Measuring inconsistency in knowledgebases

John Grant
Department of Computer and Information Sciences
and Department of Mathematics
Towson University
Towson, MD 21252, USA
(jgrant@towson.edu)
and
Anthony Hunter
Department of Computer Science
University College London
Gower Street, London WC1E 6BT, UK
(a.hunter@cs.ucl.ac.uk)

March 16, 2005

Abstract

It is well-known that knowledgebases may contain inconsistencies. We provide a measure to quantify the inconsistency of a knowledgebase, thereby allowing for the comparison of the inconsistency of various knowledgebases, represented as first-order logic formulas.. We use quasi-classical (QC) logic for this purpose. QC logic is a formalism for reasoning and analysing inconsistent information. It has been used as the basis of a framework for measuring inconsistency in propositional theories. Here we extend this framework, by using a first-order logic version of QC logic for measuring inconsistency in first-order theories. We motivate the QC logic approach by considering some formulae as database or knowledgebase integrity constraints. We then define a measure of extrinsic inconsistency that can be used to compare the inconsistency of different knowledgebases. This measure takes into account both the language used and the underlying domain. We show why this definition also captures the intrinsic inconsistency of a knowledgebase. We also provide a formalization of paraconsistent equality, called quasi-equality, and we use this in an extended example of an application for measuring inconsistency between heterogeneous sources of information and integrity constraints prior to merging.

1 Introduction

The need for handling inconsistencies in knowledgebases has been well recognized 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. Various approaches to the applications of paraconsistency to knowledgebases were surveyed in [GS00]. Another method, not discussed there, is given in [BC03] for querying inconsistent knowledgebases. Whilst these methods provide 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 formulas. 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 one that is least inconsistent. We use for our paraconsistent logic quasi-classical (QC) logic because it is well-suited to deal with typical knowledgebase statements, particularly database integrity constraints.

Paraconsistent reasoning is important in handling inconsistent information, and there have been a number of proposals for paraconsistent logics (for reviews see [Hun98, CM02, Pri02]). However, developing non-trivializable, or paraconsistent logics, necessitates some compromise, or weakening, of classical logic. Key paraconsistent logics such as C_ω [dC74] achieve this by weakening the classical connectives, particularly negation. Unfortunately, this results in useful proof rules such as disjunctive syllogism failing, and intuitive equivalences such as $\neg\alpha \vee \beta \equiv \alpha \rightarrow \beta$ not holding.

On the other hand, QC logic restricts the proof theory [BH95, Hun00a]. In this restriction, compositional proof rules (for example, disjunction introduction) cannot be followed by decompositional proof rules (for example, resolution). Whilst this gives a logic that is weaker than classical logic, it does mean that, in a strong sense, the connectives behave classically at the object level. We believe the logic is appealing for reasoning with inconsistencies arising in applications such as systems development [HN98, MDB02], and for reasoning with structured text [Hun00b].

QC logic is particularly well suited to reasoning with integrity constraints that have been violated by data. Suppose each integrity constraint is represented by a disjunction of literals and suppose data is represented by literals. For an inconsistent set of integrity constraints and data, we need to be able to reason with the information using disjunctive syllogism. So for example, we would want to derive β from set 1 of the form $\{\neg\alpha, \alpha \vee \beta\}$ and from set 2 of the form $\{\neg\alpha, \alpha \vee \beta, \neg\beta\}$. Unfortunately with most paraconsistent logics this is not possible. Some such as LPm [Pri89], and versions of Belnap's four-valued logic restricted to minimal models [AA98], will allow the derivation of β from set 1 but not set 2. In contrast, QC logic will support the derivation of β from both set 1 and set 2. More generally, for any set of formulae Δ, if there is a clause $\alpha_1 \vee .. \vee \alpha_n \in \Delta$, where $\alpha_1, ..., \alpha_n$ are literals, and for $\alpha_i \in \{\alpha_1, .., \alpha_n\}$, there is a literal $\neg\alpha_i \in \Delta$, then by QC logic, the resolvent of $\alpha_1 \vee .. \vee \alpha_n$ and $\neg\alpha_i$ (a clause formed from $\alpha_1 \vee .. \vee \alpha_n$ by deleting α_i) is an inference from Δ.

QC logic has also been used for comparing heterogeneous sources where inconsistencies often arise between the sources. Suppose we are dealing with a group of clinicians advising on some patient, a group of witnesses of some incident, or a set of newspaper reports covering some event. These are all situations where we expect some degree of inconsistency in the information. Suppose that the information by each source is represented by a set of formulae. We may then want to ask questions such as whether one source is more inconsistent than another and by what degree. A novel approach to formalizing this problem is to use propositional QC logic to measure inconsistency [Hun02, Hun03]. In this, each inconsistent set of formulae is reflected in the quasi-classical models for the set, and then the inconsistency is measured in the models. Obviously, this is not possible in classical logic, or indeed many non-classical logics, because there is no model of an inconsistent set of formulae.

A key feature of the QC semantics is that there is a model for any formula, and for any set of formulae. There is also a natural deduction proof theory for propositional QC logic [Hun00a]. Entailment for QC logic for propositional CNF formulae is coNP-complete, and via a linear time transformation these formulae can be handled using classical logic theorem provers [MP01]. QC logic has been extended to a first-order language with the semantics based on a form of Herbrand interpretation [Hun01].

In this paper, we begin the study of measuring the inconsistency of knowledgebases written as first-order formulae by using QC logic. We show that the concept of QC interpretation is natural if we think of the disjunctive formulae as integrity constraints. We develop a measure for the inconsistency of a first-order theory and give examples to illustrate its usefulness. In particular, we show how this measure allows

for the comparison of inconsistent theories. This paper therefore extends our QC logic-based approach to measuring inconsistencies in propositional theories [Hun02, Hun03] to the first-order case, by incorporating the notion of "degrees of inconsistency" based on analysing both the first-order language and the domain for the interpretations [Gra78]. This paper also extends QC logic with a form of equality, called quasi-equality, appropriate for paraconsistent reasoning. We show the utility of quasi-equality in reasoning with inconsistency in applications in reasoning with integrity constraints.

2 First-order QC logic

First-order QC logic contains the standard logical symbols: a set of variable symbols $\mathcal{V}$, the connectives $\{\neg, \vee, \wedge, \rightarrow\}$, the quantifiers $\{\forall, \exists\}$, as well as punctuation symbols as needed (parentheses and commas). A specific language $\mathcal{L}$ is determined by its set of predicate and function symbols. In this paper our focus is on finite models, so we assume that all functions are 0-ary, that is, constants. In the examples, we present specific languages by a tuple $\langle \mathcal{P}, \mathcal{C} \rangle$ where $\mathcal{P}$ is a set of predicate symbols and $\mathcal{C}$ is a set of constant symbols. We write the predicate symbols with their arities in parentheses. We use uppercase letters like P and R for predicate symbols, lowercase letters like a, b, c, and d, perhaps with subscript, for constant symbols, and x and y, perhaps with subscript, for variable symbols. A term is either a constant or a variable. We assume the usual classical definitions for the 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 $t_1, .., t_n$ 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 $\mathsf{Formulae}(\mathcal{L})$.

Definition 1. *Let α be an atom and let $\sim$ be a complementation operation such that $\sim\alpha$ is $\neg\alpha$ and $\sim(\neg\alpha)$ is α. The $\sim$ operator is not part of the object language but it makes some definitions clearer.*

In QC logic the focus of a clause, as defined below, plays an important role.

Definition 2. *Let $\alpha_1 \vee \ldots \vee \alpha_n$ be a ground clause where $n > 1$ and let α_i be a ground literal. The* **focus** *of $\alpha_1 \vee \ldots \vee \alpha_n$ by α_i, denoted $\otimes(\alpha_1 \vee \ldots \vee \alpha_n, \alpha_i)$ is defined as the clause obtained by removing α_i from $\alpha_1 \vee \ldots \vee \alpha_n$.*

Example 1. *Let $\alpha \vee \beta \vee \gamma$ be a ground clause. Then, $\otimes(\alpha \vee \beta \vee \gamma, \beta) = \alpha \vee \gamma$.*

The notion of a structure in first-order QC logic is based on a standard notion of classical interpretation.

Definition 3. *A* **classical structure** *for the language $\mathcal{L}$ is a pair (D, I) where D is a non-empty set called the* **domain** *and I is a function called the* **interpretation** *that assigns predicates over D to the symbols in $\mathcal{L}$ as follows:*

1. *For every constant symbol c, $I(c) \in D$.*
2. *For every predicate symbol $P(n)$ (of arity $n > 0$), $I(P) : D^n \mapsto \{0, 1\}$ is an n-ary predicate.*

To make the notation easier to follow, we will assume that I maps the constant symbols to themselves; hence D will always contain all the constant symbols of $\mathcal{L}$. This also means that all constant symbols are mapped to distinct elements. Hence for a given language $\mathcal{L} =< \mathcal{P}, \mathcal{C} >$ and domain D we must have $\mathcal{C} \subseteq D$. In order to express statements involving the elements of D not in $\mathcal{C}$, we expand the language $\mathcal{L}$ by adding constant symbols for all elements of $D - \mathcal{C}$. We write $\mathcal{L}(D)$ for this language; in other words, $\mathcal{L}(D) = \langle \mathcal{P}, D \rangle$. We handle variables in first-order formulae using the standard notion of an assignment.

Definition 4. *Let (D, I) be a classical structure. An* **assignment** *is a function $A : \mathcal{V} \mapsto D$. Given an assignment A, an* **x-variant assignment** *A' is the same as A except perhaps in the assignment for the variable x. An assignment A is extended to terms t as follows:*

$$A(t) = d \text{ if } t = x \text{ and } A(x) = d$$
$$A(t) = d \text{ if } t = d$$

Definition 5. *Let ϕ be a formula with one or more occurrences of a free variable x (i.e. x is not in the scope of a quantifier) and let d be a constant. The* **substitution** *of x by d in ϕ is obtained by replacing all occurences of the free variable by d, and the result of this substitution is denoted $\phi[x/d]$.*

We use a pair of classical interpretations to give a QC interpretation.

Definition 6. *A* **bistructure** *is a tuple (D, I^+, I^-) where (D, I^+) and (D, I^-) are classical interpretations.*

Using our conventions we must have for all constant symbols c, $I^+(c) = I^-(c)$. In the presence of function symbols we would have $I^+(f) = I^-(f)$. So the classical interpretations I^+ and I^- in a bistructure can only differ in their assignment to predicate symbols. In effect, I^+ is the interpretation for positive literals and I^- is the interpretation for negative literals. This is formalized in the definition for decoupled satisfaction.

Definition 7. *For a bistructure $E = (D, I^+, I^-)$ and an assignment A, we define a satisfiability relation, $\models_d$, called* **decoupled satisfaction** *for literals in $\mathcal{L}(D)$ as follows:*

$$(E, A) \models_d P(t_1, .., t_n) \quad \text{iff } I^+(P)(A(t_1), .., A(t_n)) = 1$$

$$(E, A) \models_d \neg P(t_1, .., t_n) \quad \text{iff } I^-(P)(A(t_1), .., A(t_n)) = 1$$

Since we allow both an atom and its complement to be satisfiable, we have decoupled, at the level of the structure, the link between a formula 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. This intuition coincides with that of four-valued logics [Bel77]. However, we will not follow the four-valued lattice-theoretic interpretation of connectives given in [Bel77], but instead provide a significantly different semantics next.

In the definition of entailment for QC logic two additional satisfaction relations, called strong satisfaction and weak satisfaction, are required. However, in this paper we will only deal with strong satisfaction. We will also assume that all formulae are in prenex conjunctive normal form (PCNF), that is, all the quantifiers are in front and they (if any) are followed by a conjunction of clauses or a single clause[1]. It is known that every formula is logically equivalent (using classical entailment) to one in PCNF (see for example [Smu68]). There is a standard way in which the quantifiers can be moved to the front. Putting the rest of the formula into conjunctive normal form involves using the standard commutative, associative, and distributive laws; the rewriting of $\rightarrow$ by $\neg$ and $\vee$; however we do not allow the use of any rules involving tautologies and contradictions. Our definition of strong satisfaction does not depend on which equivalent PCNF version of a formula is used, assuming the proviso above.

Definition 8. *A satisfiability relation, $\models_s$, called* **strong satisfaction** *is defined by induction on the length*

[1]The restriction to PCNF is not necessary for QC logic, but it makes the presentation simpler.

of a formula as follows:

$(E, A) \models_s \alpha$ *iff* $(E, A) \models_d \alpha$

$(E, A) \models_s \alpha_1 \vee \ldots \vee \alpha_n$
iff $[(E, A) \models_s \alpha_1$ *or ... or* $(E, A) \models_s \alpha_n]$
and $\forall i$ *s.t.* $1 \leq i \leq n$ $[(E, A) \models_s \sim\alpha_i$ *implies* $(E, A) \models_s \otimes(\alpha_1 \vee \ldots \vee \alpha_n, \alpha_i)]$

$(E, A) \models_s \phi_1 \wedge \ldots \wedge \phi_m$ *iff* $[(E, A) \models_s \phi_1$ *and* $\ldots$ *and* $(E, A) \models_s \phi_m]$

$(E, A) \models_s \exists x\ \theta$ *iff for some x-variant assignment* A', $(E, A') \models_s \theta[x/d]$ *where* $A'(x) = d$

$(E, A) \models_s \forall x\ \theta$ *iff for all x-variant assignments* A', $(E, A') \models_s \theta[x/d]$ *where* $A'(x) = d$

Definition 9. *We extend strong satisfaction to a bistructure as follows:*

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

Next we define the concept of a QC model. In this paper we will only deal with finite sets of sentences.

Definition 10. *Let Δ be a finite set of formulae and let E be a bistructure. E is a* **QC model** *of Δ iff for all $\theta \in \Delta, E \models_s \theta$.*

We also require the following subsidiary definition to help analyse QC models.

Definition 11. *For a language $\mathcal{L} = \langle \mathcal{P}, \mathcal{C} \rangle$ and a domain D,*

$$\mathsf{Groundatoms}(\mathcal{L}, D) = \{P(d_1, .., d_n) \mid P(n) \in \mathcal{P} \text{ and } d_1, .., d_n \in D\}$$
$$\mathsf{Groundliterals}(\mathcal{L}, D) = \mathsf{Groundatoms}(\mathcal{L}, D) \cup \{\neg\alpha \mid \alpha \in \mathsf{Groundatoms}(\mathcal{L}, D)\}$$

Since we restrict each language $\mathcal{L}$ in this paper to 0-ary functions, we can equivalently represent any QC model $E = (D, I^+, I^-)$ by a set of ground literals as follows.

$$\{\alpha \mid E \models_s \alpha \text{ and } \alpha \in \mathsf{Groundliterals}(\mathcal{L}, D)\}$$

We write $\mathsf{QC}(\mathcal{L}, \Delta, D)$ for the class of QC models of Δ where the formulae in Δ are formulae in $\mathcal{L}$, and the domain for the QC models is D. We will use M to denote a QC model in the form of a set of ground literals. It is possible to adopt this representation of models because earlier we assumed that for any language $\mathcal{L} =< \mathcal{P}, \mathcal{C} >$ and domain D we have $\mathcal{C} \subseteq D$. In the rest of this paper, we will assume that all languages $\mathcal{L} =< \mathcal{P}, \mathcal{C} >$ are finite (i.e. $\mathcal{P}$ is finite and $\mathcal{C}$ is finite) and that all domains D are finite. This means that all QC models we consider in this paper are finite.

Definition 12. *Let $\mathcal{L}$ be a language. For a set of QC models $\{M_1, ..., M_n\}$, where for all $M_i \in \{M_1, ..., M_n\}$, $M_i \subseteq \mathsf{Groundliterals}(\mathcal{L}, D)$, the set of* **satisfied formulae** *is given by the set $\mathsf{SF}(\{M_1, ..., M_n\})$ as follows.*

$$\mathsf{SF}(\{M_1, ..., M_n\}) = \{\alpha \in \mathsf{Formulae}(\mathcal{L}) \mid \text{ for all } M_i \in \{M_1, ..., M_n\}\ M_i \models_s \alpha\}$$

Clearly, we have $\Delta \subseteq \mathsf{SF}(\mathsf{QC}(\mathcal{L}, \Delta, D))$. Later, we use the notion of satisfied formulae to restrict our attention to the most useful QC models for analysing inconsistency.

3 QC Models

In the previous section we defined the concept of a QC model. Here we give some examples to show the desirability of using QC models for analysing inconsistencies in knowledgebases. For this we take both the language and the domain into account.

As different languages contain different sets of formulae, it will be useful to have $\mathcal{L}$ as a parameter when we discuss concepts about models. Also we will assume that Δ is a set of formulae of $\mathcal{L}$. We first note that every set of formulae Δ has a QC model, $\mathsf{UHM}(\Delta)$, the universal Herbrand model of Δ. To construct this, we start with the Herbrand universe for Δ, which is the set of all ground terms in the language of Δ (if there are no constants in Δ, add an arbitrary constant symbol), and then $\mathsf{UHM}(\Delta)$ is obtained by allowing all literals in the Herbrand universe, $P(a_1, \ldots, a_n)$ and $\neg P(a_1, \ldots, a_n)$ to hold. So $\mathsf{UHM}(\Delta)$ can be represented by the set of all ground literals in the language $\mathcal{L}$.

We note that in a sense $\mathsf{UHM}(\Delta)$ is a "big" QC model for Δ and not probably the one we had in mind. In measuring inconsistency we will usually need to deal with smaller QC models. But this depends on the domain. In knowledgebases we usually have some domain in mind; such as certain character strings or numbers. Consider a family database that might have in its domain names of people and numbers for year of birth; or an employee database that might have names, addresses, salaries and other types of data. To deal with the issue of what values are allowed where in predicates, we would need to use a sorted logic to be precise. However, in order to make the presentation less complicated we do not use a sorted logic, but assume that if sorts are needed, they can be introduced as unary predicates. In any case, we usually have various domains in mind, such as all possible names or year values between 1600 and 2004 for the family database. Thus the domain is important and we use it as a parameter. Under different circumstances, different domains may be appropriate. We also assume for this paper that all the domains are finite. This is reasonable because even for integer and real values only finitely many can be represented or would be used in any case.

As mentioned above, $\mathsf{UHM}(\Delta)$ is a big model in some sense because it may have many more literals than are needed to satisfy Δ. On the other hand, $\mathsf{UHM}(\Delta)$ may be a small QC model if the Herbrand universe is a small subset of the intended domain. For example, we may have only 50 different year values present in a particular family database for year of birth, but would like the domain to include all possible such years. For this reason we include in our definitions now the domain. Furthermore, in measuring the inconsistency of a theory, it will be helpful to consider models that are minimal in some sense.

Definition 13. *Let $\mathcal{L}$ be a language and Δ a set of formulae in $\mathcal{L}$. The set of minimal QC models of Δ with domain D is given by $\mathsf{MQC}(\mathcal{L}, \Delta, D)$ as follows.*

$$\mathsf{MQC}(\mathcal{L}, \Delta, D) = \{M \in \mathsf{QC}(\mathcal{L}, \Delta, D) \mid \textit{ if } M' \subset M, \textit{ then } M' \notin \mathsf{QC}(\mathcal{L}, \Delta, D)\}$$

The minimal QC models are just the models without irrelevant, useless information. They are analogous to the role of k-minimal models in Belnap's four-valued logic [AA98]. Restricting consideration to minimal QC models does not affect the reasoning, as illustrated by the following result.

Theorem 1. *Let $\mathcal{L}$ be a language, Δ a set of formulae in $\mathcal{L}$ and D a domain.*

$$\mathsf{SF}(\mathsf{QC}(\mathcal{L}, \Delta, D)) = \mathsf{SF}(\mathsf{MQC}(\mathcal{L}, \Delta, D))$$

Proof. First, we show that $\mathsf{SF}(\mathsf{MQC}(\mathcal{L}, \Delta, D)) \subseteq \mathsf{SF}(\mathsf{QC}(\mathcal{L}, \Delta, D))$. To do this, we require the following definition for "acceptable". For $N \subseteq \mathsf{Groundliterals}(\mathcal{L}, D)$, N is acceptable for Δ if for all $\theta \in \Delta$, N is acceptable for θ. We define acceptable for any formula θ as follows:

N is acceptable for α if α is a literal
N is acceptable for $\phi_1 \wedge .. \wedge \phi_n$ if N is acceptable for ϕ_1 and $\ldots$ and N is acceptable for ϕ_n
N is acceptable for $\alpha_1 \vee .. \vee \alpha_n$ if (for all i $(1 \leq i \leq n)$ $N \models_s \sim \alpha_i$ implies $N \models_s \otimes(\alpha_1 \vee \ldots \vee \alpha_n, \alpha_i)$)
N is acceptable for $\exists x\theta$ if N is acceptable for $\theta[x/d]$ for some $d \in D$
N is acceptable for $\forall x\theta$ if N is acceptable for $\theta[x/d]$ for all $d \in D$

So if $M \models_s \theta$, and N is acceptable for θ, then $M \cup N \models_s \theta$. Furthermore, for all $M' \in \mathsf{SF}(\mathsf{QC}(\mathcal{L}, \Delta, D))$, for all $M \in \mathsf{SF}(\mathsf{MQC}(\mathcal{L}, \Delta, D))$, if $M \subseteq M'$, then there is an $N \subseteq M'$ s.t. $M \cup N = M'$ and N is acceptable for Δ. So, $\mathsf{SF}(\mathsf{MQC}(\mathcal{L}, \Delta, D)) \subseteq \mathsf{SF}(\mathsf{QC}(\mathcal{L}, \Delta, D))$. Also $\mathsf{MQC}(\mathcal{L}, \Delta, D) \subseteq \mathsf{QC}(\mathcal{L}, \Delta, D)$, so $\mathsf{SF}(\mathsf{QC}(\mathcal{L}, \Delta, D)) \subseteq \mathsf{SF}(\mathsf{MQC}(\mathcal{L}, \Delta, D))$. Hence, $\mathsf{SF}(\mathsf{QC}(\mathcal{L}, \Delta, D)) = \mathsf{SF}(\mathsf{MQC}(\mathcal{L}, \Delta, D))$. □

Next we show the advantage of using minimal QC models for analysing inconsistent information.

Example 2. *Let $\mathcal{L} = \langle\{P(1)\},\{a,b\}\rangle$, and $\Delta = \{\neg P(a) \vee P(b), P(a)\}$. In this case there is only one minimal QC model for any D, $M = \{P(a), P(b)\}$ but there is an additional minimal 4-valued model, $M' = \{P(a), \neg P(a)\}$. M' is not a QC model of Δ because in a QC model where $P(a)$ holds, $\otimes(\neg P(a) \vee P(b), \neg P(a))$, which is $P(b)$ must also hold. One way to interpret this is to include implication ($\rightarrow$) in the explanation, that is, that the formula $\neg P(a) \vee P(b)$ represents $P(a) \rightarrow P(b)$ (as well as $\neg P(b) \rightarrow \neg P(a)$). The concept of QC model makes sure that if $P(a)$ and $P(a) \rightarrow P(b)$ both hold, then $P(b)$ must also hold.*

In the next example we see why the QC logic approach fits in well with the concept of integrity constraints in databases.

Example 3. *Let $\mathcal{L} = \langle\{P(1), Q(1)\},\{a,b,c\}\rangle$, $D = \{a,b,c\}$, and*

$$\Delta = \{\forall x(\neg P(x) \vee Q(x)), P(a), P(c), \neg P(c)\}$$

In this case there are two minimal QC models for Δ, namely M_1 and M_2 below.

$$M_1 = \{P(a), Q(a), \neg P(b), P(c), \neg P(c), Q(c)\}$$
$$M_2 = \{P(a), Q(a), Q(b), P(c), \neg P(c), Q(c)\}$$

Note that M_1' and M_2' below are not (minimal) QC models.

$$M_1' = \{P(a), \neg P(a), \neg P(b), P(c), \neg P(c), Q(c)\}$$
$$M_2' = \{P(a), \neg P(a), Q(b), P(c), \neg P(c), Q(c)\}$$

Not having M_1' and M_2' as QC models for Δ is quite reasonable when we think of the first sentence in Δ as "All P's are Q's." In that sense, if $P(a)$ is in the model, $Q(a)$ should be there also. Hence M_1' and M_2' should not be QC models.

Next we give two examples with typical database integrity constraints: first a referential integrity constraint, and then a multivalued dependency. In Section 6, we also give an example with a functional dependency.

Example 4. *Let $\mathcal{L} = \langle\{P(2), Q(2)\},\{a,b,c\}\rangle$, $D = \{a,b,c,d\}$, and*

$$\Delta = \{\forall x \forall y \exists z(\neg P(x,y) \vee Q(z,y)), P(a,b), P(b,b), Q(a,c), \neg Q(a,c)\}$$

The quantified statement represents the referential constraint (inclusion dependency) that every value of the second attribute of P must appear as the second attribute of Q. There are numerous minimal QC models for Δ. However, they all contain $P(a,b)$,$P(b,b)$,$Q(a,c)$, and $\neg Q(a,c)$. As a result of the inclusion dependency, they also all contain exactly one of $Q(a,b)$,$Q(b,b)$,$Q(c,b)$ or $Q(d,b)$. An example of one of these minimal QC models is M_1 as follows:

$$M_1 = \{P(a,b), P(b,b), Q(a,c), \neg Q(a,c), Q(a,b), Q(d,a), Q(b,c), Q(c,d)\}$$

Example 5. *Let $\mathcal{L} = \langle\{P(3)\},\{a,b,c,d,e\}\rangle$, $D = \{a,b,c,d,e\}$, and*

$$\Delta = \{\forall x \forall y \forall z \forall u \forall v(\neg P(x,y,z) \vee \neg P(x,u,v) \vee P(x,y,v)), P(a,b,c), \neg P(a,b,c), P(a,d,e)\}$$

The universal statement represents the multivalued dependency of the second and third attributes of P on the first attribute. So each minimal QC model of Δ has the following as a subset.

$$\{P(a,b,c), \neg P(a,b,c), P(a,d,e), P(a,d,c), P(a,b,e), \neg P(a,a,c), \neg P(a,c,c), \neg P(a,d,c), \neg P(a,e,c)\}$$

The multivalued dependency forces $P(a,d,c)$ and $P(a,b,e)$ into each model. It also has the effect, because of the 4-valued nature of our structures, of adding several negated atoms on account of $\neg P(a,b,c)$ being in Δ.

Whilst four-valued logic [Bel77] also directly models inconsistent sets of formulae, QC logic has the extra constraint in the semantics for disjunction that ensures that if the complement of a disjunct holds in the model, then the resolvent should also hold in the model (in effect disjunctive syllogism) irrespective of other conflicting information holding. This means that the QC models are better at describing the inconsistencies arising between data and clauses (such as integrity constraints). The shortcomings of Belnap's four-valued logic also apply to minimal four-valued logics by [AA98], and to three-valued logics such as 3-interpretations by [Lev84], and a similar proposal by [Gra78], and minimal LPm [Pri89].

4 The inconsistency measure

In this section we propose a measure for the inconsistency of a theory based on the concept of degree of inconsistency in [Gra78], but using QC logic and some ideas from [Hun02]. We take into account the fact that a theory may have many different minimal QC models depending on the language and domain. We start by defining an inconsistency measure for a QC model, based on the concept of coherence from [Hun02]. This measure is a ratio between 0 and 1 whose denominator is the total possible number of inconsistencies in the bistructure.

Definition 14. *For a QC model M,*

$$\mathsf{Conflictbase}(M) = \{\alpha \mid \alpha \in M \textit{ and } \neg\alpha \in M\}$$

Thus the size of the conflictbase of a QC model is the number of inconsistencies in the QC model.

Recall in Section 2, we assume all domains D are finite and all models M are finite.

Definition 15. *The measure of inconsistency for a model M in the context of a language $\mathcal{L}$ and a domain D (i.e.* $\mathsf{Conflictbase}(M) \subseteq \mathsf{Groundatoms}(\mathcal{L}, D)$ *holds) is given by the* ModelInc *function giving a value in $[0, 1]$ as follows.*

$$\mathsf{ModelInc}(M, \mathcal{L}, D) = \frac{|\mathsf{Conflictbase}(M)|}{|\mathsf{Groundatoms}(\mathcal{L}, D)|}$$

Example 6. $\mathcal{L} = \langle\{P(2), R(1)\}, \{\}\rangle$, $D = \{a, b, c\}$, $M = \{P(a, a), \neg P(a, a), R(a), \neg R(b), P(b, c)\}$. *Here,* $|\mathsf{Groundatoms}(\mathcal{L}, D)| = 12$ *(9 ground atoms for P and 3 for R).* $\mathsf{Conflictbase}(M) = \{P(a, a)\}$. *Hence,* $\mathsf{ModelInc}(M, \mathcal{L}, D) = \frac{1}{12}$.

The ModelInc function is antimonotonic as follows.

- If $\mathcal{L}_1 \subseteq \mathcal{L}_2$, then $\mathsf{ModelInc}(M, \mathcal{L}_2, D) \leq \mathsf{ModelInc}(M, \mathcal{L}_1, D)$;
- If $D_1 \subseteq D_2$, then $\mathsf{ModelInc}(M, \mathcal{L}, D_2) \leq \mathsf{ModelInc}(M, \mathcal{L}, D_1)$.

For $\mathcal{L}$,D, if Δ contains only ground formulae (i.e. no formulae in Δ contain variables), and $M_1 \in \mathsf{MQC}(\mathcal{L}, \Delta, D)$, and $M_2 \in \mathsf{MQC}(\mathcal{L}, \Delta, D)$, then $\mathsf{ModelInc}(M_1, \mathcal{L}, D) = \mathsf{ModelInc}(M_2, \mathcal{L}, D)$. However, in general, for $\mathcal{L}, \Delta$ and D, if $M_1 \in \mathsf{MQC}(\mathcal{L}, \Delta, D)$ and $M_2 \in \mathsf{MQC}(\mathcal{L}, \Delta, D)$, then it is not necessarily the case that $\mathsf{ModelInc}(M_1, \mathcal{L}, D) = \mathsf{ModelInc}(M_2, \mathcal{L}, D)$ holds. In fact, $\mathsf{MQC}(\mathcal{L}, \Delta, D)$ may contain both consistent ($\mathsf{ModelInc}(M) = 0$) and inconsistent models as we show in the next example.

Example 7. *Let* $\mathcal{L} = \langle\{P(1), Q(1)\}\{a\}\rangle$, $\Delta = \{\exists x\ P(x), \neg P(a) \vee Q(a), \neg Q(a)\}$ *and* $D = \{a, b\}$. *So* $\mathsf{MQC}(\mathcal{L}, \Delta, D) = \{M_1, M_2\}$ *where*

$$M_1 = \{P(a), \neg P(a), Q(a), \neg Q(a)\}$$
$$M_2 = \{P(b), \neg P(a), \neg Q(a)\}$$

Hence, $\mathsf{ModelInc}(M_1, \mathcal{L}, D) > \mathsf{ModelInc}(M_2, \mathcal{L}, D)$ *holds.*

To address the fact that not all minimal QC models for a set of formulae have the same size conflictbase, and hence do not all have the same assignment by the ModelInc function, we restrict the class of minimal QC models to the class of preferred QC models by taking only the minimal QC models with a minimal conflictbase. Essentially, we are restricting consideration to the least "inconsistent" models. This is an optimistic view saying that for the set of models for a set of formulae, we restrict consideration to the least conflicting ones. This is analogous to the models that minimize the use of assignment of $Both$ in the minimal four-valued logic [AA98] and to the minimal LP models in LPm [Pri89].

Definition 16. *Let Δ be a set of formulae in a language $\mathcal{L}$ and let D be a domain. The set of preferred QC models of Δ with domain D is given by $\mathsf{PQC}(\mathcal{L}, \Delta, D)$ as follows.*

$$\mathsf{PQC}(\mathcal{L}, \Delta, D) = \{M \in \mathsf{MQC}(\mathcal{L}, \Delta, D) \mid \text{for all } M' \in \mathsf{MQC}(\mathcal{L}, \Delta, D)\ |\mathsf{Conflictbase}(M)| \leq |\mathsf{Conflictbase}(M')|\}$$

Example 8. *Continuing Example 4,* $\mathsf{PQC}(\mathcal{L}, \Delta, D) = \mathsf{MQC}(\mathcal{L}, \Delta, D)$.

Example 9. *Continuing Example 7,* $\mathsf{PQC}(\mathcal{L}, \Delta, D) = \{M_2\}$.

There are non-minimal QC models that have a minimal conflictbase, as we show below.

Example 10. *Let $\mathcal{L} = \langle\{P(1)\}\{a\}\rangle$, $\Delta = \{P(a)\}$, and $D = \{a, b\}$. If $M = \{P(a), P(b)\}$, then $M \in \mathsf{QC}(\mathcal{L}, \Delta, D)$, but $M \notin \mathsf{MQC}(\mathcal{L}, \Delta, D)$, even though $|\mathsf{Conflictbase}(M)| = 0$.*

Recall that our interest in using $\mathsf{MQC}(\mathcal{L}, \Delta, D)$ rather than $\mathsf{QC}(\mathcal{L}, \Delta, D)$, for a set of formulae of Δ, is that the models in $\mathsf{MQC}(\Delta)$ do not contain irrelevant information for analysing inconsistency. Furthermore, we have shown in Theorem 1, that by using minimal QC models, we do not lose any useful information.

Now we show that for a particular class of formulae, all minimal models are preferred.

Theorem 2. *For $\mathcal{L}$, Δ and D, if for all $\phi \in \Delta$, ϕ does not incorporate an existential quantifier, then* $\mathsf{PQC}(\mathcal{L}, \Delta, D) = \mathsf{MQC}(\mathcal{L}, \Delta, D)$.

Proof. Assume Δ does not include any existentially quantified formulae. Let $\mathsf{Grounding}(\Delta, D)$ be the smallest set such that: (1) $\Delta \subseteq \mathsf{Grounding}(\Delta, D)$; (2) if $\forall x\psi \in \mathsf{Grounding}(\Delta, D)$ and $d \in D$, then $\psi[x/d] \in \mathsf{Grounding}(\Delta, D)$; and (3) if $\phi_1 \wedge \ldots \wedge \phi_n \in \mathsf{Grounding}(\Delta, D)$, then $\phi_1, \ldots, \phi_n \in \mathsf{Grounding}(\Delta, D)$, Let $\mathsf{G}(\Delta, D) = \{\theta \in \mathsf{Grounding}(\Delta, D) \mid \theta \text{ is a ground formula }\}$. So $\mathsf{G}(\Delta, D)$ is a set of ground clauses formed in $\mathcal{L}(D)$. Let $\mathsf{I}(\mathsf{G}(\Delta, D)) = \{\Gamma \subseteq \mathsf{G}(\Delta, D) \mid \Gamma \text{ is inconsistent }\}$. Let $\mathsf{MI}(\mathsf{G}(\Delta, D)) = \{\Gamma \in \mathsf{I}(\mathsf{G}(\Delta, D)) \mid \text{ for all } \Gamma' \in \mathsf{I}(\mathsf{G}(\Delta, D)), \text{ if } \Gamma' \subseteq \Gamma, \text{ then } \Gamma' = \Gamma\}$. For all $M \in \mathsf{MQC}(\mathcal{L}, \Delta, D)$, $\alpha_i \in M$ and $\neg\alpha_i \in M$ iff there is a $\alpha_1 \vee .. \vee \alpha_n \in \bigcup \mathsf{MI}(\mathsf{G}(\Delta, D))$ such that $\alpha_i \in \{\alpha_1, .., \alpha_n\}$. Therefore, for $M_1 \in \mathsf{MQC}(\mathcal{L}, \Delta, D)$ and $M_2 \in \mathsf{MQC}(\mathcal{L}, \Delta, D)$, $\alpha_i \in M_1$ and $\neg\alpha_i \in M_2$ iff $\alpha_i \in M_2$ and $\neg\alpha_i \in M_2$. Therefore, $\mathsf{Conflictbase}(M_1) = \mathsf{Conflictbase}(M_2)$. Hence, for all $M, M' \in \mathsf{MQC}(\mathcal{L}, \Delta, D)$, $|\mathsf{Conflictbase}(M)| = |\mathsf{Conflictbase}(M')|$, and so $\mathsf{PQC}(\mathcal{L}, \Delta, D) = \mathsf{MQC}(\mathcal{L}, \Delta, D)$. □

The need for the previous result, in part, stems from the non-standard behaviour of existential quantification. We illustrate this, by extending Example 7, as follows.

Example 11. *Let $\mathcal{L} = \langle\{P(1), Q(1)\}\{a, b\}\rangle$, and $D = \{a, b\}$.*

$$\Delta_1 = \{\exists x\ P(x), \neg P(a) \vee Q(a), \neg Q(a)\}$$
$$\Delta_2 = \{P(a) \vee P(b), \neg P(a) \vee Q(a), \neg Q(a)\}$$

In classical logic, $\exists x\ P(x)$ and $P(a) \vee P(b)$ are, in a sense, interchangeable when $D = \{a, b\}$. But, $\mathsf{MQC}(\mathcal{L}, \Delta_1, D) = \{M_1, M_2\}$, and $\mathsf{MQC}(\mathcal{L}, \Delta_2, D) = \{M_2\}$, where

$$M_1 = \{P(a), \neg P(a), Q(a), \neg Q(a)\}$$
$$M_2 = \{P(b), \neg P(a), \neg Q(a)\}$$

The non-standard behaviour is a result of the difference in the semantics for disjunction between classical logic and QC logic.

Given $\mathcal{L}$ and D, MI measures the inconsistency of a theory in a uniform manner. Assuming $M_1 \in \mathsf{PQC}(\mathcal{L}, \Delta, D)$ and $M_2 \in \mathsf{PQC}(\mathcal{L}, \Delta, D)$, we have $|\mathsf{Conflictbase}(M_1)| = |\mathsf{Conflictbase}(M_2)|$, and therefore $\mathsf{MI}(M_1, \mathcal{L}, D) = \mathsf{MI}(M_2, \mathcal{L}, D)$. This means that although there may be many different preferred QC models of a theory Δ, if we fix $\mathcal{L}$ and D, then all such preferred QC models must have the same MI value.

Example 12. *Let* $\mathcal{L} = \langle \{P(1), Q(1)\}, \{a, b, c, d\} \rangle$ *and* $D = \{a, b, c, d\}$.

$$\Delta_1 = \{P(a) \wedge \neg P(a), Q(a) \wedge (Q(b) \vee Q(c) \vee Q(d))\}$$
$$\Delta_2 = \{P(a) \wedge \neg P(a), Q(a) \vee (Q(b)\}$$

Hence

$$\mathsf{PQC}(\mathcal{L}, \Delta_1, D) = \{\{P(a), \neg P(a), Q(a), Q(b)\}, \{P(a), \neg P(a), Q(a), Q(c)\}, \{P(a), \neg P(a), Q(a), Q(d)\}\}$$
$$\mathsf{PQC}(\mathcal{L}, \Delta_2, D) = \{\{P(a), \neg P(a), Q(a)\}, \{P(a), \neg P(a), Q(b)\}\}$$

Therefore, we get the following evaluations.

$$\textit{for all } M \in \mathsf{PQC}(\mathcal{L}, \Delta_1, D)\ \mathsf{MI}(M, \mathcal{L}, D) = 1/8$$
$$\textit{for all } M \in \mathsf{PQC}(\mathcal{L}, \Delta_2, D)\ \mathsf{MI}(M, \mathcal{L}, D) = 1/8$$

Next, suppose that we take two different domains of the same size for a given $\mathcal{L}$. We show that the MI value remains the same.

Theorem 3. *Given* $\mathcal{L}$ *and* Δ*, let* D_1 *and* D_2 *be any two domains of the same size. If* $M_1 \in \mathsf{PQC}(\mathcal{L}, \Delta, D_1)$ *and* $M_2 \in \mathsf{PQC}(\mathcal{L}, \Delta, D_2)$*, then* $\mathsf{MI}(M_1, \mathcal{L}, D_1) = \mathsf{MI}(M_2, \mathcal{L}, D_2)$.

Proof. Suppose that $M_1 \in \mathsf{PQC}(\mathcal{L}, \Delta, D_1)$. We construct a model $M_3 \in \mathsf{PQC}(\mathcal{L}, \Delta, D_2)$ as follows. Define a bijective function $F : D_1 \to D_2$ such that for all $c \in \mathcal{C}$, $F(c) = c$. Write for each literal $\alpha \in \mathcal{L}$ $F(\alpha)$ for the literal where each $d \in D_1$ is replaced by $F(d)$. Let $M_3 = \{F(\alpha) \mid \alpha \in M_1\}$. Clearly, $M_3 \in \mathsf{QC}(\mathcal{L}, \Delta, D_2)$. M_3 must also be minimal because if a proper subset of M_3 were a QC model, by applying F^{-1} we could obtain a QC model with D_1 for the domain making M_1 not minimal. Similarly we can show that M_3 is preferred. The result now follows. □

Using this theorem we are able to define, for a theory in a language, a sequence of inconsistency ratios, as follows.

Definition 17. *We define the* **extrinsic inconsistency of a theory** Δ *in a language* $\mathcal{L}$*,* $\mathsf{TheoryInc}(\Delta, \mathcal{L})$*, as a sequence* $\langle r_1, r_2, \ldots \rangle$*, where for all* $n \geq 1$ *let* D_n *be a domain for size* n*, and if there is an* $M \in \mathsf{PQC}(\mathcal{L}, \Delta, D_n)$*, then let* $r_n = \mathsf{MI}(M, \mathcal{L}, D_n)$*, otherwise let* $r_n = *$*. We use* $*$ *as a kind of null value.*

This sequence captures how the inconsistency of a theory Δ in a language $\mathcal{L}$ evolves with increasing domain size. At one extreme, if Δ is consistent, then $\mathsf{TheoryInc}(\Delta, \mathcal{L}) = \langle 0, 0, 0, ... \rangle$. At the other extreme, there are theories Δ such that $\mathsf{TheoryInc}(\Delta, \mathcal{L}) = \langle 1, 1, 1, ... \rangle$.

By Theorem 3, $\mathsf{TheoryInc}$ is well-defined.

Example 13. *Let* $\mathcal{L} = \langle \{P(1)\}, \{a\} \rangle$*. For* $\Delta = \{P(a), \neg P(a)\}$

$$\mathsf{TheoryInc}(\Delta, \mathcal{L}) = \langle 1, \tfrac{1}{2}, \tfrac{1}{3}, \ldots \rangle$$

Example 14. *Let* $\mathcal{L} = \langle\{P(1)\}, \{a, b\}\rangle$. *For* $\Delta = \{P(a), \neg P(a), P(b), \neg P(b)\}$

$$\mathsf{TheoryInc}(\Delta, \mathcal{L}) = \langle *, 1, \tfrac{2}{3}, \tfrac{1}{2}, \ldots\rangle$$

Example 15. *Let* $\mathcal{L} = \langle\{P(1)\}, \{a, b, c\}\rangle$. *For* $\Delta = \{P(a), \neg P(a), P(b), \neg P(b), P(c), \neg P(c)\}$

$$\mathsf{TheoryInc}(\Delta, \mathcal{L}) = \langle *, *, 1, \tfrac{3}{4}, \tfrac{3}{5}, \ldots\rangle$$

Example 16. *Let* $\mathcal{L} = \langle\{P(1), Q(1)\}, \{\}\rangle$.

$$\Delta_1 = \{\forall x(P(x) \wedge \neg P(x)), \forall x Q(x)\}$$
$$\Delta_2 = \{\exists x(P(x) \wedge \neg P(x)), \forall x Q(x)\}$$

Hence

$$\mathsf{TheoryInc}(\Delta_1, \mathcal{L}) = \langle \tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}, \ldots\rangle$$
$$\mathsf{TheoryInc}(\Delta_2, \mathcal{L}) = \langle \tfrac{1}{2}, \tfrac{1}{4}, \tfrac{1}{6}, \ldots\rangle$$

For a language $\mathcal{L}$ with k constants, the sequence obtained from $\mathsf{TheoryInc}(\Delta, \mathcal{L})$ starts with $k-1$ asterisks.

Proposition 1. *Let* $\mathsf{TheoryInc}(\Delta, \mathcal{L}) = \langle x_1, x_2, \ldots\rangle$. *If* $|\mathcal{C}| = k$, *and* $k \geq 1$, *then for all* $1 \leq i < k$, $x_i = *$ *and for all* $i \geq k$, $x_i \neq *$.

Proof. The reason for the asterisks is that the domain, according to our definition, must have at least as many elements as the number of constants in $\mathcal{L}$, that is, $|\mathcal{C}|$. It is always possible to increase the size of the domain. □

Proposition 2. *Let* $\mathsf{TheoryInc}(\Delta, \mathcal{L}) = \langle r_1, r_2, \ldots\rangle$. *If there is an* $r_i \in \{r_1, r_2, ...\}$ *such that* $r_i = 0$, *then* $\langle r_1, r_2, \ldots\rangle$ *is of the form* $\langle *, \ldots, *, 0, \ldots, 0\rangle$.

For a sequence $\langle r_1, r_2, ..\rangle$ for a theory Δ, if r_i is $*$, then the domain of size i is too small for the theory Δ. In this sense, r_i does not tell us anything useful about the inconsistencies in the theory. This leads to the following definition as capturing a natural ordering over sequences of inconsistency ratios.

Definition 18. *We can adopt an ordering, denoted by the* $\preceq$ *relation, over the tuples generated by the* $\mathsf{TheoryInc}$ *function. Let* $\mathsf{TheoryInc}(\Delta_1, \mathcal{L}_1) = \langle r_1, r_2, \ldots\rangle$ *and* $\mathsf{TheoryInc}(\Delta_2, \mathcal{L}_2) = \langle s_1, s_2, \ldots\rangle$.

$$\mathsf{TheoryInc}(\Delta_1, \mathcal{L}_1) \preceq \mathsf{TheoryInc}(\Delta_2, \mathcal{L}_2) \text{ iff for all } i \geq 1,\ r_i \leq s_i \text{ or } r_i = * \text{ or } s_i = *$$

Let $\mathsf{TheoryInc}(\Delta_1, \mathcal{L}_1) \prec \mathsf{TheoryInc}(\Delta_2, \mathcal{L}_2)$ *abbreviate* $\mathsf{TheoryInc}(\Delta_1, \mathcal{L}_1) \preceq \mathsf{TheoryInc}(\Delta_2, \mathcal{L}_2)$ *and* $\mathsf{TheoryInc}(\Delta_2, \mathcal{L}_2) \not\preceq \mathsf{TheoryInc}(\Delta_1, \mathcal{L}_1)$ *holding. In case* $\mathcal{L}_1 = \mathcal{L}_2$, *we say that* Δ_1 *has smaller than or equal inconsistency as* Δ_2 *iff* $\mathsf{TheoryInc}(\Delta_1, \mathcal{L}) \preceq \mathsf{TheoryInc}(\Delta_2, \mathcal{L})$. *We denote this by* $\Delta_1 \leq^{\mathcal{L}}_{inc} \Delta_2$.

Example 17. *Continuing Example 16,* $\Delta_2 \leq^{\mathcal{L}}_{inc} \Delta_1$.

Example 18. *Let* $\mathcal{L} = \langle\{P(1)\}, \{a\}\rangle$, $\Delta_1 = \{P(a)\}$, *and* $\Delta_2 = \{P(a), \neg P(a)\}$. *So* $\Delta_1 \leq^{\mathcal{L}}_{inc} \Delta_2$ *holds since we have the following.*

$$\mathsf{TheoryInc}(\Delta_1, \mathcal{L}) = \langle 0, 0, 0, \ldots\rangle$$
$$\mathsf{TheoryInc}(\Delta_2, \mathcal{L}) = \langle 1, \tfrac{1}{2}, \tfrac{1}{3}, \ldots\rangle$$

Given Δ and $\mathcal{L}$, it is not necessarily the case that $\mathsf{TheoryInc}(\Delta, \mathcal{L})\langle r_1, r_2, r_3, \ldots\rangle$ is such that for all $r_i \in \{r_1, r_2, r_3, \ldots\}$, if $r_i \neq *$, then $r_i \geq r_{i+1}$. This is illustrated in the next example.

Example 19. *Let* $\mathcal{L} = \langle\{P(2), Q(1)\}, \{\}\rangle$ *and let* $\Delta = \{\forall x \forall y(P(x, y) \wedge \neg P(x, y))\}$. *So,*

$$\mathsf{TheoryInc}(\Delta, \mathcal{L}) = \langle \tfrac{1}{2}, \tfrac{2}{3}, \tfrac{3}{4}, \tfrac{4}{5}, \ldots\rangle$$

The TheoryInc function is monotonic with respect to Δ and antimonotonic with respect to $\mathcal{L}$.

Proposition 3. *If* $\Delta_1 \subseteq \Delta_2$*, then* $\mathsf{TheoryInc}(\Delta_1, \mathcal{L}) \preceq \mathsf{TheoryInc}(\Delta_2, \mathcal{L})$

Proof. Additional statements may add but cannot subtract inconsistencies. □

Proposition 4. *Let* $\mathcal{L}_1 = \langle \mathcal{P}_1, \mathcal{C}_1 \rangle$*, and* $\mathcal{L}_2 = \langle \mathcal{P}_2, \mathcal{C}_2 \rangle$*. If* $\mathcal{L}_1 \subseteq \mathcal{L}_2$ *(i.e.* $\mathcal{P}_1 \subseteq \mathcal{P}_2$ *and* $\mathcal{C}_1 \subseteq \mathcal{C}_2$*), then* $\mathsf{TheoryInc}(\Delta, \mathcal{L}_2) \preceq \mathsf{TheoryInc}(\Delta, \mathcal{L}_1)$

Proof. For any D, $\mathsf{Groundatoms}(\mathcal{L}_1, D) \subseteq \mathsf{Groundatoms}(\mathcal{L}_2, D)$, but the size of the Conflictbase does not change. □

The TheoryInc function is not monotonic with increasing membership of both $\mathcal{L}$ and Δ. The next example shows that if $\Delta_1 \subseteq \Delta_2$ and $\mathcal{L}_1 \subseteq \mathcal{L}_2$ then it is not necessarily the case that $\mathsf{TheoryInc}(\Delta_1, \mathcal{L}_1) \preceq \mathsf{TheoryInc}(\Delta_2, \mathcal{L}_2)$ holds, nor is it necessarily the case that $\mathsf{TheoryInc}(\Delta_2, \mathcal{L}_2) \preceq \mathsf{TheoryInc}(\Delta_1, \mathcal{L}_1)$ holds.

Example 20. *Consider*

$$\begin{array}{ll} \mathcal{L}_1 = \langle \{P(1)\}, \{a\} \rangle & \Delta_1 = \{P(a)\} \\ \mathcal{L}_2 = \langle \{P(1)\}, \{a\} \rangle & \Delta_2 = \{P(a), \neg P(a)\} \\ \mathcal{L}_3 = \langle \{P(1), Q(1)\}, \{a\} \rangle & \Delta_3 = \{P(a), \neg P(a), Q(a)\} \end{array}$$

In this case

$$\mathsf{TheoryInc}(\Delta_1, \mathcal{L}_1) \prec \mathsf{TheoryInc}(\Delta_2, \mathcal{L}_2)$$
$$\mathsf{TheoryInc}(\Delta_3, \mathcal{L}_3) \prec \mathsf{TheoryInc}(\Delta_2, \mathcal{L}_2)$$

The TheoryInc function is syntax independent.

Definition 19. *For* $\Delta, \Delta' \in \wp(\mathcal{L})$*,* Δ *is QC-equivalent to* Δ' *iff*

$$\forall M (M \textit{ is a QC model of } \Delta \textit{ iff } M \textit{ is a QC model of } \Delta')$$

Proposition 5. *Let* Δ_1 *and* Δ_2 *be sets of formulae in the language* $\mathcal{L}$*. If* Δ_1 *is QC-equivalent to* Δ_2 *then* $\mathsf{TheoryInc}(\Delta_1, \mathcal{L}) = \mathsf{TheoryInc}(\Delta_2, \mathcal{L})$.

We now define the intrinsic inconsistency of a theory Δ by choosing for $\mathcal{L}$ exactly the predicate and constant symbols in Δ.

Definition 20. *For a given theory* Δ*, let* $\mathcal{L}$ *be the language that contains exactly the predicate and constant symbols of* Δ*. We define the* **intrinsic inconsistency of** Δ *as* $\mathsf{TheoryInc}(\Delta) = \mathsf{TheoryInc}(\Delta, \mathcal{L})$.

So the measure of intrinsic inconsistency of a theory $\mathsf{TheoryInc}(\Delta)$ delineates the degree of the theory in its own terms, whereas the extrinsic inconsistency of a theory $\mathsf{TheoryInc}(\Delta, \mathcal{L})$ delineates the degree of the theory with respect to the language $\mathcal{L}$. The following example shows how these different views can diverge.

Example 21. *Let* $\mathcal{L} = \langle \{P(1), Q(1)\}, \{\} \rangle$.

$$\Delta_1 = \{\forall x\ (P(x) \wedge \neg P(x)), \forall x\ (Q(x) \wedge \neg Q(x))\}$$
$$\Delta_2 = \{\forall x\ (P(x) \wedge \neg P(x))\}$$

Here $\mathsf{TheoryInc}(\Delta_1) = \mathsf{TheoryInc}(\Delta_2) = \langle 1, 1, \ldots \rangle$ *but* $\mathsf{TheoryInc}(\Delta_2, \mathcal{L}) \prec \mathsf{TheoryInc}(\Delta_1, \mathcal{L})$ *because the addition of* Q *to the language reduces the inconsistency of* Δ_2*.*

We are now ready to compare the inconsistency of theories.

Definition 21. *We say that Δ_1 has smaller than or equal inconsistency as Δ_2, denoted $\Delta_1 \leq_{inc} \Delta_2$, iff* $\mathsf{TheoryInc}(\Delta_1) \preceq \mathsf{TheoryInc}(\Delta_2)$ *holds.*

Clearly, all consistent theories[2] have an inconsistency of the form $\langle *, \ldots, *, 0, \ldots, 0\rangle$, with the number of $*$s depending on the number of constants in the language. Hence, from our definition, their inconsistency is the same. Also every inconsistent theory has greater inconsistency than any consistent theory.

We now illustrate the comparison of various inconsistent theories by focusing on the workings of the quantifiers of first-order logic.

Example 22. *Let $\mathcal{L} = \langle\{P(2)\}, \{\}\rangle$ and $D = \{a, b, c\}$.*

1. $\Delta_1 = \{\forall x \forall y(P(x, y) \wedge \neg P(x, y))\}$ *has one preferred QC model which is represented by* $M_1 = \{P(a, a), \neg P(a, a), \ldots, P(c, c), \neg P(c, c)\}$, *so* $\mathsf{ModelInc}(M_1, \mathcal{L}, D) = \frac{9}{9} = 1$. M_1 *is totally inconsistent.*
2. $\Delta_2 = \{\exists x \exists y(P(x, y) \wedge \neg P(x, y))\}$ *has 9 preferred QC models. One of them is* $M_{21} = \{P(a, b), \neg P(a, b)\}$, *so* $\mathsf{ModelInc}(M_{21}, \mathcal{L}, D) = \frac{1}{9}$.
3. $\Delta_3 = \{\forall x \exists y(P(x, y) \wedge \neg P(x, y))\}$ *has 9 preferred QC models. One is* $M_{31} = \{P(a, a), \neg P(a, a), P(b, c), \neg P(b, c), P(c, a), \neg P(c, a)\}$, *so* $\mathsf{ModelInc}(M_{31}, \mathcal{L}, D) = \frac{3}{9} = \frac{1}{3}$.
4. $\Delta_4 = \{\exists x \forall y(P(x, y) \wedge \neg P(x, y))\}$ *has 9 preferred QC models. One is* $M_{41} = \{P(b, a), \neg P(b, a), P(b, b), \neg P(b, b), P(b, c), \neg P(b, c)\}$, *so* $\mathsf{ModelInc}(M_{41}, \mathcal{L}, D) = \frac{3}{9} = \frac{1}{3}$.

The theories above have no constants, hence the smallest domain has one element. As a result, we obtain the following

$$\begin{array}{l}
\mathsf{TheoryInc}(\Delta_1) = \langle 1, 1, \ldots\rangle \\
\mathsf{TheoryInc}(\Delta_2) = \langle 1, \frac{1}{4}, \frac{1}{9}, \ldots\rangle \textit{ where each ratio is } \frac{1}{n^2} \\
\mathsf{TheoryInc}(\Delta_3) = \langle 1, \frac{1}{2}, \frac{1}{3}, \ldots\rangle \textit{ where each ratio is } \frac{1}{n} \\
\mathsf{TheoryInc}(\Delta_4) = \langle 1, \frac{1}{2}, \frac{1}{3}, \ldots\rangle \textit{ where each ratio is } \frac{1}{n}
\end{array}$$

So $\mathsf{TheoryInc}(\Delta_2) <_{inc} \mathsf{TheoryInc}(\Delta_3) =_{inc} \mathsf{TheoryInc}(\Delta_4) <_{inc} \mathsf{TheoryInc}(\Delta_1)$. *In fact, Δ_1 is as inconsistent as any theory can be.*

We now show how to get some sequences for $\mathsf{TheoryInc}(\Delta)$.

Proposition 6. *For any fraction $\frac{r}{s}$ $(r \leq s)$ there is a theory Δ such that* $\mathsf{TheoryInc}(\Delta) = \langle \frac{r}{s}, \frac{r}{2s}, \frac{r}{3s}, \ldots\rangle$

Proof. Let $\Delta = \{\forall x\ (P_1(x) \wedge \ldots \wedge P_s(x)), \neg P_1(a), \ldots, \neg P_r(a)\}$. □

Both the intrinsic and extrinsic measures of inconsistency have a variety of potential applications. In Section 6, we consider analysing sources of information as a precursor to selecting sources for merging. To support this application area, we introduce the notion of quasi-equality in the next section.

[2]These consistent theories exclude theories with equality where $t_1 = t_2$ holds in a model iff t_1 and t_2 denote the same object. In Section 5, we return to issues of capturing a form of equality in QC logic.

5 Reasoning with quasi-equality

Reasoning with equality is a problematical and important issue in handling inconsistent first-order information. In this section, we show how we can capture a form of equality, called quasi-equality, in first-order QC logic, and then show how we can analyse inconsistencies in reasoning with equality.

In classical logic with equality, for ground terms t_1 and t_2, t_1 equals t_2 is denoted $t_1 = t_2$, and it has the following semantics: $t_1 = t_2$ is true in an interpretation iff t_1 and t_2 denote the same individual in the interpretation. In contrast, in QC logic we have assumed (in Section 2) that each constant symbol is interpreted by being mapped to itself in the domain. This means that even if two constant symbols are synonyms for the same thing in the real world, the constant symbols are not mapped to the same object in the model. So to capture a form of equality in QC logic, we use the notion of a quasi-equality relation, denoted $\simeq$, holding in the context of some knowledge Δ: So "a is equal to b", denoted $a \simeq b$, in the context of Δ iff $a \simeq b \in M$ for all $M \in \mathsf{QC}(\mathcal{L}, \Delta, D)$. Hence, we treat the quasi-equality predicate as any other predicate in $\mathcal{L}$. To support this, we still maintain the assumption (in Section 2) that the set of constant symbols in the language $\mathcal{L}$ is a subset of the domain D.

For convenience we write $t_1 \not\simeq t_2$ instead of $\neg(t_1 \simeq t_2)$.

Example 23. *For $\Delta = \{a \simeq b, b \not\simeq c\}$, we have $\{a, b, c\} \subseteq D$.*

However to get the behaviour normally associated with equality, we need to add extra axioms to a knowledgebase: the reflexivity axiom, the symmetry axiom, and the transitivity axiom. It also includes a unique names axiom that states that each constant symbol is assumed to be unique in a knowledgebase [Rei78]. Our definition for quasi-equality is similar to the idea of indiscernibility in classical logic.

Definition 22. *For $\mathcal{L} = \langle\{P_1(k_1), \ldots, P_z(k_z), \simeq(2)\}, \{c_1, .., c_n\}\rangle$, the set of* **quasi-equality axioms**, *denoted Δ^{qe}, is defined as follows, where we assume that for $1 \leq m \leq z$, P_m does not denote $\simeq$.*

- *Reflexivity axiom*
$$\forall x\ (x \simeq x)$$
- *Symmetry axiom*
$$\forall x, y\ ((x \not\simeq y) \vee (y \simeq x))$$
- *Transitivity axiom*
$$\forall x, y, z\ ((x \not\simeq y) \vee (y \not\simeq z) \vee (x \simeq z))$$
- *Unique names axiom*
$$\bigwedge_{c_i, c_j \in \{c_1, .., c_n\} \text{ such that } c_i \neq c_j} c_i \not\simeq c_j$$
- *Substitution axiom*

$$\forall x_1, \ldots, x_n, y_1, \ldots, y_n(\neg P_m(x_1, ..., x_n) \vee x_1 \not\simeq y_1 \vee ... \vee x_n \not\simeq y_n \vee P_m(y_1, ..., y_n))$$

Example 24. *Let $\mathcal{L} = \langle\{P(2), \simeq(2)\}, \{a, b, c\}\rangle$. The unique names axiom for $\mathcal{L}$ is*

$$(a \not\simeq b) \wedge (a \not\simeq c) \wedge (b \not\simeq a) \wedge (b \not\simeq c) \wedge (c \not\simeq a) \wedge (c \not\simeq b)$$

Suppose $\Delta_1 = \{a \simeq b, b \simeq c)\}$. There is one preferred QC model of $\Delta_1 \cup \Delta^{qe}$ which is below.

$$\{a \simeq a, a \simeq b, a \not\simeq b, a \simeq c, a \not\simeq c, b \simeq a, b \not\simeq a, b \simeq b, \\ b \simeq c, b \not\simeq c, c \simeq a, c \not\simeq a, c \simeq b, c \not\simeq b, c \simeq c\}$$

Next let $\Delta_2 = \{P(a, c), \neg P(b, b), a \simeq b\}$. The one preferred model of $\Delta_2 \cup \Delta^{qe}$ is as follows:

$$\{P(a, c), P(b, c), \neg P(a, a), \neg P(a, b), \neg P(b, a), \neg P(b, b) \\ a \simeq a, a \simeq b, a \not\simeq b, a \not\simeq c, b \simeq a, b \not\simeq a, b \simeq b, b \not\simeq c, c \not\simeq a, c \not\simeq b, c \simeq c\}$$

In some domains we know for sure what is equal and not equal. So in some domains we may want something closer to equality as in classical logic. We could extend QC logic to incorporate a "built-in" predicate capturing "real" equality, denoted $=$, in such a way that no QC model is allowed to have both $a = b$ and $a \neq b$ for any a or b. If we deal with integers we know that $5 = 5$ and would not accept $5 = 6$ or $5 \neq 5$. This is similar to what happens if we have other built-in predicates such as "less than". We know that $5 < 6$ and would not allow $6 < 5$. Adopting such built-in predicates requires some alterations to the definitions in Section 2. We do not provide the revised definitions in this paper, but give a simple example below to show the difference between using equality and quasi-equality in an integrity constraint.

Example 25. *Let* $\mathcal{L} = \langle\{P(3)\}, \{a, b, c\}\rangle$, $D = \{a, b, c\}$, *and*

$$\Delta_1 = \{\forall x \forall y \forall z \forall u \forall v(\neg P(x, y, z) \vee \neg P(x, u, v) \vee y = u), P(a, b, c), P(a, c, c)\}$$

The universal statement represents the functional dependency of the second attribute of P *on the first attribute. So any two tuples in* P *that have the same first value must have second values that are equal. Because of our earlier statement that inconsistencies are not allowed for equalities,* Δ_1 *has no models. Next, let*

$$\Delta_2 = \{\forall x \forall y \forall z \forall u \forall v(\neg P(x, y, z) \vee \neg P(x, u, v) \vee y \simeq u), P(a, b, c), P(a, c, c)\}$$

In contrast to Δ_1, Δ_2 *has models such as the one below:*

$$\begin{array}{l}\{P(a,b,c), P(a,b,b), P(a,c,b), P(a,c,c), \neg P(a,a,a), \neg P(a,a,b), \neg P(a,a,c)\\ \quad \neg P(a,b,a), \neg P(a,b,b), \neg P(a,b,c), \neg P(a,c,a), \neg P(a,c,b), \neg P(a,c,c)\\ \quad a \simeq a, a \not\simeq a, a \not\simeq b, a \not\simeq c, b \not\simeq a, b \simeq b, b \not\simeq b, b \simeq c, b \not\simeq c, c \not\simeq a, c \simeq b, c \not\simeq b, c \simeq c, c \not\simeq c\}\end{array}$$

So with the "real" equality predicate, we may represent information involving equality that we may not tolerate as inconsistent, or we can assume will not be inconsistent. This has various advantages including the fact that it can be used to provide constraints on the size and composition of the domain. It can also be used to specify a uniqueness quantifier $\exists x!$ so that for example $\exists x! P(x)$ means that there is exactly one element in the domain that has the property P. This can be defined in first-order classical logic as follows.

$$\exists x! P(x) \equiv_{def} \exists x \forall y(x = y \leftrightarrow P(x))$$

There are disadvantages to treating equality as "real" equality. Because we can express statements about the number of elements in a model in first-order logic with real equality, the inconsistency sequences become more complicated and may have a more complex pattern of asterisks as the number of elements in any model can be limited. Perhaps more significantly, some of the key results regarding QC logic would be compromised. For example, as we shown in the example above, some theories have no models at all.

6 Analysing sources of information

Using information from heterogeneous sources is an increasingly important topic in computing. This includes information integration (e.g. [SL90, Hal01]), knowledgebase merging (e.g [BKMS92, KP98]), and query answering in inconsistent databases (e.g. [BC03]). The central problem in using such information is the preponderance of inconsistencies.

Most proposals for using such information offer techniques for addressing the inconsistencies that may arise, but none consider analysing the information before using a technique. Yet analysing information first means that we can be selective in the information we use. As a simple example, if a knowledgebase Δ is such that $\mathsf{TheoryInc}(\Delta) = \langle 1, 1, \ldots\rangle$, we may be reluctant to use it if we have other knowledgebases that are significantly less inconsistent.

Definition 23. *Let Δ^{db} be a set of ground predicates in $\mathcal{L}$ denoting a database, let Δ^{ic} be any set of formulae in $\mathcal{L}$ denoting a set of integrity constraints, and let Δ^{qe} be the set of quasi-equality axioms.*

$$\Delta^{db} \text{ is consistent w.r.t. } \Delta^{ic} \text{ iff } \Delta^{db} \cup \Delta^{ic} \cup \Delta^{qe} \text{ is consistent}$$

If a database is inconsistent with a set of integrity constraints, then we want to measure the degree of inconsistency. Furthermore, we want to measure both the intrinsic inconsistency and the extrinsic inconsistency. The intrinsic inconsistency can be viewed as the inconsistency of a source in terms of its own language and that of its integrity constraints. The extrinsic inconsistency can be viewed as the inconsistency of a source in terms of the language of all the sources and all the integrity constraints.

In the following examples, we illustrate some of the useful insights that intrinsic and extrinsic measures of inconsistency can provide in analysing sources of information.

Example 26. *Let $\mathcal{L} = \langle \mathcal{P}, \mathcal{C} \rangle$ where $\mathcal{P} = \{Married(2)\}$ and $\mathcal{C}$ is the following set.*

$$\{AnnJones, AlanJones, MarySmith, FredSmith, PeterChan, SueChan\}$$

Using this language we can obtain a couple of simple databases that we can use to show how violations of an integrity constraint may be analysed.

$$\Delta^{ic} = \{\forall x \forall y \ (\neg Married(x, y) \vee Married(y, x))\}$$

We consider the following two databases.

$$\begin{aligned} \Delta_1^{db} &= \{Married(AnnJones, AlanJones), \neg Married(AlanJones, AnnJones)\} \\ \Delta_2^{db} &= \{Married(MarySmith, FredSmith), \neg Married(MarySmith, FredSmith), \\ &\qquad Married(PeterChan, SueChan)\} \end{aligned}$$

Using these, we consider the following two theories

$$\begin{aligned} \Delta_1 &= \Delta^{ic} \cup \Delta_1^{db} \\ \Delta_2 &= \Delta^{ic} \cup \Delta_2^{db} \end{aligned}$$

For these we get the following valuations for the TheoryInc *functions.*

$$\begin{aligned} \mathsf{TheoryInc}(\Delta_1) &= \langle *, \tfrac{2}{4}, \tfrac{2}{9}, \tfrac{2}{16}, \ldots \rangle \\ \mathsf{TheoryInc}(\Delta_2) &= \langle *, *, *, \tfrac{2}{16}, \tfrac{2}{64}, \ldots \rangle \end{aligned}$$

By inspection of the sequences obtained by the TheoryInc *function, we see that maximum inconsistency ratio for Δ_1 is $2/4$, whereas for Δ_2 it is $2/16$, which reflects that more of Δ_1 is involved in an inconsistency than Δ_2. Then by comparing the sequences using the $\preceq$ relation, we see the theories are equally inconsistent. In other words, as more of the domain is taken into account, we see that at each cardinality of the domain, the inconsistency ratio is the same.*

Example 27. *Let $\mathcal{L} = \langle \mathcal{P}, \mathcal{C} \rangle$ where $\mathcal{P} = \{Employee(1), Supervisor(2), Department(2), Role(3)\}$ and $\mathcal{C} = \{AnnJones, FredSmith, MikeBollo\}$. Using this language we can obtain a simple database that we can use to show how the language can affect the analysis of violations of an integrity constraint.*

$$\Delta^{ic} = \{\forall x \exists y \ (\neg Employee(x) \vee Supervisor(y, x))\}$$

We consider the following database.

$$\begin{aligned} \Delta^{db} = \{&Employee(FredSmith), \neg Supervisor(AnnJones, FredSmith), \\ &\neg Supervisor(FredSmith, FredSmith), \\ &\neg Supervisor(MikeBollo, FredSmith)\} \end{aligned}$$

Using these, we consider the following theory

$$\Delta = \Delta^{ic} \cup \Delta^{db}$$

For these we get the following valuations for the TheoryInc *functions.*

$$\begin{array}{ll} \mathsf{TheoryInc}(\Delta, \mathcal{L}) & = \langle *, *, \frac{2}{48}, \frac{2}{100}, \frac{2}{180}, \ldots \rangle \\ \mathsf{TheoryInc}(\Delta) & = \langle *, *, \frac{2}{12}, \frac{2}{20}, \frac{2}{30}, \ldots \rangle \end{array}$$

By inspection of the sequences obtained by the TheoryInc *functions, we see that as the domain increases, the inconsistency ratio rapidly diminishes. This is markedly so when we take the language into account. This indicates that if Δ is a true reflection of all the inconsistencies in the data in this language, then the proportion of data that is affected is small.*

We now consider a more complex example in more detail.We will use quasi-equality for these examples.

Example 28. *Let $\mathcal{L} = \langle \{P(2), Q(2), \simeq(2)\}, \{a, b, c, d\} \rangle$. Below we give a set of integrity constraints and three sets of data.*

$$\begin{array}{l} \Delta_1^{ic} = \{\forall x \forall y \ (\neg P(x, y) \vee \neg P(x, z) \vee y \simeq z)\} \\ \Delta_2^{ic} = \{\neg P(a, b) \vee \neg P(c, d)\} \\ \Delta_3^{ic} = \{\forall x \forall y \forall z \ (\neg P(x, y) \vee \neg Q(x, z) \vee y \simeq z)\} \end{array}$$

$$\begin{array}{l} \Delta_1^{db} = \{P(a, b), P(a, c)\} \\ \Delta_2^{db} = \{P(c, d), \neg P(c, d)\} \\ \Delta_3^{db} = \{Q(a, b), Q(a, c), P(a, c)\} \end{array}$$

We consider the following theories

$$\begin{array}{l} \Delta_1 = \Delta_1^{ic} \cup \Delta_1^{db} \cup \Delta^{qe} \\ \Delta_2 = \Delta_2^{ic} \cup \Delta_2^{db} \cup \Delta^{qe} \\ \Delta_3 = \Delta_3^{ic} \cup \Delta_3^{db} \cup \Delta^{qe} \\ \Delta_{12} = \Delta_1^{ic} \cup \Delta_2^{ic} \cup \Delta_1^{db} \cup \Delta_2^{db} \cup \Delta^{qe} \\ \Delta_{13} = \Delta_1^{ic} \cup \Delta_3^{ic} \cup \Delta_1^{db} \cup \Delta_3^{db} \cup \Delta^{qe} \\ \Delta_{23} = \Delta_2^{ic} \cup \Delta_3^{ic} \cup \Delta_2^{db} \cup \Delta_3^{db} \cup \Delta^{qe} \\ \Delta_{123} = \Delta_1^{ic} \cup \Delta_2^{ic} \cup \Delta_3^{ic} \cup \Delta_1^{db} \cup \Delta_2^{db} \cup \Delta_3^{db} \cup \Delta^{qe} \end{array}$$

In this example the number of inconsistent atoms does not change as the size of the domain changes. Hence the numerators of the fractions do not change, only the denominators change as larger domains have more atoms. First we list for each theory the atoms in the conflictbase; the number of these will be the number in the numerator of the fractions.

$$\begin{array}{l} \forall M \in \mathsf{MQC}(\Delta_1) \ \mathsf{Conflictbase}(M) = \\ \qquad \{P(a, b), P(a, c), a \simeq a, b \simeq b, c \simeq c, b \simeq c, c \simeq b\} \\ \forall M \in \mathsf{MQC}(\Delta_2) \ \mathsf{Conflictbase}(M) = \\ \qquad \{P(c, d), c \simeq c, d \simeq d\} \\ \forall M \in \mathsf{MQC}(\Delta_3) \ \mathsf{Conflictbase}(M) = \\ \qquad \{P(a, c), Q(a, b), Q(a, c), a \simeq a, b \simeq b, c \simeq c, b \simeq c, c \simeq b\} \\ \forall M \in \mathsf{MQC}(\Delta_{12}) \ \mathsf{Conflictbase}(M) = \\ \qquad \{P(a, b), P(a, c), P(c, d), a \simeq a, b \simeq b, c \simeq c, d \simeq d, b \simeq c, c \simeq b\} \\ \forall M \in \mathsf{MQC}(\Delta_{13}) \ \mathsf{Conflictbase}(M) = \\ \qquad \{P(a, b), P(a, c), Q(a, b), Q(a, c), a \simeq a, b \simeq b, c \simeq c, b \simeq c, c \simeq b\} \\ \forall M \in \mathsf{MQC}(\Delta_{23}) \ \mathsf{Conflictbase}(M) = \\ \qquad \{P(a, c), P(c, d), Q(a, b), Q(a, c), a \simeq a, b \simeq b, c \simeq c, d \simeq d, b \simeq c, c \simeq b\} \\ \forall M \in \mathsf{MQC}(\Delta_{123}) \ \mathsf{Conflictbase}(M) = \\ \qquad \{P(a, b), P(a, c), P(c, d), Q(a, b), Q(a, c), a \simeq a, b \simeq b, c \simeq c, d \simeq d, b \simeq c, c \simeq b\} \end{array}$$

Hence we get the following evaluations of extrinsic inconsistency. The fractions are not simplified. The numerator is the cardinality of the conflictbase for the models considered and denominator is the cardinality of the set of ground atoms for the language and domain considered.

$$\begin{array}{ll}
\mathsf{TheoryInc}(\Delta_1, \mathcal{L}) & = \langle *, *, *, \frac{7}{48}, \frac{7}{75}, \ldots \rangle \\
\mathsf{TheoryInc}(\Delta_2, \mathcal{L}) & = \langle *, *, *, \frac{3}{48}, \frac{3}{75}, \ldots \rangle \\
\mathsf{TheoryInc}(\Delta_3, \mathcal{L}) & = \langle *, *, *, \frac{8}{48}, \frac{8}{75}, \ldots \rangle \\
\mathsf{TheoryInc}(\Delta_{12}, \mathcal{L}) & = \langle *, *, *, \frac{9}{48}, \frac{9}{75}, \ldots \rangle \\
\mathsf{TheoryInc}(\Delta_{13}, \mathcal{L}) & = \langle *, *, *, \frac{9}{48}, \frac{9}{75}, \ldots \rangle \\
\mathsf{TheoryInc}(\Delta_{23}, \mathcal{L}) & = \langle *, *, *, \frac{10}{48}, \frac{10}{75}, \ldots \rangle \\
\mathsf{TheoryInc}(\Delta_{123}, \mathcal{L}) & = \langle *, *, *, \frac{11}{48}, \frac{11}{75}, \ldots \rangle
\end{array}$$

From the measure of extrinsic inconsistency, source Δ_2 is clearly the least inconsistent. If we want to take pairs of sources then $\Delta_2 \cup \Delta_3$, is slightly more inconsistent than $\Delta_1 \cup \Delta_2$ and $\Delta_1 \cup \Delta_3$.

We also get the following evaluations of intrinsic inconsistency. Again the fractions are not simplified.

$$\begin{array}{ll}
\mathsf{TheoryInc}(\Delta_1) & = \langle *, *, \frac{7}{18}, \frac{7}{32}, \frac{7}{50}, \ldots \rangle \\
\mathsf{TheoryInc}(\Delta_2) & = \langle *, *, *, \frac{3}{32}, \frac{3}{50}, \ldots \rangle \\
\mathsf{TheoryInc}(\Delta_3) & = \langle *, *, \frac{8}{27}, \frac{8}{48}, \frac{8}{75}, \ldots \rangle \\
\mathsf{TheoryInc}(\Delta_{12}) & = \langle *, *, *, \frac{9}{32}, \frac{9}{50}, \ldots \rangle \\
\mathsf{TheoryInc}(\Delta_{13}) & = \langle *, *, \frac{9}{27}, \frac{9}{48}, \frac{9}{75}, \ldots \rangle \\
\mathsf{TheoryInc}(\Delta_{23}) & = \langle *, *, *, \frac{10}{48}, \frac{10}{75}, \ldots \rangle \\
\mathsf{TheoryInc}(\Delta_{123}) & = \langle *, *, *, \frac{11}{48}, \frac{11}{75}, \ldots \rangle
\end{array}$$

With intrinsic inconsistency, Δ_2 is still the least inconsistent source, but the difference between Δ_2 and Δ_3 is less pronounced. Also, among pairs of sources Δ_{12} and Δ_{13} are no longer tied and Δ_{13} is the least inconsistent. The reason for the changes is that the language of Δ_1 and Δ_2 is more restricted than the language of Δ_3.

We now consider an example of comparing two sources of first-order knowledge each of which is inconsistent with a set of beliefs, and the task is to select the source that is least inconsistent with the beliefs.

Example 29. *Let a set of beliefs be represented by the set Δ as follows*

$$\Delta = \{\forall x \forall y\ \alpha(x, y), \forall x \forall y \neg \beta(x, y), \forall x \forall y \gamma(x, y)\}$$

Let Γ_1 and Γ_2 be two sources of information defined as follows.

$$\begin{array}{c}
\Gamma_1 = \{\forall x \exists y(\alpha(x, y) \rightarrow \beta(x, y))\} \\
\Gamma_2 = \{\forall x \forall y(\alpha(x, y) \wedge \gamma(x, y) \rightarrow \beta(x, y))\}
\end{array}$$

Now we consider the following theories.

$$\begin{array}{c}
\Delta_1 = \Gamma_1 \cup \Delta \\
\Delta_2 = \Gamma_2 \cup \Delta
\end{array}$$

For these theories, we get the following evaluations of intrinsic inconsistency.

$$\begin{array}{ll}
\mathsf{TheoryInc}(\Delta_1) & = \langle \frac{2}{3}, \frac{4}{12}, \frac{6}{27}, \frac{8}{48}, \ldots \rangle \\
\mathsf{TheoryInc}(\Delta_2) & = \langle \frac{3}{3}, \frac{12}{12}, \frac{27}{27}, \frac{48}{48}, \ldots \rangle
\end{array}$$

From these evaluations of intrinsic inconsistency, we see that Δ_1 is the less inconsistent source, and by this criterion, it should be the selected source.

From these examples, we see that both the measure of intrinsic inconsistency and the measure of extrinsic inconsistency provide the basis of objective criteria for evaluating the quality of sources of information, and that this can then be used to improve the quality of merged information.

7 Discussion

In this paper, we have presented a first-order version of QC measurement of inconsistency. This version, together with the proposal for quasi-equality offers a useful approach to analysing inconsistent information in knowledgebases.

Whilst a number of approaches to handling inconsistent information touch on the issue of measurement of inconsistency, the topic is underdeveloped. Information theory can be used to measure the information content of sets of inconsistent formulae [Loz94]. This increases with additions of consistent information and decreases with additions of inconsistent information. However, it does not provide a direct measure of inconsistency since for example, the value for $\{\alpha\}$ is the same as for $\{\alpha, \neg\alpha, \beta\}$. Another approach to handling inconsistent information is that of possibility theory [DLP94]. Let (ϕ, α) be a weighted formula where ϕ is a classical formula and $\alpha \in [0, 1]$. A possibilistic knowledgebase B is a set of weighted formulae. An α-cut of a possibilistic knowledgebase, denoted $B_{\geq\alpha}$, is $\{(\psi, \beta) \in B \mid \beta \geq \alpha\}$. The inconsistency degree of B, denoted $Inc(B)$, is the maximum value of α such that the α-cut is inconsistent. However, this approach does not discriminate between different inconsistencies. For this, there is a need for an underlying paraconsistent logic such as QC logic.

An alternative approach to quantifying degrees of information and contradiction in propositional logic is based on a framework of "epistemic actions" [KLM03]. The degree of information in a knowledgebase is based on the number (or the cost) of actions needed to identify the truth value of each atomic proposition: the lower the cost, the more information is contained in the base. The degree of contradiction in a knowledgebase is based on the number (or the cost) of actions needed to render the knowledgebase classically consistent. Both measurements are dependent on the language, logic, and tests used. This is a very general framework for propositional logic. Our work goes beyond theirs by providing a comprehensive analysis of intrinsic and extrinsic inconsistency in the predicate case.

In this paper, we have shown how the QC measurement of inconsistency may be useful in analysing violations of integrity constraints, and in comparing sources of information prior to merging. Another potentially important application is in supporting negotiation. Consider a negotiation between two agents. Each agent represents their position by a set of first-order formulae and the union of these positions is inconsistent. The aim of negotiation is for the agents to shift their respective positions to ones where the union is consistent. Normally this involves a number of cycles of compromises by the agents. Each compromise is effectively revising a position to a logically weaker position. So agent 1 may make a compromise, resulting in a lowering in the degree of inconsistency. Then agent 2 may make a compromise, resulting in a further lowering in the degree of inconsistency. This should continue until sufficient compromises have been made to give a consistent union. Our theoretical framework can be used to ensure that the degree of inconsistency is lowered in each cycle, and more importantly to ensure that the compromises are equitable. In other words, using instrinsic and extrinsic measures of inconsistency, we can ensure that if agent 1 reduces inconsistency by a certain degree, then agent 2 should also reduce inconsistency by the same degree. This commensurate compromising can be enforced at each cycle, or enforced over all the cycles. Furthermore, we can set a threshold for what is a minimum degree of compromise per cycle to ensure that the total number of cycles is minimized.

Acknowlegements

The authors are grateful to the referee for important comments and suggestions that led to a substantial improvement in the paper.

References

[AA98] O Arieli and A. Avron. The value of the four values. *Artificial Intelligence*, 102:97–141, 1998.

[BC03] L Bertossi and J Chomicki. Query answering in inconsistent databases. In G. Saake J. Chomicki and R. van der Meyden, editors, *Logics for Emerging Applications of Databases*. Springer, 2003.

[Bel77] N Belnap. A useful four-valued logic. In G Epstein, editor, *Modern Uses of Multiple-valued Logic*, pages 8–37. Reidel, 1977.

[BH95] Ph Besnard and A Hunter. Quasi-classical logic: Non-trivializable classical reasoning from inconsistent information. In *Symbolic and Quantitative Approaches to Uncertainty*, volume 946 of *LNCS*, pages 44–51, 1995.

[BKMS92] C Baral, S Kraus, J Minker, and V Subrahmanian. Combining knowledgebases of first-order theories. *Computational Intelligence*, 8:45–71, 1992.

[CM02] W Carnielli and J Marcos. A taxonomy of C systems. In *Paraconsistency: The Logical Way to the Inconsistent*, pages 1–94. Marcel Dekker, 2002.

[dC74] N C da Costa. On the theory of inconsistent formal systems. *Notre Dame Journal of Formal Logic*, 15:497–510, 1974.

[DLP94] D Dubois, J Lang, and H Prade. Possibilistic logic. In *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 3, pages 439–513. Oxford University Press, 1994.

[Gra78] J Grant. Classifications for inconsistent theories. *Notre Dame Journal of Formal Logic*, 19:435–444, 1978.

[GS00] J Grant and V. S. Subrahmanian. Applications of paraconsistency in data and knowledge bases. *Synthese*, 125:121–132, 2000.

[Hal01] A Halevy. Answering queries using views: A survey. *VLDB Journal*, 10(4), 2001.

[HN98] A Hunter and B Nuseibeh. Managing inconsistent specifications: Reasoning, analysis and action. *ACM Transactions on Software Engineering and Methodology*, 7:335–367, 1998.

[Hun98] A Hunter. Paraconsistent logics. In *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, volume 2, pages 11–36. Kluwer, 1998.

[Hun00a] A Hunter. Reasoning with conflicting information using quasi-classical logic. *Journal of Logic and Computation*, 10:677–703, 2000.

[Hun00b] A Hunter. Reasoning with inconsistency in structured text. *Knowledge Engineering Review*, 15:317–337, 2000.

[Hun01] A Hunter. A semantic tableau version of first-order quasi-classical logic. In *Symbolic and Quantitative Approaches to Uncertainty*, volume 2143 of *LNCS*, pages 544–556, 2001.

[Hun02] A Hunter. Measuring inconsistency in knowledge via quasi-classical models. In *Proceedings of the National Conference on Artificial Intelligence (AAAI'02)*, pages 68–73. MIT Press, 2002.

[Hun03] A Hunter. Evaluating significance of inconsistencies. In *Proceedings of the 18th International Joint Conference on Artificial Intellignce (IJCAI'03)*, pages 468–473, 2003.

[KLM03] S Konieczny, J Lang, and P Marquis. Quantifying information and contradiction in propositional logic through epistemic actions. In *Proceedings of the 18th International Joint Conference on Artificial Intellignce (IJCAI'03)*, pages 106–111, 2003.

[KP98] S Konieczny and R Pino Perez. On the logic of merging. In *Proceedings of the Sixth International Conference on Principles of Knowledge Representation and Reasoning (KR98)*, pages 488–498. Morgan Kaufmann, 1998.

[Lev84] H Levesque. A logic of implicit and explicit belief. In *Proceedings of the National Conference on Artificial Intelligence (AAAI'84)*, pages 198–202, 1984.

[Loz94] E Lozinskii. Information and evidence in logic systems. *Journal of Experimental and Theoretical Artificial Intelligence*, 6:163–193, 1994.

[MDB02] R Miarka, J Derrick, and E Boiten. Handling inconsistencies in z using quasi-classical logic. In D Bert, J Bowen, M Henson, and K Robinson, editors, *ZB2002: Formal Specification and Development in Z and B*, volume 2272 of *Lecture Notes in Computer Science*, pages 204–225. Springer, 2002.

[MP01] P Marquis and N Porquet. Computational aspects of quasi-classical entailment. *Journal of Applied Non-classical Logics*, 11:295–312, 2001.

[Pri89] G Priest. Reasoning about truth. *Artificial Intelligence*, 39:231–244, 1989.

[Pri02] G Priest. Paraconsistent logic. In *Handbook of Philosophical Logic*, volume 6. Kluwer, 2002.

[Rei78] R Reiter. Equality and domain closure. *J. ACM*, 27:235–249, 1978.

[SL90] A Sheth and J Larson. Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Computing Surveys*, 22:183–236, 1990.

[Smu68] R Smullyan. *First-order Logic*. Springer, 1968.