0 \leqslant \mathsf{Inc}(\Pi, E) \leqslant 1$.
- For any Π and E, if $\mathsf{Inc}(\Pi, E) = 1$, then for all $\Pi' \subseteq \Pi$, $\mathsf{Inc}(\Pi', E) = 1$.
- For any Π and E, if $\mathsf{Inc}(\Pi, E) = 0$, then for all $\Pi' \subseteq \Pi$, $\mathsf{Inc}(\Pi', E) = 0$.

In tolerant logic for any knowledgebase there is always a model. This model can be obtained by taking each atom in the language of the knowledgebase, and letting the model satisfy the atom and the negation of the atom. This model gives the maximum degree of inconsistency, as formalised in the next proposition.

Proposition 1. *For all Δ, there is an $E \in \mathsf{Models}(\Delta)$, such that $\mathsf{Inc}(\Pi, E) = 1$ where Π is the set of all predicate symbols in Δ.*

Proof. Consider the bistructure E that for each n-ary $P_i \in \Pi$, and $d_1, \ldots, d_n \in D$, both $E \models P_i(d_1, \ldots, d_n)$ and $E \models \neg P_i(d_1, \ldots, d_n)$ hold. So $E \in \mathsf{Models}(\Delta)$ and $\mathsf{Inc}(\Pi, E) = 1$ because $\mathsf{CollisionCount}(\Pi, E) = \mathsf{UniverseCount}(\Pi, E)$. $\square$

6.2. Bounded degree of inconsistency

Now we consider the measure of inconsistency for a bounded frame. It is particularly useful if there is a maximum finite size for the intended models. Essentially, it takes a credulous point of view by using the bistructure, from the frame, with the minimum degree of inconsistency.

Definition 17. Let Π be a set of predicate symbols, and Φ be a bounded frame. The *bounded degree of inconsistency* of (Π, Φ), denoted $\mathsf{MicroInc}(\Pi, \Phi)$, is defined as follows.

$$\mathsf{MicroInc}(\Pi, \Phi) = \mathsf{Min}\big(\big\{\mathsf{Inc}(\Pi, E) \mid E \in \Phi\big\}\big)$$

We can use this measure for giving a measure of inconsistency for a knowledgebase Δ when there is a case for a bounded frame Φ that is in some sense representative of the knowledge. For example, if Δ is a set of ground literals, then a bounded frame containing just one model that satisfies exactly the literals in Δ is in a sense representative of the knowledge in Δ. The second reason we introduce the bounded degree of inconsistency is that we use it as part of the definition of the unbounded degree of inconsistency that we introduce in the next section.

Example 8. Let $\Pi = \{P\}$, $D = \{d_1, d_2, d_3, d_4\}$, and $\Phi = \{M_1, M_2\}$ where $M_1 = (D, I_1^+, I_1^-)$, and $M_2 = (D, I_2^+, I_2^-)$.

$$I_1^+(P) = \{\langle d_1\rangle, \langle d_2\rangle, \langle d_3\rangle, \langle d_4\rangle\} \quad I_1^-(P) = \{\langle d_4\rangle\}$$
$$I_2^+(P) = \{\langle d_1\rangle, \langle d_2\rangle, \langle d_3\rangle, \langle d_4\rangle\} \quad I_2^-(P) = \{\langle d_1\rangle, \langle d_2\rangle, \langle d_3\rangle, \langle d_4\rangle\}$$

Therefore,

$$\mathsf{CollisionCount}(\{P\}, M_1) = 1$$
$$\mathsf{CollisionCount}(\{P\}, M_2) = 4$$
$$\mathsf{UniverseCount}(\{P\}, M_1) = 4$$
$$\mathsf{UniverseCount}(\{P\}, M_2) = 4$$

So $\mathsf{Inc}(\{P\}, M_1) = 1/4$, and $\mathsf{Inc}(\{P\}, M_2) = 4/4$, and therefore $\mathsf{MicroInc}(\{P\}, \Phi) = 1/4$.

Example 9. Let $\Pi = \{P, Q\}$, $D = \{d_1, d_2\}$, and $\Phi = \{M_1, M_2, M_3\}$ where $M_1 = (D, I_1^+, I_1^-)$, $M_2 = (D, I_2^+, I_2^-)$, and $M_3 = (D, I_3^+, I_3^-)$.

$$I_1^+(P) = \{\langle d_1\rangle, \langle d_2\rangle\} \quad I_1^-(P) = \{\langle d_1\rangle, \langle d_2\rangle\}$$
$$I_1^+(Q) = \{\langle d_1\rangle\} \quad I_1^-(Q) = \{\langle d_1\rangle\}$$
$$I_2^+(P) = \{\langle d_1\rangle\} \quad I_2^-(P) = \{\langle d_1\rangle, \langle d_2\rangle\}$$
$$I_2^+(Q) = \{\langle d_2\rangle\} \quad I_2^-(Q) = \{\langle d_2\rangle\}$$
$$I_3^+(P) = \{\langle d_1\rangle, \langle d_2\rangle\} \quad I_3^-(P) = \{\langle d_1\rangle\}$$
$$I_3^+(Q) = \{\langle d_1\rangle, \langle d_2\rangle\} \quad I_3^-(Q) = \{\langle d_1\rangle, \langle d_2\rangle\}$$

Since, for $M_i \in \{M_1, M_2, M_3\}$, $\mathsf{UniverseCount}(\{P\}, M_i) = 2$, $\mathsf{UniverseCount}(\{Q\}, M_i) = 2$, and $\mathsf{UniverseCount}(\{P, Q\}, M_1) = 4$, we have

$$\mathsf{Inc}(\{P\}, M_1) = \frac{2}{2} \quad \mathsf{Inc}(\{P\}, M_2) = \frac{1}{2} \quad \mathsf{Inc}(\{P\}, M_3) = \frac{1}{2}$$
$$\mathsf{Inc}(\{Q\}, M_1) = \frac{1}{2} \quad \mathsf{Inc}(\{Q\}, M_2) = \frac{1}{2} \quad \mathsf{Inc}(\{Q\}, M_3) = \frac{2}{2}$$
$$\mathsf{Inc}(\{P, Q\}, M_1) = \frac{3}{4} \quad \mathsf{Inc}(\{P, Q\}, M_2) = \frac{2}{4} \quad \mathsf{Inc}(\{P, Q\}, M_3) = \frac{3}{4}$$

So $\mathsf{MicroInc}(\{P\}, \Phi) = 1/2$, $\mathsf{MicroInc}(\{Q\}, \Phi) = 1/2$, and $\mathsf{MicroInc}(\{P, Q\}, \Phi) = 1/2$.

We get $\mathsf{MicroInc}(\Pi, \Phi) = 0$ when Φ is a finite set of finite models for a knowledgebase that is consistent according to classical logic, and we get $\mathsf{MicroInc}(\Pi, \Phi) = 1$ when Φ is a finite set of finite models for a "completely inconsistent" knowledgebase (i.e. a knowledgebase for which each model of the knowledgebase, and for each atom in the language of the knowledgebase, the model satisfies the atom and its negation).

Proposition 2. *For all knowledgebases Δ, and for all sets of predicate symbols Π, if $\Delta \nvdash \bot$, and $\Phi = \{E \mid E \in \mathsf{Models}(\Delta)$ and $|\mathsf{Domain}(E)| \leqslant n$ for some $n \in \mathbb{N}\}$, then* $\mathsf{MicroInc}(\Pi, \Phi) = 0$.

Proof. Φ is a nonempty bounded frame. Since $\Delta \nvdash \bot$ there is an $E \in \Phi$ such that $\mathsf{Inc}(\Pi, E) = 0$. Hence, $\mathsf{MicroInc}(\Pi, \Phi) = 0$. □

The definitions of Inc and MicroInc are quite general definitions characterising inconsistency for several reasons: they actually support the use of diverse logics (not just tolerant logics) for generating the models of a knowledgebase, the definitions are based on frames rather than knowledgebases, and they are able to focus attention on particular predicates rather than all predicates used in the knowledgebase or language, thus providing a finer grained analysis of inconsistency.

6.3. Unbounded degree of inconsistency

Now we consider how to measure inconsistency in unbounded frames. An unbounded frame may contain infinite bistructures; however, in the measure we define we consider only the finite bistructures in the frame. In particular, this means that if the frame consists of the models of a knowledgebase, we restrict consideration to the finite models. In the following definition we identify a function that for each $n \in \mathbb{N}$ gives the bounded degree of inconsistency for the bistructures with domain of cardinality n.

Definition 18. Let Π be a set of predicate symbols, let Φ be an unbounded frame, and for each $n \in \mathbb{N}$, let $\Phi[n] = \{E \in \Phi \mid |\mathsf{Domain}(E)| = n\}$. The *discord function* for (Π, Φ) is a function $f_\Pi^\Phi : \mathbb{N} \mapsto [0,1]$ such that for each $n \in \mathbb{N}$,

$$\text{if } \Phi[n] \neq \emptyset, \quad \text{then } f_\Pi^\Phi(n) = \mathsf{MicroInc}\big(\Pi, \Phi[n]\big), \quad \text{otherwise } f_\Pi^\Phi(n) = 0.$$

So we can consider an unbounded frame as a sequence of bounded frames $\Phi[n]$ (each of which contains just bistructures of domain size n), then obtain the bounded degree of inconsistency of $\Phi[n]$ for each n, and then represent this sequence of values by a univariate function which we call a discord function.

Example 10. Let $\Delta = \{P(a), \neg P(a)\}$, $\Pi = \{P\}$, and $\Phi = \mathsf{Models}(\Delta)$. So, for all n, $f_\Pi^\Phi(n) = 1/n$.

Example 11. Let $\Delta = \{P(a), \exists x.\neg P(x)\}$, with $\Phi = \mathsf{Models}(\Delta)$, and $\Pi = \{P\}$. Hence, $f_\Pi^\Phi(1) = 1$, and for all $n > 1$, $f_\Pi^\Phi(n) = 0$ because in the second case we can always choose an element of the domain other than a, say b, for which $\neg P(b)$ holds and there is no inconsistency.

The next two examples illustrate the effect of imposing a restriction on the models considered for a knowledgebase.

Example 12. Let $\Delta = \{P(a), \neg P(a), P(b), \neg P(b), P(c), \neg P(c)\}$, with $\Phi = \mathsf{Models}(\Delta)$, and $\Pi = \{P\}$. In this case, the models in Φ with fewest conflicts are those where all constant symbols are assigned the same domain object. Hence for all n, $f_\Pi^\Phi(n) = 1/n$.

Example 13. Let $\Delta = \{P(a), \neg P(a), P(b), \neg P(b), P(c), \neg P(c)\}$, with $\Phi = \mathsf{UNAModels}(\Delta)$, and $\Pi = \{P\}$. In this case, because of the unique names assumption, the models in Φ assign each constant symbol a different domain object. Hence there are no models of cardinality 1 or 2 and we obtain $f_\Pi^\Phi(1) = 0$, $f_\Pi^\Phi(2) = 0$, and for all $n > 2$, $f_\Pi^\Phi(n) = 3/n$.

Whilst in general, the discord function f_Π^Φ is just a summary of the inconsistency arising in the bistructures in Φ, for some frames, the discord function is particularly interesting for our purposes. To consider some of these, we define the concept of a special type of rational function that we call a special function.

Definition 19. A *special function* is a function $r : \mathbb{N} \mapsto [0,1]$ of the following form where r_1 and r_2 are each a nonnegative and nondecreasing univariate polynomial function such that for all $n \in \mathbb{N}$ $r_2(n) \neq 0$, there is $k \in \mathbb{N}$ such that for all $n \geqslant k$, $0 \leqslant r_1(n) \leqslant r_2(n)$, and

$$r(n) = \frac{r_1(n)}{r_2(n)}$$

For some classes of frames, we can show that the discord function is a special function. In Examples 10 and 12 each discord function is a rational function. We give some further examples for discord functions below and in Fig. 1.

Now we define the unbounded degree of inconsistency as a special function for an important class of unbounded frames.

Definition 20. Let Π be a set of predicate symbols, Φ an unbounded frame, and let f_Π^Φ be the corresponding discord function. Suppose that there is a special function h_Π^Φ such that for all $n > n_0$, $f_\Pi^\Phi(n) = h_\Pi^\Phi(n)$. We call h_Π^Φ the *unbounded degree of inconsistency* for (Π, Φ). If an unbounded frame Φ for Π, written (Φ, Π), has an unbounded degree of inconsistency, we call it a *smooth frame*. We also let $\mathsf{Threshold}(h_\Pi^\Phi)$ be the lowest value for n_0 where $n_0 \in \mathbb{N} \cup \{0\}$.

		$\Pi_1 = \{P\}$	$\Pi_2 = \{Q\}$	$\Pi_3 = \{P, Q\}$
Δ_1	$\{P(a), \neg P(a) \vee \neg Q(a), Q(a)\}$	0	0	$\frac{1}{2n}$
Δ_2	$\{\forall x.P(x), \forall x.\neg P(x) \vee \neg Q(x), \forall x.Q(x)\}$	0	0	$\frac{1}{2}$
Δ_3	$\{\forall x.P(x) \wedge \neg P(x), \forall x.Q(x)\}$	1	0	$\frac{1}{2}$
Δ_4	$\{\forall x.P(x) \wedge \neg P(x), \forall x.Q(x) \wedge \neg Q(x)\}$	1	1	1
Δ_5	$\{P(a) \wedge \neg P(a), \forall x.Q(x)\}$	$\frac{1}{n}$	0	$\frac{1}{2n}$
Δ_6	$\{\forall x, y.P(x, y) \wedge \neg P(x, y), \forall x.Q(x)\}$	1	0	$\frac{n}{n+1}$
Δ_7	$\{\exists x, y.P(x, y) \wedge \neg P(x, y), \forall x, y.Q(x, y)\}$	$\frac{1}{n^2}$	0	$\frac{1}{2n^2}$
Δ_8	$\{P(a) \wedge \neg P(a), \forall x.Q(x) \wedge \neg Q(x)\}$	$\frac{1}{n}$	1	$\frac{n+1}{2n}$

Fig. 1. Examples of discord functions f_Π^Φ such that Φ is $\mathsf{Models}(\Delta_i)$ with different knowledgebases Δ_i and with different sets of predicate symbols Π_j. In each case, the discord function is a special function.

Examples 10–13 illustrate smooth frames. In particular, for Example 13, $h_\Pi^\Phi(n) = 3/n$ and $\mathsf{Threshold}(h_\Pi^\Phi) = 2$. The following example of an unbounded frame is not smooth.

Example 14. Let $\Phi = \{E_i \mid i \in \mathbb{N}\}$ and $\Pi = \{P(1)\}$, where $E_i = (D_i, I_i^+, I_i^-)$ such that $D_i = \{d_1, \ldots, d_i\}$, $I_i^+(P) = \{\langle d_1\rangle\}$ and for the odd values of i, $I_i^-(P) = \{\langle d_1\rangle\}$ while for the even values of i, $I_i^-(P) = \emptyset$. In this case for the odd values of n, $f_\Pi^\Phi(n) = 1/n$ and for the even values of n, $f_\Pi^\Phi(n) = 0$. Of course $1/n$ is a special function; however, there is no finite threshold to allow us to identify it with f_Π^Φ after the threshold value.

If Δ is consistent according to classical logic, and $\Phi = \mathsf{Models}(\Delta)$ then $h_\Pi^\Phi(n) = 0 = f_\Pi^\Phi(n)$, for all $n > 1$, because for any such $n \in \mathbb{N}$, either there is no model of size n or there is a model of size n with no collisions. We excluded the case of $n = 1$ because of examples such as the consistent theory $\Delta = \{\exists x.P(x), \exists x.\neg P(x)\}$ that has a collision in a model of size 1. In contrast, we can characterise a "maximally inconsistent" model as follows.

Proposition 3. *If for all $n \in \mathbb{N}$, $f_\Pi^\Phi(n) = 1 = h_\Pi^\Phi(n)$, then for each arity m predicate symbol $P \in \Pi$, and for all $E \in \Phi$,*

$$E \models \forall x_1, \ldots, x_m.P(x_1, \ldots, x_m) \wedge \neg P(x_1, \ldots, x_m)$$

Proof. Since $f_\Pi^\Phi(n)$ is always 1, every atom must be involved in a collision. □

To illustrate how the unbounded degree of inconsistency can be useful, we give two examples of ontologies presented in classical logic. As discussed in Section 2, we can adopt the following conventions: (1) A concept P is represented by a monadic predicate $P(x)$; (2) An individual c that is a member of a concept P is represented by a ground predicate $P(c)$; (3) The relationship that the concept Q is a subconcept of the concept P is represented by $\forall x.Q(x) \to P(x)$; and (4) The relationship that the concept Q is disjoint with the concept P is represented by $\forall x.Q(x) \to \neg P(x) \wedge \forall x.P(x) \to \neg Q(x)$.

Example 15. Let Δ be the following set of formulae where instance c is member of disjoint concepts.

$$\begin{array}{lll} \forall x.S(x) \to R(x) & \forall x.\big(Q(x) \to \neg S(x) \wedge S(x) \to \neg Q(x)\big) & S(c) \\ \forall x.Q(x) \to P(x) & \forall x.\big(P(x) \to \neg R(x) \wedge R(x) \to \neg P(x)\big) & Q(c) \end{array}$$

If $\Pi = \{P, Q, R, S\}$ and $\Phi = \mathsf{ConceptModels}(\Delta)$, then for all $E \in \Phi$,

$$\begin{array}{llll} E \models P(c) & E \models Q(c) & E \models R(c) & E \models S(c) \\ E \models \neg P(c) & E \models \neg Q(c) & E \models \neg R(c) & E \models \neg S(c) \end{array}$$

Hence, for all $E \in \Phi$, if $|E| = n$ then $\mathsf{Min}(\{\mathsf{CollisionCount}(\Pi, E)\}) = 4$ and $\mathsf{UniverseCount}(\Pi, E) = 4n$. Therefore $f_\Pi^\Phi(n) = 4/4n = 1/n$ for all $n \in \mathbb{N}$, and so $h_\Pi^\Phi(n) = 1/n$ with $\mathsf{Threshhold}(h_\Pi^\Phi) = 0$. Note, we get the same result if we replace $S(c)$ and $Q(c)$ by $\exists x.(S(x) \wedge Q(x))$ in Δ.

Example 16. Let Δ be the following set of formulae where (1) states that T and U are disjoint concepts, (2) states that T and U are the same concept, and (3) states that T and U comprise everything.

(1) $\forall x.(T(x) \rightarrow \neg U(x) \wedge U(x) \rightarrow \neg T(x))$
(2) $\forall x.(T(x) \rightarrow U(x) \wedge U(x) \rightarrow T(x))$
(3) $\forall x.T(x) \vee U(x)$

If $\Pi = \{T, U\}$, $E \in \Phi = \mathsf{ConceptModels}(\Delta)$, and $d \in \mathsf{Domain}(E)$, then according to (3) either $E \models T(d)$ or $E \models U(d)$ (or both). Assume w.l.g. that $E \models T(d)$. Then by (1) $E \models \neg U(d)$ and by (2) $E \models U(d)$. Applying (1) again yields $E \models \neg T(d)$. Hence, for all $E \in \Phi$, $\mathsf{Inc}(\Pi, E) = 2n/2n$. Therefore, $f_\Pi^\Phi(n) = h_\Pi^\Phi(n) = 1$, and $\mathsf{Threshold}(h_\Pi^\Phi) = 0$.

Example 17. Continuing Example 16, we extend Δ by adding the following set of formulae.

$$\forall x.S(x) \rightarrow R(x) \quad \forall x.\big(Q(x) \rightarrow \neg S(x) \wedge S(x) \rightarrow \neg Q(x)\big)$$
$$\forall x.Q(x) \rightarrow P(x) \quad \forall x.\big(P(x) \rightarrow \neg R(x) \wedge R(x) \rightarrow \neg P(x)\big)$$

Let $\Pi = \{P, Q, R, S, T, U\}$ and $\Phi = \mathsf{ConceptModels}(\Delta)$. Now for any $E \in \Phi$ and for any $d \in \mathsf{Domain}(E)$, it is possible to just make $E \models \neg P(d)$, $E \models \neg Q(d)$, $E \models \neg R(d)$, and $E \models \neg S(d)$ with no collisions for the predicates P, Q, R, and S. Therefore, $f_\Pi^\Phi(n) = h_\Pi^\Phi(n) = 2n/6n = 1/3$, and $\mathsf{Threshold}(h_\Pi^\Phi) = 0$.

As we showed in Example 14 not all unbounded frames are smooth. However, the following result shows that if we consider all the models of a knowledgebase Δ (i.e. $\mathsf{Models}(\Delta)$), or if we consider $\mathsf{UNAModels}(\Delta)$, or if we consider $\mathsf{ConceptModels}(\Delta)$, then there is an unbounded degree of inconsistency for any Π, and hence such a class of models is a smooth frame.

Theorem 1. *Let Δ be a knowledgebase, and let Π be a set of predicate symbols. If $\Phi = \mathsf{Models}(\Delta)$, or if $\Phi = \mathsf{UNAModels}(\Delta)$, or if $\Phi = \mathsf{ConceptModels}(\Delta)$, then (Π, Φ) is a smooth frame.*

Proof. We start with the case where $\Phi = \mathsf{Models}(\Delta)$ and Π is the set of all predicate symbols in Δ. It is known that for all $n \in \mathbb{N}$, $\Phi[n] \neq \emptyset$ (where $\Phi[n]$ contains the models of size n in Φ). What is needed is to show that the discord function f_Π^Φ is a special function after a possible threshold value, as explained in Definitions 19 and 20. Since for all $E \in \Phi[n]$, $\mathsf{UniverseCount}(\Pi, E)$ is the same, let $r_2(n) = \mathsf{UniverseCount}(\Pi, E)$ for some $E \in \Phi[n]$. Clearly, r_2 can never be 0 because the domains and Π are not empty. Also, let $r_1(n) = \mathsf{Min}(\{\mathsf{CollisionCount}(\Pi, E) | E \in \Phi[n]\})$. We have previously observed that $0 \leqslant \mathsf{Inc}(\Pi, E) \leqslant 1$ for all Π and E, hence $0 \leqslant r_1(n) \leqslant r_2(n)$ holds for all $n \in \mathbb{N}$. So we must show that both r_1 and r_2 are nondecreasing univariate polynomial functions. By definition both are univariate functions. It remains to show that both r_1 and r_2 are nondecreasing polynomial functions.

We start with r_2. We can actually compute $\mathsf{UniverseCount}(\Pi, E)$ for $E \in \Phi[n]$ as follows. Suppose Π contains the predicate symbols $P_1(m_1), \ldots, P_k(m_k)$ (the arities are in parentheses). Then $r_2(n) = \mathsf{UniverseCount}(\Pi, E) = n^{m_1} + \cdots + n^{m_k}$. This is a polynomial, although it may have to be simplified to write in standard form. Clearly, r_2 is nondecreasing.

Computing r_1 can be quite complex for an arbitrary Δ. What we need to show is that r_1 is nondecreasing and polynomial. It is clear that r_1 is nondecreasing after a threshold of $n = 1$ because as we enlarge a model by adding a domain element, the number of collisions cannot decrease. There may be a problem in the special case of going from 1 to 2 elements, for example, if $\Delta = \{\exists x.P(x), \exists x.\neg P(x)\}$. So it remains to show that r_1 is a polynomial function. We do not give all the details here but explain the basic idea through an example. At the end of this subsection we actually calculate r_1 for some special cases.

Consider the case where $\Pi = \{P(2)\}$ and there are no constant or function symbols in the language. Recall that to compute $r_1(n)$ we try to find the minimal number of collisions in a model of size n. We choose 4 statements in Δ that cause collisions, taking care of all quantifier combinations (for arity 2):

(1) $\exists x, y.P(x, y) \wedge \neg P(x, y)$

(2) $\exists x, \forall y. P(x, y) \wedge \neg P(x, y)$
(3) $\forall x, \exists y. P(x, y) \wedge \neg P(x, y)$
(4) $\forall x, y. P(x, y) \wedge \neg P(x, y)$

The minimal number of collisions for (1) is 1, for (2) and (3) is n, and for (4) is n^2. What happens is that for each argument an existential quantifier provides a multiplicative factor 1 while a universal quantifier provides a multiplicative factor n. This does not change even if there is a mix of different quantifiers, such as

(5) $\forall x, \exists y. P(x, y) \wedge \exists x, \forall y. \neg P(x, y)$

However, we may have to subtract overlaps and recognise when there are no collisions such as for

(6) $\exists x, y. P(x, y) \wedge \exists x, y. \neg P(x, y)$

(except for the case $n = 1$ that we have eliminated). The important point is that every number we get for collisions must be n^i for some i, where $0 \leqslant i \leqslant \mathsf{Arity}(P)$, including the number of elements in an overlap. Adding and subtracting such powers of n always yields a polynomial.

Now suppose that Π contains the predicate symbols $P_1(m_1), \ldots, P_k(m_k)$ as given above. Then for any P_j, $1 \leqslant j \leqslant k$, the minimal number of collisions is calculated as above for P except that now we get powers of n up to the largest arity of the predicate symbols. So far, implicitly we have restricted our analysis to conjunctions. In the case of disjunction, such as $\theta = \theta_1 \vee \cdots \vee \theta_t$ we take the minimum number of collisions in any of the θ_i, $1 \leqslant i \leqslant t$, while implication can be rewritten using disjunction (and negation). In all cases the calculation yields a sum of powers of n with subtractions, also powers of n for overlaps, and hence the result is a polynomial. A similar argument works if Π is a subset of the predicates in Δ.

Let us now consider the case where the language contains constant and function symbols. For the purpose of counting collisions we can always interpret a function symbol as a constant function in which case it has the effect of a constant symbol in counting collisions (as discussed in Section 2). So we need not deal separately with function symbols. But a constant symbol has the same effect as an existential quantifier, so for example,

(7) $\exists y. P(c, y) \wedge \neg P(c, y)$

gives the same number of minimal collisions as (1). This completes the proof for the case where $\Phi = \mathsf{Models}(\Delta)$.

We now show how this result extends to the other classes of types of models. Consider the case where $\Psi = \mathsf{ConceptModels}(\Delta)$. If for all $n \in \mathbb{N}$, $\Psi[n]$ contains a model in $\Phi[n]$ with a minimal number of collisions, then everything works as before. So let us consider how models of $\Phi[n]$ with a minimal number of collisions might not be in $\Psi[n]$. The following formula illustrates what might happen:

(8) $\forall x. P(x) \wedge \neg P(x) \wedge \neg Q(x) \wedge (P(x) \rightarrow Q(x))$

Consider a model E of Φ with minimal number of collisions, where for all $d \in \mathsf{Domain}(E)$, $\langle d \rangle \in I^+(P) \cap I^-(P) \cap I^-(Q)$ but $d \notin I^+(Q)$. There are n collisions. However $E \notin \Psi$. For E to satisfy the requirement for a concept model, d must also be in $I^+(Q)$. This requires adding n collisions. In the general case the number of collisions that must be added will again be a power of n, so r_1 is still a polynomial.

Finally, we consider the case where $\Gamma = \mathsf{UNAModels}(\Delta)$. Let n_Δ be the number of different constant symbols in Δ or 1 if there are none. Here the proof for $\Phi[n]$ goes through for $\Gamma[n]$, $n \geqslant n_\Delta$. Hence the same result holds but with $\mathsf{Threshold}(h_\Pi^\Gamma) = n_\Delta - 1$. $\square$

Using the proof of this theorem we can show that when the knowledgebase is exclusively ground formulae (i.e. there are no variables), then the numerator of the discord function is a constant.

Proposition 4. *Let Δ be a knowledgebase which incorporates no variable symbols, and let Π be a set of predicate symbols. If $\Phi = \mathsf{Models}(\Delta)$, or $\Phi = \mathsf{ConceptModels}(\Delta)$, or $\Phi = \mathsf{UNAModels}(\Delta)$, then there is an $m \in \mathbb{N}$, and $n_0 \in \mathbb{N}$, such that for all $n \in \mathbb{N}$, where $n > n_0$, the numerator of $h_\Pi^\Phi(n)$ is m.*

Proof. Clearly, Δ must have at least one constant symbol. Now recall from the proof of Theorem 1 that each constant symbol acts as an existential quantifier from the point of view of counting collisions. Also, each existential quantifier yields a multiplicative factor of 1. Hence there are no terms in the polynomial for r_1 (the numerator) with n^i for $i > 0$, so it must be a constant. We can actually get an upper bound on m as follows. If there are t predicate symbols then $m \leqslant t$ for $\Phi = \mathsf{Models}(\Delta)$ and $\Phi = \mathsf{ConceptModels}(\Delta)$, because we can identify all the constant symbols with a single element in the domain of the model and each predicate symbol causes at most one collision. Here $n_0 = 0$. The calculation is more complicated in the case where $\Phi = \mathsf{UNAModels}(\Delta)$ because all constant symbols must be interpreted as different elements of D. Suppose Δ contains c constant symbols and the predicate symbols are: $P_1(k_1), \ldots, P_q(k_q)$. Then the number of collisions for any $E \in \Phi$ of size $\geqslant c$ (there are no models of size $< c$) must be at most $c^{k_1} + \cdots + c^{k_q}$ whose sum is the upper bound for m with $n_0 = c - 1$. □

We can compare discord functions using the following ordering relation.

Definition 21. The *discord ordering*, denoted $\preceq$, is defined as follows, where f_Π^Φ and $f_{\Pi'}^{\Phi'}$ are discord functions.

$$f_\Pi^\Phi \preceq f_{\Pi'}^{\Phi'} \quad \text{iff} \quad \text{there is an } n' \in \mathbb{N} \text{ such that for all } n \in \mathbb{N}, \text{ if } n' \leqslant n, \text{ then } f_\Pi^\Phi(n) \leqslant f_{\Pi'}^{\Phi'}(n)$$

Intuitively, if $f_\Pi^\Phi \preceq f_{\Pi'}^{\Phi'}$, then (Φ, Π) is less or equally inconsistent with (Φ', Π').

Next we show by examples that the discord function is not monotonic or antimonotonic in general.

Example 18. Let $\Delta_2 = \{P(c)\}$, $\Pi = \{P\}$, and $\Delta_1 = \{P(c), \neg P(c)\}$ and $\Phi_i = \mathsf{Models}(\Delta_i)$ for $i = 1, 2$. Clearly $\Phi_1 \subset \Phi_2$ and for all $n \in \mathbb{N}$, $f_\Pi^{\Phi_2}(n) < f_\Pi^{\Phi_1}(n)$, so we cannot have $f_\Pi^{\Phi_1} \preceq f_\Pi^{\Phi_2}$. Now let $\Phi_4 = \mathsf{Models}(\Delta_1)$ for the Δ_1 given above and $\Phi_3 = \{E \mid E \in \Phi_4 \text{ and } |\mathsf{Domain}(E)| \text{ is even}\}$. Here, $\Phi_3 \subset \Phi_4$, but now $f_\Pi^{\Phi_4}(n) = 1/n$ for all $n \in \mathbb{N}$, while $f_\Pi^{\Phi_3}(n) = 1/n$ for all even n and 0 for all odd n. Hence it is not the case that $f_\Pi^{\Phi_4} \preceq f_\Pi^{\Phi_3}$.

This shows that in general, $\Phi_1 \subseteq \Phi_2$ implies neither $f_\Pi^{\Phi_1} \preceq f_\Pi^{\Phi_2}$ nor $f_\Pi^{\Phi_2} \preceq f_\Pi^{\Phi_1}$. Note that Φ_3 is not a smooth frame.

Example 19. Let Δ_2, Δ_3, Π_1, and Π_3 be the examples in Fig. 1. Here, $\Pi_1 \subset \Pi_3$. Let $\Phi_2 = \mathsf{Models}(\Delta_2)$ and $\Phi_3 = \mathsf{Models}(\Delta_3)$. We obtain for all $n \in \mathbb{N}$ $f_{\Pi_1}^{\Phi_2}(n) < f_{\Pi_3}^{\Phi_2}(n)$ and $f_{\Pi_3}^{\Phi_3}(n) < f_{\Pi_1}^{\Phi_3}(n)$. This shows that in general, $\Pi_1 \subseteq \Pi_2$ implies neither $f_{\Pi_1}^\Phi \preceq f_{\Pi_2}^\Phi$ nor $f_{\Pi_2}^\Phi \preceq f_{\Pi_1}^\Phi$.

We noted in Example 18 that Φ_3 is not a smooth frame. In fact, for smooth frames, the discord function is antimonotonic in Φ.

Proposition 5. *If Φ_1 and Φ_2 are both smooth frames and $\Phi_1 \subseteq \Phi_2$, then $f_\Pi^{\Phi_2} \preceq f_\Pi^{\Phi_1}$.*

Proof. Let $\Phi_1 \subseteq \Phi_2$ for smooth frames Φ_1 and Φ_2 where the corresponding special functions are $h_\Pi^{\Phi_1}$ and $h_\Pi^{\Phi_2}$ with $\mathsf{Threshold}(h_\Pi^{\Phi_1}) = n_1$ and $\mathsf{Threshold}(h_\Pi^{\Phi_2}) = n_2$. Let $n_0 = \max\{n_1, n_2\}$. Recalling that additional models cannot increase the minimum number of collisions for a $\Phi[n]$, we obtain for all $n > n_0$, $h_\Pi^{\Phi_2}(n) = f_\Pi^{\Phi_2}(n) \leqslant h_\Pi^{\Phi_1}(n) = f_\Pi^{\Phi_1}(n)$, hence $f_\Pi^{\Phi_2} \preceq f_\Pi^{\Phi_1}$. □

For a knowledgebase Δ, let $\mathsf{Con}(\Delta) = \{\Gamma \subseteq \Delta \mid \Gamma \nvdash \bot\}$ be the set of consistent subsets of Δ. The set of maximally consistent subsets of Δ is defined as follows.

$$\mathsf{MaxCon}(\Delta) = \big\{\Gamma \in \mathsf{Con}(\Delta) \mid \text{for all } \Gamma' \in \mathsf{Con}(\Delta)(\Gamma \not\subset \Gamma')\big\}$$

Also let $\mathsf{Free}(\Delta) = \bigcap \mathsf{MaxCon}(\Delta)$ be the set of formulae that are in all maximally consistent subsets of Δ. These may be regarded as the uncontroversial formulae in Δ since they do not appear in any minimally inconsistent subset of Δ

(where a minimally inconsistent subset of Δ is a subset Γ of Δ such that Γ is inconsistent and for all $\Gamma' \subset \Gamma$, Γ' is consistent).

The following result shows that if a formula θ is not involved in any inconsistency in $\Delta \cup \{\theta\}$, then adding θ to Δ cannot make the inconsistencies in Δ worse.

Proposition 6. *Let $\Phi_1 = \mathsf{Models}(\Delta)$, and $\Phi_2 = \mathsf{Models}(\Delta \cup \{\alpha\})$. If $\alpha \in \mathsf{Free}(\Delta \cup \{\alpha\})$, then $f_\Pi^{\Phi_2} \preceq f_\Pi^{\Phi_1}$. Furthermore, if α contains only nonlogical symbols in Δ then $f_\Pi^{\Phi_2} = f_\Pi^{\Phi_1}$.*

Proof. Let $\mathsf{Threshold}(h_\Pi^{\Phi_1}) = n_0^1$, and let $\mathsf{Threshold}(h_\Pi^{\Phi_2}) = n_0^2$. Since $\Delta \subseteq \Delta \cup \{\alpha\}$, $n_0^2 \geqslant n_0^1$. So for each $n > n_0^2$, $\mathsf{Min}(\{\mathsf{CollisionCount}(\Pi, E) \mid E \in \Phi_1[n]\} = \mathsf{Min}(\{\mathsf{CollisionCount}(\Pi, E) \mid E \in \Phi_2[n]\}$. Also for each $n > n_0^2$, for each $E_1 \in \Phi_1[n]$, and for each $E_2 \in \Phi_2[n]$, $\mathsf{UniverseCount}(\Pi, E_1) \leqslant \mathsf{UniverseCount}(\Pi, E_2)$. So for each $n \in \mathbb{N}$, if $n > n_0^2$, then $\mathsf{MicroInc}(\Pi, \Phi_2[n]) \leqslant \mathsf{MicroInc}(\Pi, \Phi_1[n])$. Hence, $f_\Pi^{\Phi_2} \preceq f_\Pi^{\Phi_1}$. In case α contains only nonlogical symbols in Δ, then we have $\mathsf{UniverseCount}(\Pi, E_1) = \mathsf{UniverseCount}(\Pi, E_2)$, and so $f_\Pi^{\Phi_2} = f_\Pi^{\Phi_1}$. $\square$

In the proof of Theorem 1 we gave a general argument to explain why the discord function is a special function for a smooth frame. Now we give specific results for some special cases.

Proposition 7. *Let ψ be a sentence of the form $Q_1x_1, \ldots, Q_kx_k.P(t_1, \ldots, t_m) \wedge \neg P(t_1, \ldots, t_m)$ where $\{x_1, \ldots, x_k\} \subseteq \{t_1, \ldots, t_m\}$, and $\{x_1, \ldots, x_k\} \neq \emptyset$, and each $t_i \in \{t_1, \ldots, t_m\} \setminus \{x_1, \ldots, x_k\}$ is a constant symbol, and $Q_1, \ldots, Q_k \in \{\forall, \exists\}$. If $\Phi = \mathsf{Models}(\{\psi\})$, and $\Pi = \{P\}$, then*

$$f_\Pi^\Phi(n) = \frac{d_1 \times \cdots \times d_k}{n^m}$$

where for each $d_i \in \{d_1, \ldots, d_k\}$, $d_i = n$ if Q_i is a universal quantifier, and $d_i = 1$ if Q_i is an existential quantifier.

Proof. If ψ is of the form $Q_1x_1, \ldots, Q_kx_k.P(t_1, \ldots, t_m) \wedge \neg P(t_1, \ldots, t_m)$, then for each n, there is an $E \in \Phi[n]$, with a minimal number of collisions, such that $\mathsf{CollisionCount}(\Pi, E) = d_1 \times \cdots \times d_k$, where for each $d_i \in \{d_1, \ldots, d_k\}$, $d_i = n$ if Q_i is a universal quantifier, because all n domain elements are involved in collisions, and $d_i = 1$ if Q_i is an existential quantifier, because one domain element must be involved in a collision. Furthermore, for all $E \in \Phi[n]$, $\mathsf{UniverseCount}(\Pi, E) = n^m$, therefore $f_\Pi^\Phi(n) = (d_1 \times \cdots \times d_k)/n^m$. $\square$

For example, consider $\psi = \forall x, y, \exists z.P(x, y, z) \wedge \neg P(x, y, z)$. So for $\Phi = \mathsf{Models}(\{\psi\})$, and $\Pi = \{P\}$, we get $f_\Pi^\Phi = n^2/n^3 = 1/n$. Also consider $\psi' = \forall y, \exists z.P(a, y, z) \wedge \neg P(a, y, z)$. So for $\Phi' = \mathsf{Models}(\{\psi'\})$, and $\Pi = \{P\}$, we get $f_\Pi^{\Phi'} = n/n^3 = 1/n^2$.

Corollary 1. *Let ψ be a sentence of the form $\forall x_1, \ldots, x_k.P(x_1, \ldots, x_k) \wedge \neg P(x_1, \ldots, x_k)$. If $\Phi = \mathsf{Models}(\{\psi\})$, and $\Pi = \{P\}$, then $f_\Pi^\Phi(n) = 1$.*

Corollary 2. *Let ψ be a sentence of the form $\exists x_1, \ldots, x_k(P(t_1, \ldots, t_m) \wedge \neg P(t_1, \ldots, t_m))$. If $\Phi = \mathsf{Models}(\{\psi\})$, and $\Pi = \{P\}$, then $f_\Pi^\Phi(n) = 1/n^m$.*

When ψ is a sentence of the form $Q_1x_1, \ldots, Q_kx_k.P(t_1, \ldots, t_m) \wedge \neg P(t_1, \ldots, t_m) \wedge P'(t'_1, \ldots, t'_m) \wedge \neg P'(t'_1, \ldots, t'_m)$ where $\{x_1, \ldots, x_k\} \subseteq \{t_1, \ldots, t_m, t'_1, \ldots, t'_m\}$ and $\{x_1, \ldots, x_k\} \neq \emptyset$, and each $t_i \in \{t_1, \ldots, t_m\}$ is either a variable symbol or a constant symbol, and $P \neq P'$, we can obtain the special function for the models of each of the following using Proposition 7 and then sum them.

$$Q_1x_1, \ldots, Q_kx_k.P(t_1, \ldots, t_m) \wedge \neg P(t_1, \ldots, t_m)$$
$$Q_1x_1, \ldots, Q_kx_k.P'(t'_1, \ldots, t'_m) \wedge \neg P'(t'_1, \ldots, t'_m)$$

For example, consider $\psi = \forall x, y, \exists z.P(a, y, z) \wedge \neg P(a, y, z) \wedge P'(x, y, z) \wedge \neg P'(x, y, z)$. So for $\Phi = \mathsf{Models}(\{\psi\})$, and $\Pi = \{P, P'\}$, we get $f_\Pi^\Phi = (n + n^2)/2n^3$.

Similarly when ψ is a sentence of the form $Q_1x_1, \ldots, Q_kx_k.P(t_1, \ldots, t_m) \wedge \neg P(t_1, \ldots, t_m) \wedge P(t'_1, \ldots, t'_m) \wedge \neg P(t'_1, \ldots, t'_m)$ where $\{x_1, \ldots, x_k\} \subseteq \{t_1, \ldots, t_m, t'_1, \ldots, t'_m\}$ and $\{x_1, \ldots, x_k\} \neq \emptyset$, and each $t_i \in \{t_1, \ldots, t_m\}$ is either a variable symbol or a constant symbol, we can obtain the special function for the models of each of the following using the above result, then obtain the special function by taking their sum minus their overlap.

$$Q_1x_1, \ldots, Q_kx_k.P(t_1, \ldots, t_m) \wedge \neg P(t_1, \ldots, t_m)$$
$$Q_1x_1, \ldots, Q_kx_k.P(t'_1, \ldots, t'_m) \wedge \neg P(t'_1, \ldots, t'_m)$$

For example, consider $\psi = \forall x, y, \exists z.P(a, y, z) \wedge \neg P(a, y, z) \wedge P(x, b, z) \wedge \neg P(x, b, z)$. So for $\Phi = \mathsf{Models}(\{\psi\})$, and $\Pi = \{P\}$, we get $f_\Pi^\Phi = (2n-1)/n^3$.

6.4. Inconsistency in the limit for smooth frames

We now consider the limit behaviour of the unbounded degree of inconsistency for smooth frames. In the limit, $\lim_{n\to\infty} f_\Pi^\Phi = \lim_{n\to\infty} h_\Pi^\Phi$, so it suffices to write only one of these functions. An advantage of considering the limit is that we can identify a measure of inconsistency that is a rational number in the [0, 1] interval. This provides a simple summary of the unbounded degree of inconsistency for a knowledgebase.

Theorem 2. *If f_Π^Φ is a discord function, and Φ is a smooth frame, then there is a rational number $k \in [0, 1]$ such that $\lim_{n\to\infty} f_\Pi^\Phi(n) = k$.*

Proof. Let f_Π^Φ be a discord function, and let Φ be a smooth frame. By definition, there is an n_0 such that for all $n > n_0$, f_Π^Φ is a special function. Therefore, there are polynomials $b_0 + b_1n + b_2n^2 + \cdots + b_pn^p$ and $c_0 + c_1n + c_2n^2 + \cdots + c_qn^q$, such that for all $n > n_0$,

$$f_\Pi^\Phi(n) = \frac{c_0 + c_1n + c_2n^2 + \cdots + c_qn^q}{b_0 + b_1n + b_2n^2 + \cdots + b_pn^p}$$

Since, $q = p$ or $q < p$, we have the following two cases.

$$\text{If } q = p, \quad \text{then } \lim_{n\to\infty} \frac{c_0 + c_1n + c_2n^2 + \cdots + c_pn^p}{b_0 + b_1n + b_2n^2 + \cdots + b_pn^p} = \frac{c_p}{b_p} \quad \text{where } c_p \leqslant b_p$$

$$\text{If } q < p, \quad \text{then } \lim_{n\to\infty} \frac{c_0 + c_1n + c_2n^2 + \cdots + c_qn^q}{b_0 + b_1n + b_2n^2 + \cdots + b_pn^p} = 0$$

Therefore, there exists a rational number $k \in [0, 1]$ such that $\lim_{n\to\infty} f_\Pi^\Phi(n) = k$. □

Example 20. Returning to Fig. 1, with $\Pi = \Pi_3$, the limits are as follows: for Δ_1, Δ_5, and Δ_7, $\lim_{n\to\infty} f_\Pi^\Phi = 0$, for Δ_2, Δ_3, and Δ_8, $\lim_{n\to\infty} f_\Pi^\Phi = 1/2$, and for Δ_4, and Δ_6, $\lim_{n\to\infty} f_\Pi^\Phi = 1$.

Following on from Theorem 2, when $\lim_{n\to\infty} f_\Pi^\Phi(n)$ is nonzero, we have the following characterisation of conflicts arising.

Proposition 8. *Let f_Π^Φ be a discord function and Φ a smooth frame such that for all $n > n_0$,*

$$f_\Pi^\Phi(n) = \frac{c_0 + c_1n + c_2n^2 + \cdots + c_pn^p}{b_0 + b_1n + b_2n^2 + \cdots + b_pn^p}$$

Then for all $E \in \Phi$, there is a $P \in \Pi$ such that $E \models \forall x_1, \ldots, x_p.P(x_1, \ldots, x_p) \wedge \neg P(x_1, \ldots, x_p)$.

Proof. Since f_Π^Φ is a discord function for a smooth frame, for all $n > n_0$, $f_\Pi^\Phi = h_\Pi^\Phi$ is a special function. Therefore, for all $n > n_0$, $\mathsf{MicroInc}(\Pi, \Phi[n])$ is a special function, and so, for all $n > n_0$, $\mathsf{Min}(\{\mathsf{Inc}(\Pi, E) \mid E \in \Phi[n]\})$ is a special function. This implies that for all $n > n_0$, the following is a special function:

$$\mathsf{Min}\left(\left\{\frac{\mathsf{CollisionCount}(\Pi, E)}{\mathsf{UniverseCount}(\Pi, E)} \;\middle|\; E \in \Phi[n]\right\}\right)$$

Since, for all n, for all $E, E' \in \Phi[n]$, $\mathsf{UniverseCount}(\Pi, E) = \mathsf{UniverseCount}(\Pi, E')$, we can rewrite the above as

$$\frac{\mathsf{Min}(\{\mathsf{CollisionCount}(\Pi, E) \mid E \in \Phi[n]\})}{\mathsf{UniverseCount}(\Pi, E)}$$

Thus,

$$\mathsf{Min}\big(\big\{\mathsf{CollisionCount}(\Pi, E) \mid E \in \Phi[n]\big\}\big) = c_0 + c_1 n + c_2 n^2 + \cdots + c_p n^p \tag{1}$$

$$\mathsf{UniverseCount}(\Pi, E) = b_0 + b_1 n + b_2 n^2 + \cdots + b_p n^p \tag{2}$$

Using the proof of Theorem 1, (2) implies that there is an arity p predicate symbol $P \in \Pi$, and therefore from (1) and the fact that $c_p \neq 0$ and the highest power of n is p, it follows that for all $E \in \Phi$, there is an arity p predicate symbol $P \in \Pi$, such that $E \models \forall x_1, \ldots, x_p.P(x_1, \ldots, x_p) \wedge \neg P(x_1, \ldots, x_p)$. □

The next result is basically the reverse of Theorem 2.

Theorem 3. *If $k \in [0, 1]$, and k is rational, then there is a smooth frame Φ with discord function f_Π^Φ such that $\lim_{n\to\infty} f_\Pi^\Phi(n) = k$.*

Proof. Let $k = s/t$ where $0 \leqslant s \leqslant t$. Let $r_1(n) = s \times n$ and $r_2(n) = t \times n$. Clearly, r_1 and r_2 are both nondecreasing and nonnegative univariate polynomial functions such that for all $n \in \mathbb{N}$ $(r_1(n)/r_2(n)) = k$. Now we construct a knowledgebase Δ, such that $\Phi = \mathsf{Models}(\Delta)$, and Π is the set of all predicate symbols used in Δ, so that

$$r_1(n) = \mathsf{Min}\big(\big\{\mathsf{CollisionCount}(\Pi, E) \mid E \in \Phi[n]\big\}\big)$$
$$r_2(n) = \mathsf{UniverseCount}(\Pi, E), \quad \text{when } E \in \Phi[n]$$

Let Π have t monadic predicate symbols. So for any n, and any $E \in \Phi[n]$,

$$\mathsf{UniverseCount}(\Pi, E) = n \times t$$

Then we continue the construction by putting s formulae into Δ of the form $\forall x.P_i(x) \wedge \neg P_i(x)$ using s different predicate symbols available in Π. So, for any n,

$$\mathsf{Min}\big(\big\{\mathsf{CollisionCount}(\Pi, E) \mid E \in \Phi[n]\big\}\big) = n \times s$$

Hence, for any n, $r_1(n) = n \times s$. By definition, $\mathsf{Min}(\{\mathsf{Inc}(\Pi, E) \mid E \in \Phi[n]\}) = \mathsf{MicroInc}(\Pi, \Phi[n]) = f_\Pi^\Phi(n)$. Since for all n, $(r_1(n)/r_2(n)) = s/t = k$, $\lim_{n\to\infty} f_\Pi^\Phi(n) = k$. □

Example 21. For $k = 2/5$, if $\Delta = \{\forall x.P_1(x) \wedge \neg P_1(x), \forall x.P_2(x) \wedge \neg P_2(x)\}$, and $\Pi = \{P_1, P_2, P_3, P_4, P_5\}$, all unary predicate symbols, then for each n, $r_1(n) = 2 \times n$ and $r_2(n) = 5 \times n$. Hence, $\lim_{n\to\infty} f_\Pi^\Phi(n) = 2/5$.

Proposition 9. *Let Δ be a knowledgebase that incorporates no variable symbols, and let Π be a set of predicate symbols. If $\Phi = \mathsf{Models}(\Delta)$, then $\lim_{n\to\infty} f_\Pi^\Phi(n) = 0$.*

Proof. If Δ is a knowledgebase which incorporates no variable symbols, and $\Phi = \mathsf{Models}(\Delta)$, then there is a $m \in \mathbb{N}$, such that for all $n \in \mathbb{N}$, the numerator of $h_\Pi^\Phi(n)$ is m (according to Proposition 4). Since, the denominator of $h_\Pi^\Phi(n)$ is a nondecreasing and nonnegative univariate polynomial function of n of degree at least 1, $h_\Pi^\Phi(n)$ converges to 0. □

So whilst (Π, Φ) may be inconsistent for any finite bistructure, in the limit it may be consistent (i.e. k may be 0). Intuitively, this means that the "inconsistent part" of the knowledge becomes "infinitely insignificant" in the limit. We use this result to give us the following nomenclature for a discord function f_Π^Φ.

Definition 22. For any Φ and Π, and for any $k \in [0, 1]$,

$$(\Pi, \Phi) \text{ is } k\textit{-inconsistent in the limit} \quad \text{iff} \quad \lim_{n\to\infty} f_\Pi^\Phi(n) = k.$$

So by Proposition 9, a knowledgebase without variable symbols is always 0-inconsistent in the limit.

Proposition 10. *Let Δ be a knowledgebase, Π a set of predicate symbols and $\Pi' \subseteq \Pi$ where Π' includes the predicate symbols of Π of the highest arity. If $\Phi = \mathsf{Models}(\Delta)$ then (Π, Φ) is k-inconsistent in the limit iff (Π', Φ) is k-inconsistent in the limit.*

Proof. By Definition 22, (Π, Φ) is k-inconsistent in the limit iff $\lim_{n\to\infty} f^{\Phi}_{\Pi}(n) = k$, where the discord function is f^{Φ}_{Π}. Similarly, there is a discord function $f^{\Phi}_{\Pi'}$ for (Π', Φ) such that $\lim_{n\to\infty} f^{\Phi}_{\Pi'}(n) = k'$. Let the highest arity of a predicate symbol in Π be p. Hence, there are polynomials $c_0 + c_1 n + c_2 n^2 + \cdots + c_q n^q$ and $b_0 + b_1 n + b_2 n^2 + \cdots + b_p n^p$ such that

$$f^{\Phi}_{\Pi}(n) = \frac{c_0 + c_1 n + c_2 n^2 + \cdots + c_q n^q}{b_0 + b_1 n + b_2 n^2 + \cdots + b_p n^p}$$

and there is a polynomial $e_0 + e_1 n + e_2 n^2 + \cdots + e_s n^s$ such that

$$f^{\Phi}_{\Pi'}(n) = \frac{e_0 + e_1 n + e_2 n^2 + \cdots + e_s n^s}{b_0 + b_1 n + b_2 n^2 + \cdots + b_r n^p}$$

since the power of the leading term for UniverseCount does not change.

There are two cases.

(Case 1) $k = 0$. This means that $q < p$ and hence $\Phi \not\models \forall x_1, \ldots, x_p . P(x_1, \ldots, x_p) \wedge \neg P(x_1, \ldots, x_p)$ for any arity p predicate symbol P. This remains the case if some predicate symbols are removed from Π, hence $k' = 0$.

(Case 2) $k > 0$. This means that $q = p$ and hence there are c_q predicate symbols $P_1, \ldots, P_{c_q}$, such that $\Phi \models \forall x_1, \ldots, x_p . (P_i(x_1, \ldots, x_p) \wedge \neg P_i(x_1, \ldots, x_p))$ for all i, $1 \leqslant i \leqslant c_q$. As long as all of these arity n predicate symbols stay in Π', we obtain $s = q = p$ and $c_q = e_s$. So $\lim_{n\to\infty} f^{\Phi}_{\Pi'}(n) = \lim_{n\to\infty} f^{\Phi}_{\Pi}(n) = k$, proving the result. □

Since we are primarily interested in measuring inconsistency in knowledgebases, the unbounded degree of inconsistency, in the form of a special function, is an efficient way of describing and analysing inconsistency in knowledgebases that have finite models. Furthermore, with the unbounded degree of inconsistency, we can categorise the models for a knowledgebase as being either consistent in the limit (i.e. $k = 0$) or inconsistent in the limit (i.e. $k > 0$). Though note that the discord function gives a finer distinction than the k in the limit: Consider for example two discord functions f and f' with the same limit but for which $f^{\Phi}_{\Pi} \preceq f'^{\Phi}_{\Pi}$ holds (such as $1/n^2$ and $1/n$).

6.5. Infinite degree of inconsistency

Now we turn to measuring inconsistency in bistructures with infinite domains. We have already argued that infinite domains arise in diverse applications in artificial intelligence and computer science, such as when reasoning about temporal knowledge. Unfortunately, when we consider infinite domains, we are unable to consider inconsistency as a ratio of the number of collisions in a model and the size of the universe. Therefore we need to take a more abstract approach for extending our framework to analyse infinite frames adequately. To this end, we introduce the macrotypes poset.

Definition 23. The *macrotypes poset* is a poset $(\leqslant_m, \{\infty^{\top}, \infty^{\perp}\})$ where $\infty^{\top} \leqslant_m \infty^{\perp}$ We call $\leqslant_m$ the *macrotype ordering* and we call $\{\infty^{\top}, \infty^{\perp}\}$ the *macrotypes*.

For each bistructure with an infinite domain, we assign a macrotype as follows.

Definition 24. Let Π be a set of predicate symbols and let E be a bistructure such that $|\mathsf{Domain}(E)| \geqslant \aleph_0$. The Mac function is defined as follows:

If $|\mathsf{Domain}(E)| \geqslant \aleph_0$, and $\mathsf{CollisionCount}(\Pi, E) < \aleph_0$, then $\mathsf{Mac}(\Pi, E) = \infty^{\top}$
If $|\mathsf{Domain}(E)| \geqslant \aleph_0$, and $\mathsf{CollisionCount}(\Pi, E) \geqslant \aleph_0$, then $\mathsf{Mac}(\Pi, E) = \infty^{\perp}$

Intuitively, for the macrotypes, the superscript $\top$ denotes the bistructure is “overwhelmingly consistent”, and the superscript $\perp$ denotes the bistructure is “substantially inconsistent”.

Example 22. Let $\Delta = \{\forall x.P(x), \forall x.\neg P(x)\}$, $\Pi = \{P\}$, and $E \in \mathsf{Models}(\Delta)$. If $\mathsf{Domain}(E) = \mathbb{N}$ or $\mathsf{Domain}(E) = \mathbb{R}$, then $\mathsf{Mac}(\Pi, E) = \infty^{\perp}$.

Example 23. Let $\Delta = \{\forall x.P(x), Q(a) \wedge \neg Q(a)\}$, $\Pi = \{P\}$, and $E \in \mathsf{Models}(\Delta)$. If $\mathsf{Domain}(E) = \mathbb{N}$ or $\mathsf{Domain}(E) = \mathbb{R}$, then $\mathsf{Mac}(\Pi, E) = \infty^{\top}$.

Now we can provide a measure of inconsistency for infinite frames. The definition takes a credulous view by using the bistructure in the frame with the minimum inconsistency according to the macrotype ordering.

Definition 25. Let Π be a set of predicate symbols, and Φ an infinite frame. Let $\mathsf{Infinite}(\Phi) = \{E \in \Phi \mid |D| \geqslant \aleph_0\}$, where D is the domain of E. The *infinite degree of inconsistency* of (Π, Φ), denoted $\mathsf{MacroInc}(\Pi, \Phi)$, is defined as follows.

For $E \in \mathsf{Infinite}(\Phi)$,
if $\forall E' \in \mathsf{Infinite}(\Phi)\big(\mathsf{Mac}(\Pi, E) \leqslant_m \mathsf{Mac}(\Pi, E')\big)$,
then $\mathsf{MacroInc}(\Pi, \Phi) = \mathsf{Mac}(\Pi, E)$

In order to better illustrate the measurement of the infinite degree of inconsistency, we consider some simple examples of knowledgebases for representing and reasoning with temporal knowledge.

Example 24. For $\Delta = \{\forall x.(x \preceq 5) \rightarrow \neg P(x), \forall x.P(x)\}$ and $\Pi = \{P\}$,

$$\mathsf{MacroInc}\big(\Pi, \mathsf{CPModels}(\Delta)\big) = \infty^{\top}$$
$$\mathsf{MacroInc}\big(\Pi, \mathsf{CIModels}(\Delta)\big) = \infty^{\perp}$$

The infinite degree of inconsistency is a function that is monotonic in one argument and antimonotonic in the other argument, as captured by the following results.

Proposition 11. *If $\Pi' \subseteq \Pi$, and Φ is an infinite frame, then* $\mathsf{MacroInc}(\Pi', \Phi) \leqslant_m \mathsf{MacroInc}(\Pi, \Phi)$.

Proof. Assume $\Pi' \subseteq \Pi$. So, for all $E \in \Phi$, $\mathsf{CollisionCount}(\Pi', E) \leqslant \mathsf{CollisionCount}(\Pi, E)$. Therefore, for all $E \in \mathsf{Infinite}(\Phi)$, $\mathsf{Mac}(\Pi', E) \leqslant_m \mathsf{Mac}(\Pi, E)$. Therefore, $\mathsf{MacroInc}(\Pi', \Phi) \leqslant_m \mathsf{MacroInc}(\Pi, \Phi)$. □

Proposition 12. *If $\Phi' \subseteq \Phi$, and Φ and Φ' are infinite frames, then* $\mathsf{MacroInc}(\Pi, \Phi) \leqslant_m \mathsf{MacroInc}(\Pi, \Phi')$.

Proof. Assume $\Phi' \subseteq \Phi$. For all $E' \in \Phi'$, there is an $E \in \Phi$ (namely E'), such that $\mathsf{CollisionCount}(\Pi, E) \leqslant \mathsf{CollisionCount}(\Pi, E')$. So for all $E' \in \mathsf{Infinite}(\Phi')$, there is an $E \in \mathsf{Infinite}(\Phi)$, such that $\mathsf{Mac}(\Pi, E) \leqslant_m \mathsf{Mac}(\Pi, E')$. Therefore, $\mathsf{MacroInc}(\Pi, \Phi) \leqslant_m \mathsf{MacroInc}(\Pi, \Phi')$. □

For knowledgebases Δ that are consistent by classical logic (i.e. $\Delta \nvdash \perp$) we have the following result for the infinite degree of inconsistency.

Proposition 13. *Let Δ be a knowledgebase and let Π be a set of predicate symbols. If $\Delta \nvdash \perp$, then* $\mathsf{MacroInc}(\Pi, \mathsf{Models}(\Delta)) = \infty^{\top}$.

Proof. If $\Delta \nvdash \perp$, then there is a model $E \in \mathsf{Infinite}(\mathsf{Models}(\Delta))$ such that $\mathsf{CollisionCount}(\Pi, E) = 0$. Hence, $\mathsf{MacroInc}(\Pi, \mathsf{Models}(\Delta)) = \infty^{\top}$. □

Similarly, if a knowledgebase is "completely inconsistent", then we have the following result for the infinite degree of inconsistency.

Proposition 14. *Let Δ be a knowledgebase and let Π be a set of predicate symbols. If for all $E \in \mathsf{Finite}(\mathsf{Models}(\Delta))$, $\mathsf{Inc}(\Pi, E) = 1$, then* $\mathsf{MacroInc}(\Pi, \mathsf{Models}(\Delta)) = \infty^{\perp}$.

Proof. Let $\Phi = \mathsf{Models}(\Delta)$. If for all $E \in \mathsf{Finite}(\Phi)$, $\mathsf{Inc}(\Pi, E) = 1$, then by Proposition 3, we must have for all $E \in \mathsf{Finite}(\Phi)$, $E \models \forall x_1, \ldots, x_k.P(x_1, \ldots, x_k) \wedge \neg P(x_1, \ldots, x_k)$ for all $P \in \Pi$. Hence, for all $E \in \mathsf{Infinite}(\Phi)$, $\mathsf{CollisionCount}(\Pi, E) \geqslant \aleph_0$. Therefore, $\mathsf{MacroInc}(\Pi, \mathsf{Models}(\Delta)) = \infty^{\perp}$. □

The following result shows that for any knowledgebase Δ, there is a "continuity" of the measurement of inconsistencies going from the subframe $\mathsf{Finite}(\mathsf{Models}(\Delta))$ to the subframe $\mathsf{Infinite}(\mathsf{Models}(\Delta))$.

Theorem 4. *Let Δ be a knowledgebase and let $\Phi = \mathsf{Models}(\Delta)$. If (Π, Φ) is k-inconsistent in the limit, and $k > 0$, then* $\mathsf{MacroInc}(\Pi, \Phi) = \infty^{\perp}$.

Proof. By Theorem 1, (Π, Φ) is a smooth frame. Let (Π, Φ) be k-inconsistent in the limit and let $k \in (0, 1]$. Therefore there exists a discord function $f_{\Pi}^{\Phi}(n)$ and polynomials $b_0 + b_1 n + b_2 n^2 + \cdots + b_p n^p$ and $c_0 + c_1 n + c_2 n^2 + \cdots + c_q n^q$, $q \leqslant p$, such that for all $n > n_0$,

$$h_{\Pi}^{\Phi}(n) = f_{\Pi}^{\Phi}(n) = \frac{c_0 + c_1 n + c_2 n^2 + \cdots + c_q n^q}{b_0 + b_1 n + b_2 n^2 + \cdots + b_p n^p} = \frac{r_1(n)}{r_2(n)}$$

where

$$k = \lim_{n \to \infty} \frac{c_0 + c_1 n + c_2 n^2 + \cdots + c_q n^q}{b_0 + b_1 n + b_2 n^2 + \cdots + b_p n^p}$$

Since $k > 0$, it follows that $q = p$. As shown in Proposition 8, the only way this is possible is if we have at least one arity p predicate symbol $P \in \Pi$ for which for all $E \in \Phi$, $E \models \forall x_1 \ldots x_p.P(x_1, \ldots, x_p) \wedge \neg P(x_1, \ldots, x_p)$. Now, if we consider any $E \in \mathsf{Infinite}(\mathsf{Models}(\Delta))$, then $\mathsf{CollisionCount}(\Pi, E) \geqslant \aleph_0$ and so $\mathsf{MacroInc}(\Pi, \Phi) = \infty^{\perp}$. □

However, if Φ is k-inconsistent in the limit, and $k = 0$, then it is not necessarily the case that $\mathsf{MacroInc}(\Pi, \Phi) = \infty^{\top}$, as illustrated by the following example.

Example 25. Consider a language that has a unary predicate P_1 and a binary predicate P_2. Let $\Delta = \{\forall x.P_1(x) \wedge \neg P_1(x)\}$. Here Φ is 0-inconsistent in the limit but $\mathsf{MacroInc}(\Pi, \Phi) = \infty^{\perp}$ for $\Pi = \{P_1, P_2\}$.

7. Framework for measuring consistency

We can extend our analysis of inconsistency by also measuring "harmonies". We view this as the complement to $\mathsf{CollisionCount}(\Pi, E)$, as explicated in the following definition for harmony.

Definition 26. Let E be a bistructure, where $D = \mathsf{Domain}(E)$ and let Π be a set of predicate symbols.

$$\mathsf{HarmonyCount}(\Pi, E) = \sum_{P_i \in \Pi} \left|\mathsf{Harmony}(P_i, E)\right|$$

where for each arity n predicate symbol P_i, $\mathsf{Harmony}(P_i, E) = D^n \setminus \mathsf{Collision}(P_i, E)$

We can now use $\mathsf{HarmonyCount}(\Pi, E)$ as a second dimension, along with $\mathsf{CollisionCount}(\Pi, E)$, to compare models. We can use this for both finite models and infinite models, but for finite models, with $\mathsf{CollisionCount}(\Pi, E)$, we can calculate $\mathsf{HarmonyCount}(\Pi, E)$, whereas for infinite models, we cannot calculate $\mathsf{HarmonyCount}(\Pi, E)$ from $\mathsf{CollisionCount}(\Pi, E)$.

Consider a finite model with 5 elements for a language with a unary predicate R_1 and a binary predicate R_2. There are 5 atoms for R_1 and 25 for R_2, altogether there are 30 atoms. Suppose that for this model there are 4 collisions. Then there must be $30 - 4 = 26$ "harmonies". There is no need to deal separately with the number 26 because if another finite model with 5 elements for the same language has 3 collisions, it must have 27 harmonies and the second dimension is irrelevant for comparing harmonies.

The situation is different for infinite models. Suppose an infinite model for the same language has infinitely many collisions. That does not tell us how many harmonies there are. Furthermore, suppose that D_1 has infinitely many collisions and 0 harmonies, while D_2 has infinitely many collisions and 5 harmonies and D_3 has infinitely many collisions and infinitely many harmonies. It would be very reasonable to say that D_3 is the least inconsistent model, D_2 is next, and D_1 is the most inconsistent. To capture this idea, we require the following notion of a profile.

For the rest of this section, we will only deal with infinite bistructures.

Definition 27. Let Π be a set of predicate symbols, and E a bistructure. The *profile* of (Π, E), denoted $\mathsf{Pro}(\Pi, E)$, is defined as follows.

$$\mathsf{Pro}(\Pi, E) = \big\langle \mathsf{CollisionCount}(\Pi, E), \mathsf{HarmonyCount}(\Pi, E)\big\rangle$$

Using the profile function, we define an ordering relation over profiles as follows.

Definition 28. Let Π be a set of predicate symbols, and E_1 and E_2 bistructures. If $\mathsf{Pro}(\Pi, E_1) = \langle a_1, b_1\rangle$ and $\mathsf{Pro}(\Pi, E_2) = \langle a_2, b_2\rangle$, then the *profile ordering*, denoted $\leqslant_\Pi$, is defined as follows.

$$E_1 \leqslant_\Pi E_2 \quad \text{iff} \quad a_1 \leqslant a_2 \text{ and } b_2 \leqslant b_1$$

Also let $E_1 <_\Pi E_2$ denote that $E_1 \leqslant_\Pi E_2$, and $E_2 \nleqslant_\Pi E_1$, and let $E_1 \sim_\Pi E_2$ denote that $E_1 \leqslant_\Pi E_2$, and $E_2 \leqslant_\Pi E_1$.

So if $E_1 \leqslant_\Pi E_2$, then E_1 is less inconsistent than, or equally inconsistent with, E_2.

Example 26. Let $E_1 = (D_1, I_1^+, I_1^-)$, $E_2 = (D_2, I_2^+, I_2^-)$, and $E_3 = (D_3, I_3^+, I_3^-)$ be such that $D_1 = D_2 = D_3 = \mathbb{N}$ and

$$\begin{array}{ll}
I_1^+(P) = \{\langle 1\rangle, \langle 2\rangle, \langle 3\rangle, \ldots\} & I_1^-(P) = \{\langle 1\rangle, \langle 2\rangle, \langle 3\rangle, \ldots\} \\
I_1^+(Q) = \{\langle 1\rangle, \langle 2\rangle\} & I_1^-(Q) = \{\} \\
I_2^+(P) = \{\langle 1\rangle, \langle 2\rangle, \langle 3\rangle, \ldots\} & I_2^-(P) = \{\langle 1\rangle, \langle 2\rangle, \langle 3\rangle, \ldots\} \\
I_2^+(Q) = \{\langle 1\rangle, \langle 2\rangle, \langle 3\rangle, \ldots\} & I_2^-(Q) = \{\langle 6\rangle, \langle 7\rangle, \langle 8\rangle, \ldots\} \\
I_3^+(P) = \{\langle 1\rangle, \langle 2\rangle, \langle 3\rangle, \ldots\} & I_3^-(P) = \{\langle 1\rangle, \langle 2\rangle, \langle 3\rangle, \ldots\} \\
I_3^+(Q) = \{\langle 1\rangle, \langle 2\rangle, \langle 3\rangle, \ldots\} & I_3^-(Q) = \{\langle 2\rangle, \langle 3\rangle, \langle 4\rangle, \ldots\}
\end{array}$$

Hence,

$$\begin{array}{ll}
\mathsf{CollisionCount}(\{P\}, E_1) = \aleph_0 & \mathsf{HarmonyCount}(\{P\}, E_1) = 0 \\
\mathsf{CollisionCount}(\{Q\}, E_1) = 0 & \mathsf{HarmonyCount}(\{Q\}, E_1) = \aleph_0 \\
\mathsf{CollisionCount}(\{P, Q\}, E_1) = \aleph_0 & \mathsf{HarmonyCount}(\{P, Q\}, E_1) = \aleph_0 \\
\mathsf{CollisionCount}(\{P\}, E_2) = \aleph_0 & \mathsf{HarmonyCount}(\{P\}, E_2) = 0 \\
\mathsf{CollisionCount}(\{Q\}, E_2) = \aleph_0 & \mathsf{HarmonyCount}(\{Q\}, E_2) = 5 \\
\mathsf{CollisionCount}(\{P, Q\}, E_2) = \aleph_0 & \mathsf{HarmonyCount}(\{P, Q\}, E_2) = 5 \\
\mathsf{CollisionCount}(\{P\}, E_3) = \aleph_0 & \mathsf{HarmonyCount}(\{P\}, E_3) = 0 \\
\mathsf{CollisionCount}(\{Q\}, E_3) = \aleph_0 & \mathsf{HarmonyCount}(\{Q\}, E_3) = 1 \\
\mathsf{CollisionCount}(\{P, Q\}, E_3) = \aleph_0 & \mathsf{HarmonyCount}(\{P, Q\}, E_3) = 1
\end{array}$$

So for $\Pi = \{P\}$, $E_1 \sim_\Pi E_2$ and $E_2 \sim_\Pi E_3$. And for $\Pi' = \{P, Q\}$, $E_1 <_{\Pi'} E_2$ and $E_2 <_{\Pi'} E_3$.

We now focus on a particular class of frames, called equiframes, that are defined next.

Definition 29. A frame Φ is an *equiframe* iff for all $E_i, E_j \in \Phi$, $\mathsf{Domain}(E_i) = \mathsf{Domain}(E_j)$.

In the following, we consider equiframes Φ for which each $E \in \Phi$ is such that $|\mathsf{Domain}(E)| = \aleph_0$, and give a definition for a profile for such equiframes. Some examples are given in Examples 27–30. We can consider alternative types of equiframes, for example equiframes Φ for which each $E \in \Phi$ is such that $\mathsf{Domain}(E) = [0, 1]$, in a similar way.

Definition 30. Let Π be a set of predicate symbols and let Φ be an equiframe of bistructures $\{E_i \mid i \in I\}$ where I is an index set and the domain cardinality is $\aleph_0$. Let $\mathsf{Pro}(\Pi, E_i) = \langle a_i, b_i \rangle$ for all $i \in I$. Let $a = \mathsf{Min}(\{a_i \mid i \in I\})$ and let

$$b = \begin{cases} \mathsf{Max}(\{b_i \mid i \in I\}) & \text{if it exists} \\ \aleph_0 & \text{otherwise} \end{cases}$$

The *profile* of (Π, Φ) is defined as $\mathsf{Pro}(\Pi, \Phi) = \langle a, b \rangle$.

Example 27. For $\Delta_1 = \{\forall x.P(x), \forall x.(x \preceq 6) \rightarrow \neg P(x)\}$, let $\Phi_1 = \mathsf{CPModels}(\Delta_1)$. So $\mathsf{Pro}(\{P\}, \Phi_1) = \langle 6, \aleph_0 \rangle$.

Example 28. For $\Delta_2 = \{\forall x.P(x), \forall x.\neg P(x)\}$, let $\Phi_2 = \mathsf{CPModels}(\Delta_2)$. So $\mathsf{Pro}(\{P\}, \Phi_2) = \langle \aleph_0, 0 \rangle$.

Example 29. For $\Delta_3 = \{\forall x.P(x), \forall x.Q(x) \wedge \neg Q(x)\}$, let $\Phi_3 = \mathsf{CPModels}(\Delta_3)$.

$$\mathsf{Pro}\big(\{P\}, \Phi_3\big) = \langle 0, \aleph_0 \rangle$$
$$\mathsf{Pro}\big(\{Q\}, \Phi_3\big) = \langle \aleph_0, 0 \rangle$$
$$\mathsf{Pro}\big(\{P, Q\}, \Phi_3\big) = \langle \aleph_0, \aleph_0 \rangle$$

Example 30. For $\Delta_4 = \{\forall x.P(x)\}$, let $\Phi_4 = \mathsf{CPModels}(\Delta_4)$. So $\mathsf{Pro}(\{P\}, \Phi_4) = \langle 0, \aleph_0 \rangle$.

Example 31. Let Φ_5 be the frame $\{E_i \mid i \in \mathbb{N}\}$ where $E_i = (\mathbb{N}, I_i^+, I_i^-)$ for $i = 1, 2, 3, \ldots$ defined by $I_i^+(P) = \{\langle 1 \rangle, \langle 2 \rangle, \langle 3 \rangle, \ldots\}$ and by $I_i^-(P) = \{\langle i+1 \rangle, \langle i+2 \rangle, \langle i+3 \rangle, \ldots\}$. This means that for all $i = 1, 2, 3, \ldots$ $\mathsf{CollisionCount}(\{P\}, E_i) = \aleph_0$ and $\mathsf{HarmonyCount}(\{P\}, E_i) = i$. This is where we need to use the second line in the definition of b in Definition 30, and find that $\mathsf{Pro}(\{P\}, \Phi_5) = \langle \aleph_0, \aleph_0 \rangle$, even though no E_i has infinitely many harmonies.

We can compare profiles using the following profile ranking.

Definition 31. If $\mathsf{Pro}(\Pi_1, \Phi_1) = \langle a_1, b_1 \rangle$, and $\mathsf{Pro}(\Pi_2, \Phi_2) = \langle a_2, b_2 \rangle$, then the *profile ranking* is defined as follows:

$$\mathsf{Pro}(\Pi_1, \Phi_1) \preceq \mathsf{Pro}(\Pi_2, \Phi_2) \quad \text{iff} \quad a_1 \leqslant a_2 \text{ and } b_2 \leqslant b_1$$

Let $\mathsf{Pro}(\Pi_1, \Phi_1) \prec \mathsf{Pro}(\Pi_2, \Phi_2)$ denote that $\mathsf{Pro}(\Pi_1, \Phi_1) \preceq \mathsf{Pro}(\Pi_2, \Phi_2)$ and $\mathsf{Pro}(\Pi_2, \Phi_2) \not\preceq \mathsf{Pro}(\Pi_1, \Phi_1)$ and $\mathsf{Pro}(\Pi_1, \Phi_1) \sim \mathsf{Pro}(\Pi_2, \Phi_2)$ denote that $\mathsf{Pro}(\Pi_1, \Phi_1) \preceq \mathsf{Pro}(\Pi_2, \Phi_2)$ and $\mathsf{Pro}(\Pi_2, \Phi_2) \preceq \mathsf{Pro}(\Pi_1, \Phi_1)$.

Clearly, $\preceq$ defines a total order.

Example 32. Using Examples 27–30, we get the following profile ranking.

$$\begin{aligned} &\mathsf{Pro}\big(\{P\}, \Phi_3\big) \sim \mathsf{Pro}\big(\{P\}, \Phi_4\big) \prec \mathsf{Pro}\big(\{P\}, \Phi_1\big) \\ &\qquad \prec \mathsf{Pro}\big(\{P, Q\}, \Phi_3\big) \sim \mathsf{Pro}\big(\{P\}, \Phi_5\big) \prec \mathsf{Pro}\big(\{Q\}, \Phi_3\big) \sim \mathsf{Pro}\big(\{P\}, \Phi_2\big) \end{aligned}$$

In general, for an equiframe Φ that is a set of models from a knowledgebase Δ, such as $\mathsf{CPModels}(\Delta)$, $\mathsf{Pro}(\Pi, \Phi)$ is minimal in the $\preceq$ ranking when Δ is consistent, and $\mathsf{Pro}(\Pi, \Phi)$ is maximal in the $\preceq$ ranking when for each $E \in \Phi$, for all predicates $P \in \Pi$, $E \models \forall x_1, \ldots, x_k.P(x_1, \ldots, x_k) \wedge \neg P(x_1, \ldots, x_k)$.

Proposition 15. *For equiframes Φ_1 and Φ_2, if $\Phi_1 \subseteq \Phi_2$, then* $\mathsf{Pro}(\Pi, \Phi_2) \preceq \mathsf{Pro}(\Pi, \Phi_1)$.

Proof. Let $\mathsf{Pro}(\Pi, \Phi_1) = \langle a_1, b_1 \rangle$ and $\mathsf{Pro}(\Pi, \Phi_2) = \langle a_2, b_2 \rangle$. Also let I_1 be the index set for Φ_1 and let I_2 be the index set for Φ_2. Since $\Phi_1 \subseteq \Phi_2$, we have that $I_1 \subseteq I_2$. So by Definition 30, $a_2 \leqslant a_1$ and $b_1 \leqslant b_2$. Therefore, $\mathsf{Pro}(\Pi, \Phi_2) \preceq \mathsf{Pro}(\Pi, \Phi_1)$. $\square$

So increasing an equiframe Φ monotonically decreases $\mathsf{Pro}(\Pi, \Phi)$ in the $\preceq$ ordering. However increasing Π neither monotonically increases nor decreases $\mathsf{Pro}(\Pi, \Phi)$ in the $\preceq$ ordering, as we show by referring to Example 32 for Φ_3 where $\mathsf{Pro}(\{P\}, \Phi_3) \prec \mathsf{Pro}(\{P, Q\}, \Phi_3) \prec \mathsf{Pro}(\{Q\}, \Phi_3)$.

The notions of harmony count, and of a profile, provide a potentially useful addition to our framework for analysing inconsistent first-order knowledgebases. As we discussed above, for finite domains, we can determine the harmonies for a bistructure from the collisions, but for infinite domains this is not possible. So for finite domains, harmony count can be made explicit, and for infinite domains, harmony count provides further useful information for evaluating and comparing inconsistent knowledgebases as illustrated in Examples 27–31. Furthermore, using the profile ranking (as illustrated in Example 32), we may for example choose a knowledgebase for which its set of models are minimal in the profile ranking, if we are seeking better (i.e. less conflicting and more harmonious) sources of information.

8. Tolerant logic generalises Belnap's logic

We now compare our presentation of tolerant logic with Belnap's four-valued logic. We only deal with the ground portion of FOL, because Belnap's four-valued logic is propositional. Let $\mathcal{G} \subseteq \mathsf{Formulae}(\mathcal{L})$ be the set of ground formulae in $\mathcal{L}$ that involve the $\neg$, $\vee$ and $\wedge$ symbols. So if $\theta \in \mathcal{G}$, then θ contains no variable symbols nor any quantifier symbols. An interpretation in Belnap's four-valued logic is a truth assignment t that for the atoms of $\mathcal{G}$ assigns a value in $\{T, F, N, B\}$, and for an arbitrary formula $\theta \in \mathcal{G}$, $t(\theta)$ is defined by the truth tables for Belnap's four-valued logic given in Tables 1 to 3. The lattice ordering is given as $B > T$, $B > F$, $T > N$, and $F > N$. In particular, $t(\theta) \geqslant T$ iff $t(\theta) = B$ or $t(\theta) = T$.

Definition 32. Let t be a Belnap truth assignment, $t : \mathcal{G} \mapsto \{T, F, N, B\}$, and let E be a bistructure. t *represents* E for all atoms in $\mathcal{G}$ iff for all atoms $\alpha \in \mathcal{G}$, the following constraints hold for t and E.

$$\begin{array}{lll} t(\alpha) = N & \text{iff} & E \not\models \alpha \text{ and } E \not\models \neg\alpha \\ t(\alpha) = F & \text{iff} & E \not\models \alpha \text{ and } E \models \neg\alpha \\ t(\alpha) = T & \text{iff} & E \models \alpha \text{ and } E \not\models \neg\alpha \\ t(\alpha) = B & \text{iff} & E \models \alpha \text{ and } E \models \neg\alpha \end{array}$$

For $\theta \in \mathcal{G}$, the satisfaction relation for tolerant logic, given by $\models$, coincides with the semantics for Belnap's four-valued logic as shown by the following result.

Table 1
Truth table for negation

α	N	F	T	B
$\neg\alpha$	N	T	F	B

Table 2
Truth table for conjunction

$\wedge$	N	F	T	B
N	N	F	N	F
F	F	F	F	F
T	N	F	T	B
B	F	F	B	B

Table 3
Truth table for disjunction

$\vee$	N	F	T	B
N	N	N	T	T
F	N	F	T	B
T	T	T	T	T
B	T	B	T	B

Theorem 5. *Let t be a Belnap truth assignment $t : \mathcal{G} \mapsto \{T, F, N, B\}$, and let E be a bistructure such that t represents E for $\mathcal{A}$. For $\theta \in \mathcal{G}$,*

$$t(\theta) \geqslant T \quad \textit{iff} \quad E \models \theta$$

Proof. By induction on the length of a formula. The base case is where θ is a literal. Then $t(\theta) \geqslant T$ iff $E \models \theta$ by Definition 32. For the induction we assume the inductive hypothesis that $t(\alpha) \geqslant T$ iff $E \models \alpha$ and $t(\beta) \geqslant T$ iff $E \models \beta$. The following 5 cases exhaust all the possibilities and complete the proof.

Case 1: θ is of the form $\alpha \wedge \beta$.

$$\begin{aligned} t(\alpha \wedge \beta) \geqslant T \quad & \text{iff} \quad t(\alpha) \geqslant T \text{ and } t(\beta) \geqslant T \text{ by Table 2} \\ & \text{iff} \quad E \models \alpha \text{ and } E \models \beta \text{ by the inductive hypothesis} \\ & \text{iff} \quad E \models \alpha \wedge \beta \text{ by Definition 5} \end{aligned}$$

Case 2: θ is of the form $\alpha \vee \beta$.

$$\begin{aligned} t(\alpha \vee \beta) \geqslant T \quad & \text{iff} \quad t(\alpha) \geqslant T \text{ or } t(\beta) \geqslant T \text{ by Table 3} \\ & \text{iff} \quad E \models \alpha \text{ or } E \models \beta \text{ by the inductive hypothesis} \\ & \text{iff} \quad E \models \alpha \vee \beta \text{ by Definition 5} \end{aligned}$$

Case 3: θ is of the form $\neg(\alpha \wedge \beta)$.

$$\begin{aligned} t\big(\neg(\alpha \wedge \beta)\big) \geqslant T \quad & \text{iff} \quad t(\alpha \wedge \beta) \geqslant F \text{ by Table 1} \\ & \text{iff} \quad t(\alpha) \geqslant F \text{ or } t(\beta) \geqslant F \text{ by Table 2} \\ & \text{iff} \quad t(\neg\alpha) \geqslant T \text{ or } t(\neg\beta) \geqslant T \text{ by Table 1} \\ & \text{iff} \quad E \models \neg\alpha \text{ or } E \models \neg\beta \text{ by the inductive hypothesis} \\ & \text{iff} \quad E \models \neg(\alpha \wedge \beta) \text{ by Definition 5} \end{aligned}$$

Case 4: θ is of the form $\neg(\alpha \vee \beta)$.

$$\begin{aligned} t\big(\neg(\alpha \vee \beta)\big) \geqslant T \quad & \text{iff} \quad t(\alpha \vee \beta) \geqslant F \text{ by Table 1} \\ & \text{iff} \quad t(\alpha) \geqslant F \text{ and } t(\beta) \geqslant F \text{ by Table 3} \\ & \text{iff} \quad t(\neg\alpha) \geqslant T \text{ and } t(\neg\beta) \geqslant T \text{ by Table 1} \\ & \text{iff} \quad E \models \neg\alpha \text{ and } E \models \neg\beta \text{ by the inductive hypothesis} \\ & \text{iff} \quad E \models \neg(\alpha \vee \beta) \text{ by Definition 5} \end{aligned}$$

Case 5: θ is of the form $\neg\neg\alpha$.

$$\begin{aligned} t(\neg\neg\alpha) \geqslant T \quad & \text{iff} \quad t(\alpha) \geqslant T \text{ by Table 1} \\ & \text{iff} \quad E \models \alpha \text{ by the inductive hypothesis} \\ & \text{iff} \quad E \models \neg\neg\alpha \text{ by Definition 5} \qquad \square \end{aligned}$$

Whilst Belnap's four-valued logic is a simple, intuitive and well-known proposal, there are some interesting, though more complex proposals, that are variants of Belnap's proposal, such as a proposal by Arieli and Avron that has a preferential semantics which selects those models for a knowledgebase that are minimal with respect to the assignment of the B truth value [1]. These variants on Belnap's proposal may offer some useful developments of the framework for analysing inconsistent information presented in this paper.

9. Discussion

The need to develop robust, but principled, logic-based techniques for analysing inconsistent information is increasingly recognised as an important research area for artificial intelligence in particular, and for computer science in general. This interest stems from the recognition that the dichotomy between consistent and inconsistent set of formulae that comes from classical logics is not sufficient for describing inconsistent information.

A number of proposals have been made for measuring the degree of information in the presence of inconsistency [24,26,29,37], and for measuring the degree of inconsistency in information [14,15,18,20–22,24–26]. For a review see [17].

These measures are potentially important in diverse applications in artificial intelligence, such as belief revision, belief merging, negotiation, multi-agent systems, decision-support, and software engineering tools. Already, measuring inconsistency has been seen to be a useful tool in analysing a diverse range of information types including news reports [23], integrity constraints [14], information merging [34], ontologies [32], software specifications [8,30], and ecommerce protocols [10].

The current proposals for measuring inconsistency can be classified in two ways. The first approach involves "counting" the minimal number of formulae needed to produce the inconsistency. The more formulae needed to produce the inconsistency, the less inconsistent the set [25]. This idea is an interesting one, but it rejects the possibility of a more fine-grained inspection of the (content of the) formulae. In particular, if one looks to singleton sets only, one is back to the initial problem, with only two values: consistent or inconsistent.

The second approach (which includes the proposal presented in this paper) involves looking at the proportion of the language that is touched by the inconsistency. This allows us to look *inside* the formulae [14,20,24]. This means that two formulae (singleton sets) can have different inconsistency measures. In these proposals one can identify the set of formulae with its conjunction (i.e. the set $\{\varphi, \varphi'\}$ has the same inconsistency measure as the set $\{\varphi \wedge \varphi'\}$). This means that the distribution of the contradiction among the formulae is not taken into account.

Recently, there has been a proposal to combine the first and second approaches in a unified framework [18]. The framework, based on coalitional game theory, supports inconsistency measures that are able to look inside the formulae, but also to take into account the distribution of the contradiction among the different formulae of the set, allowing for the identification of the blame/responsibility of each formula of the knowledgebase in the inconsistency.

All the proposals discussed above are based on propositional logic, apart from [15], which deals mainly with infinite models, and [14] which is based on a restricted form of first-order logic. So the proposal in this paper, based on tolerant logic, is the first proposal for measuring inconsistency in first-order knowledge. This potentially offers considerable advantages for applications in artificial intelligence and computer science involving first-order logic.

Acknowledgement

The authors are very grateful to the anonymous reviewers of this paper as well as Yue Ma and Guilin Qi for their valuable suggestions for improvements.

References

[1] O. Arieli, A. Avron, The value of the four values, Artificial Intelligence 102 (1998) 97–141.
[2] M. Arenas, L. Bertossi, M. Kifer, Applications of annotated predicate calculus to querying inconsistent databases, in: Computational Logic (CL'2000), in: LNCS, vol. 1861, Springer, 2000, pp. 926–941.
[3] L. Bertossi, J. Chomicki, Query answering in inconsistent databases, in: G. Saake, J. Chomicki, R. van der Meyden (Eds.), Logics for Emerging Applications of Databases, Springer, 2003.
[4] S. Benferhat, D. Dubois, S. Kaci, H. Prade, Encoding information fusion in possibilistic logic: A general framework for rational syntactic merging, in: Proceedings of the 14th European Conference on Artificial Intelligence (ECAI'2000), IOS Press, 2000, pp. 3–7.
[5] N. Belnap, A useful four-valued logic, in: G. Epstein (Ed.), Modern Uses of Multiple-Valued Logic, Reidel, 1977, pp. 8–37.
[6] L. Bertossi, A. Hunter, T. Schaub, Introduction to inconsistency tolerance, in: Inconsistency Tolerance, in: LNCS, vol. 3300, Springer, 2004, pp. 1–16.
[7] C. Baral, S. Kraus, J. Minker, V.S. Subrahmanian, Combining knowledgebases of first-order theories, Computational Intelligence 8 (1992) 45–71.
[8] B. Barragáns-Martínez, J. Pazos-Arias, A. Fernández-Vilas, On measuring levels of inconsistency in multi-perspective requirements specifications, in: Proceedings of the 1st Conference on the Principles of Software Engineering (PRISE'04), 2004, pp. 21–30.
[9] W. Carnielli, J. Marcos, A taxonomy of C systems, in: Paraconsistency: The Logical Way to the Inconsistent, Marcel Dekker, 2002, pp. 1–94.
[10] Q. Chen, C. Zhang, S. Zhang, A verification model for electronic transaction protocols, in: Advanced Web Technologies and Applications, in: LNCS, vol. 3007, Springer, 2004, pp. 824–833.
[11] N.C. da Costa, On the theory of inconsistent formal systems, Notre Dame Journal of Formal Logic 15 (1974) 497–510.
[12] S. de Amo, W. Carnielli, J. Marcos, A logical framework for integrating inconsistent information in multiple databases, in: Proceedings of the Second International Symposium on Foundations of Information and Knowledge Systems, in: LNCS, vol. 2284, Springer, 2002, pp. 67–84.
[13] A. Finkelstein, D. Gabbay, A. Hunter, J. Kramer, B. Nuseibeh, Inconsistency handling in multi-perspective specifications, IEEE Transactions on Software Engineering 20 (8) (1994) 569–578.
[14] J. Grant, A. Hunter, Measuring inconsistency in knowledgebases, Journal of Intelligent Information Systems 27 (2006) 159–184.
[15] J. Grant, Classifications for inconsistent theories, Notre Dame Journal of Formal Logic 19 (1978) 435–444.
[16] J. Grant, V.S. Subrahmanian, Applications of paraconsistency in data and knowledge bases, Synthese 125 (2000) 121–132.

[17] A. Hunter, S. Konieczny, Approaches to measuring inconsistent information, in: Inconsistency Tolerance, in: LNCS, vol. 3300, Springer, 2004, pp. 189–234.
[18] A. Hunter, S. Konieczny, Shapley inconsistency values, in: Proceedings of the 10th International Conference on Knowledge Representation (KR'06), AAAI Press, 2006, pp. 249–259.
[19] A. Hunter, B. Nuseibeh, Managing inconsistent specifications: Reasoning, analysis and action, ACM Transactions on Software Engineering and Methodology 7 (1998) 335–367.
[20] A. Hunter, Measuring inconsistency in knowledge via quasi-classical models, in: Proceedings of the National Conference on Artificial Intelligence (AAAI'02), MIT Press, 2002, pp. 68–73.
[21] A. Hunter, Evaluating significance of inconsistencies, in: Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI'03), 2003, pp. 468–473.
[22] A. Hunter, Logical comparison of inconsistent perspectives using scoring functions, Knowledge and Information Systems Journal 6 (2004) 528–543.
[23] A. Hunter, How to act on inconsistent news: Ignore, resolve, or reject, Data and Knowledge Engineering 57 (2006) 221–239.
[24] S. Konieczny, J. Lang, P. Marquis, Quantifying information and contradiction in propositional logic through epistemic actions, in: Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI'03), 2003, pp. 106–111.
[25] K. Knight, Measuring inconsistency, Journal of Philosophical Logic 31 (2001) 77–98.
[26] K. Knight, Two information measures for inconsistent sets, Journal of Logic, Language and Information 12 (2003) 227–248.
[27] S. Konieczny, R. Pino Perez, On the logic of merging, in: Proceedings of the Sixth International Conference on Principles of Knowledge Representation and Reasoning (KR'98), Morgan Kaufmann, 1998, pp. 488–498.
[28] J. Lin, A.O. Mendelzon, Merging databases under constraints, International Journal of Cooperative Information Systems 7 (1) (1998) 55–76.
[29] E. Lozinskii, Information and evidence in logic systems, Journal of Experimental and Theoretical Artificial Intelligence 6 (1994) 163–193.
[30] K. Mu, Z. Jin, R. Lu, W. Liu, Measuring inconsistency in requirements engineering, in: Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU'05), in: LNAI, vol. 3571, Springer, 2005, pp. 440–451.
[31] M. Martinez, A. Pugliese, G. Simari, V.S. Subrahmanian, H. Prade, How dirty is your relational database? an axiomatic approach, in: Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU'07), in: LNCS, vol. 4724, Springer, 2007, pp. 103–114.
[32] Y. Ma, G. Qi, P. Hitzler, Z. Lin, Measuring inconsistency for description logics based on paraconsistent semantics, in: Symbolic and Quantitative Approaches to Reasoning and Uncertainty (ECSQARU'07), in: LNCS, vol. 4724, Springer, 2007, pp. 30–41.
[33] G. Priest, Reasoning about truth, Artificial Intelligence 39 (1989) 231–244.
[34] G. Qi, W. Liu, D. Bell, Measuring conflict and agreement between two prioritized belief bases, in: Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI'05), 2005, pp. 552–557.
[35] R. Reiter, Equality and domain closure in first-order databases, Journal of the Association of Computing Machinery 27 (2) (1980) 235–249.
[36] D. Sarenac, R. Jennings, Implication, preservation, and truth, in: Proceedings of the International Conference on Artificial Intelligence, CSREA Press, 2001.
[37] P. Wong, Ph. Besnard, Paraconsistent reasoning as an analytic tool, Journal of the Interest Group in Propositional Logic 9 (2001) 233–246.