SCISPACE
formerly Typeset

Open access • Journal Article • DOI:10.1093/JIGPAL/9.2.217

Paraconsistent reasoning as an analytic tool — Source link

Paul Wong, Philippe Besnard

Published on: 01 Mar 2001 - Logic Journal of the IGPL (Oxford University Press)

Topics: Paraconsistent logic, Computational logic, Philosophy of logic, Higher-order logic and Predicate logic

Related papers:

- Paraconsistent reasoning as an analytic tool
- Measuring inconsistency in knowledge via quasi-classical models
- Classifications for inconsistent theories.
- Quantifying information and contradiction in propositional logic through test actions
- Two Information Measures for Inconsistent Sets

Share this paper:

View more about this paper here: https://typeset.io/papers/paraconsistent-reasoning-as-an-analytic-tool-1jmo1m179e

Paraconsistent Reasoning as an Analytic Tool

PAUL WONG, *Automated Reasoning Group, Computer Science Laboratory, Research School of Information Sciences and Engineering, Australian National University, Canberra ACT 0200 Australia*
E-mail: wongas@arp.anu.edu.au

PHILIPPE BESNARD, *Applied Logic Group IRIT, Université Paul Sabatier 118 route de Narbonne, F-31062 Toulouse cedex 4 France*
E-mail: besnard@irit.fr

Abstract

The study of logic usually focuses on either the proof theoretic or the model theoretic properties of logic. Yet the *pragmatics* of logic is often ignored. In this paper we would like to demonstrate that a logic can be practical in the sense that it can assist us in evaluating and measuring the amount of information in an inconsistent set of data. The underlying notion of information is inspired by Shannon's communication theory. It defines the amount of information of a message in terms of the probability of the message being true. The logic presented here is the paraconsistent logic QC. As such QC logic can be seen as an analytical tool for evaluating data.

Keywords: Paraconsistency, non-classical logics, information theory

1 Introduction

Logic has long been recognized as the study of reasoning – reasoning not in the psychological sense of how people actually reason or what inferences people tend to draw given some initial assumptions, but reasoning in the sense of providing some standards for evaluating reasoning patterns and distinguishing good ones from bad ones. The development of logic in the past has concentrated on both the proof theoretic and model theoretic aspects of logic. Yet the *pragmatics* aspect of logic seems not to have received the same attention. In this paper we would like to demonstrate that a logic can be *practical* in the sense that it can assist us in evaluating and measuring the *amount* of information in an inconsistent set of data. Though we envision that any intelligent practical reasoning system must have some mechanism for handling inconsistencies, our goal here is not to address the issue of what is reasonable to conclude given some inconsistent data. Indeed there seems to be no *a priori* reason to favor any one particular inconsistency tolerant system. Rather we would like to illustrate how a particular paraconsistent logic can be used as a tool for analyzing inconsistent information. In particular, we would like to be able to quantitatively compare the relative information value of different sets of inconsistent data.

In the second section we'll motive the project with an example and draw some methodological points from the example. In the third section, we'll introduce the paraconsistent logic QC and present its proof theory. In the fourth section we'll

 ©Oxford University Press

introduce some definitions of information based on Shannon's communication theory. In the fifth section we'll show how QC logic can provide a foundation for a definition of information for inconsistent sets. In the sixth section we'll show how the definition of information can be applied to analyze over-constrained problems.

2 The Role of Logic in Reasoning

Consider a simple situation in which an object O may be located in one out of nine possible locations represented by a 3×3 grid. The location of O is encoded in a simple propositional language with p's representing the rows and q's representing the columns (see figure 1). Information is gathered from various sensors or sources about the location of O:

	q_1	q_2	q_3
p_1	$\times$		$\times$
p_2			
p_3			

figure 1.

Suppose we receive two messages:

$$A : p_1 \qquad B : \neg q_2$$

From the received messages we can conclude that

$$C : p_1 \wedge (q_1 \vee q_3)$$

Our example highlights three important methodological points. The first is the obvious point that information can be encapsulated in a formal language. The practical corollary of this is that more expressive formal languages are required for more demanding representational tasks. But more importantly, since a more expressive language may involve a greater computational cost, the choice of language should be gauged in terms of the representational task at issue. In our example it is clear that a simple propositional language suffices for the representational task.

The second point is that contextual information is often crucial to a reasoning task. In our example, the background information is that the object O is located in exactly one and no more then one location, and that there are exactly nine possible locations of O. It is only in the context of this background information that we can deduce C from A and B. More importantly, background information is not always explicitly stated in a given situation.

Finally, our example illustrates how the process of reasoning can be viewed as exploration in the space of possibilities – eliminating some and further exploring others. Each of A and B can be viewed as a restriction on the space of possibilities. Furthermore, information is compositional in the sense that the aggregate of A and B is simply the aggregate of their restrictions. The conclusion C is simply what is possible relative to the restrictions imposed by A and B together with the background information.

In the subsequent discussion we'll assume the simplest logical language. We assume that Φ is a set of propositional formulae generated from propositional atoms or variables $\{p_1, q_1, p_2, q_2, \ldots\}$, with the usual connectives, $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$. We use A, B, C, $\ldots$, to denote formulae, $\top$ for any tautology, $\bot$ for any contradiction, Γ, Σ, Δ, $\ldots$, to denote sets of formulae. We assume the equivalence between $A \rightarrow B$ and $\neg A \vee B$. We use $\vdash$ to denote the classical provability relation and $Cn(\Gamma)$ to denote $\{A \in \Phi : \Gamma \vdash A\}$. A set of formulae Γ is inconsistent if $\Gamma \vdash \bot$, otherwise Γ is consistent.

3 Paraconsistent Logics

A recalcitrant problem in the development of practical reasoning systems is the issue of uncertainty. One sort of uncertainty is the result of *underdetermination* of data. Another sort is the result of *overdetermination* of data. When information gathered from different sources is either incomplete or inconsistent, it is difficult to draw reliable conclusions to guide further action. More importantly, when inconsistencies arise a reasoner must take measures to guard against drawing trivial conclusions. Revising one's data to restore consistency may be an option available, but on occasions it is more important to maintain the integrity of the original data – perhaps the inconsistent data is irrelevant to one's overall project. On other occasions it may even be 'desirable' to have inconsistencies in one's database; for instance, inconsistencies may be deployed as directives to guide learning, and inconsistencies in a taxpayer's records can be used as a reason to prompt further investigation (see [9] for more discussion). The important point is that many ordinary circumstances require us to reason in the presence of inconsistencies. The main motivation for paraconsistent logics is precisely to develop reasoning systems that can tolerate inconsistencies. In classical logic, the rule *ex falso quodlibet* is admissible:

$$\frac{A \qquad \neg A}{B}$$

The practical implication of this is that classical logic is unsuitable as a practical reasoning system – it provides no guidance on what can be concluded when inconsistent information is presented, any formula can be derived from an inconsistent set of assumptions. In paraconsistent logics however *ex falso quodlibet* is no longer admissible. But as a result paraconsistent logics are also weaker then classical logic. In C. I. Lewis's original proof of *ex falso quodlibet* [18], various classical rules are deployed and thus various strategies are open to weaken classical logic:

(1)	$A \wedge \neg A$	Assumption
(2)	A	1, $\wedge$-E
(3)	$\neg A$	1, $\wedge$-E
(4)	$A \vee B$	2, $\vee$-I
(5)	B	3,4 $\vee$-E

Ignoring for now the difference between $\{A \wedge \neg A\}$ and $\{A, \neg A\}$, it is clear that we can block the derivation by blocking any one of the rules in line (2), (3), (4) or (5).

Relevant logicians, for instance, opt for a solution by blocking the use of $\lor$-E, also known as disjunctive syllogism (see [2, 3]); logicians favoring analytic implication opt for blocking the use of $\lor$-I, also known as the rule of addition (see [8, 23]); connexive logicians opt for blocking the use of $\land$-E instead. [1] Yet another novel approach is to restrict the ordering in which the rules are applied. Clearly Lewis's derivation requires that $\lor$-I be used before the use of $\lor$-E. So we can impose restrictions on both $\lor$-I and $\lor$-E so that they cannot be used in that specific combination. The resulting logic is called Quasi-classical logic (QC logic) by Besnard and Hunter in [5] and Hunter in [14]. Indeed a very simple way to characterize QC logic is this: rules in classical logic are divided into *composition* rules and *decomposition* rules; a derivation in QC logic proceeds by first applying decomposition rules and then applying composition rules, but not vice versa.[2]

One of the main advantages of QC logic is that all connectives are interpreted classically as *boolean* connectives. The composition and decomposition rules are divided roughly along the line of introduction and elimination rules associated with these connectives. Thus we have not changed any of the meanings of $\neg$, $\land$ or $\lor$. To simplify matters we take $\neg$, $\land$ and $\lor$ to be the primitive connectives and assume that $\land$ and $\lor$ are both commutative and associative and satisfy the contraction rules: $\frac{A\lor A}{A}$ $\frac{A\land A}{A}$. The following are the decomposition rules:

$\land$–Elimination	$\frac{A\land B}{A}$		
$\neg$–Elimination	$\frac{\neg\neg A\lor B}{A\lor B}$	$\frac{\neg\neg A}{A}$	
Resolution	$\frac{A\lor B \qquad \neg A\lor C}{B\lor C}$	$\frac{A\lor B \qquad \neg A}{B}$	$\frac{A \qquad \neg A\lor B}{B}$
D-Distribution	$\frac{A\lor(B\land C)}{(A\lor B)\land(A\lor C)}$	$\frac{(A\land B)\lor(A\land C)}{A\land(B\lor C)}$	
D-de Morgan	$\frac{\neg(A\land B)\lor C}{\neg A\lor\neg B\lor C}$	$\frac{\neg(A\lor B)\lor C}{(\neg A\land\neg B)\lor C}$	
	$\frac{\neg(A\land B)}{\neg A\lor\neg B}$	$\frac{\neg(A\lor B)}{\neg A\land\neg B}$	

The following are the composition rules:

$\land$–Introduction	$\frac{A \qquad B}{A\land B}$			
$\neg$–Introduction	$\frac{A\lor B}{\neg\neg A\lor B}$	$\frac{A}{\neg\neg A}$		
$\lor$–Introduction	$\frac{A}{A\lor B}$			
C-Distribution	$\frac{(A\lor B)\land(A\lor C)}{A\lor(B\land C)}$	$\frac{A\land(B\lor C)}{(A\land B)\lor(A\land C)}$		
C-de Morgan	$\frac{\neg A\lor\neg B\lor C}{\neg(A\land B)\lor C}$	$\frac{(\neg A\land\neg B)\lor C}{\neg(A\lor B)\lor C}$	$\frac{\neg A\lor\neg B}{\neg(A\land B)}$	$\frac{\neg A\land\neg B}{\neg(A\lor B)}$

---

[1] See [22] for a detailed discussion of these positions.

[2] The reader may notice that this is similar to *normalization proofs* of intuitionistic logic. The idea of a *normal proof* is to weed out unnecessary *detours* in a proof by blocking the use of elimination rule for a connective c followed by the introduction rule for c. The concern for QC, however, is to achieve paraconsistency.

A few comments about the rules are in order. The composition rules are, for the most part, the reversal of the decomposition rules. $\neg$–Introduction, C-Distribution and C-de Morgan are the reversal of $\neg$–Elimination, D-Distribution and D-de Morgan respectively. Obviously all our rules are classically valid. But more importantly, all the rules except $\wedge$–Elimination and $\vee$–Introduction preserve exactly the classical models of their premises. By this we mean that any two-valued interpretation is a model of the premises iff it is also a model of the conclusion. In the case of $\wedge$–Elimination and $\vee$–Introduction however, the set of models for the premises is properly contained in the set of models for the conclusion, i.e. the conclusions of these rules are strictly weaker then their assumptions. Amongst all the decomposition and composition rules, $\vee$–Introduction is the only rule which allows the introduction of new propositional variables not contained in the premise.

Also note that the set of decomposition rules is sufficient to reduce any formula to its Conjunctive Normal Form (CNF) and thus to an equivalent set of clauses (disjunctions of literals). We can further obtain the *resolvents* from these clauses via the use of the resolution rule. Normally the use of the resolution rule in automated theorem proving aims at deriving the empty clause. But in our case, the role of resolution is to decompose clauses into literals so that we can identify and isolate all the inconsistencies in the assumptions.

Officially a derivation in QC logic takes a set of formulae Γ as assumptions and a formula A as a conclusion. We write $\Gamma \vdash_{QC} A$ to denote that there is a QC derivation of A from Γ. The derivation proceeds first by the construction of a series of decomposition trees via the decomposition rules. The leaves of these decomposition trees are simply members of Γ; nodes are formulae obtained via the application of the decomposition rules, and finally their roots are either clauses or resolvents of clauses obtained by application of the resolution rule. The roots of the decomposition trees are then used, as leaves, to construct a composition tree via the composition rules. The composition tree terminates if the root is a formula in CNF which is classically equivalent to the conclusion A.

Example 3.1 For $\Gamma = \{p \vee q,\ p \vee \neg q,\ \neg p \wedge r\}$

$$\Gamma \vdash_{QC} p \qquad \Gamma \vdash_{QC} \neg p$$

$$\Gamma \vdash_{QC} q \qquad \Gamma \vdash_{QC} \neg q$$

$$\Gamma \vdash_{QC} r \qquad \Gamma \nvdash_{QC} \neg r$$

In our example there is a clear sense in which the variable r is not involved in any inconsistency though it is conjoined with $\neg p$ which is a culprit. One of the key features of QC logic is its ability to identify literals that are involved in an inconsistency. Other paraconsistent logics such as Belnap's *First Degree Entailment* [4] or da Costa's C_ω [7] lack this feature since they lack the resolution rule.

It is also easy to see that QC logic has no theorems, i.e. no formula is derivable from the empty set of assumptions. Moreover, the derivability relation $\vdash_{QC}$ does not satisfy transitivity (also known as the *cut* rule). Surprisingly though, $\vdash_{QC}$ is monotonic (see Hunter [14] for details). But like classical logic QC logic is decidable: there is a simple decision procedure to determine whether a formula A is QC derivable from a finite set of assumptions Γ.

Definition 3.2 The decomposition closure of a set Γ, denoted by $C_D(\Gamma)$ is the least superset of Γ that is closed under the decomposition rules (including the contraction rules for $\wedge$ and $\vee$).

We note two important facts about C_D: for any Γ the set of propositional variables occurring in Γ is exactly the set of propositional variables occurring in $C_D(\Gamma)$. Moreover, if Γ is finite, then $C_D(\Gamma)$ is also finite. We may say that C_D is a variable and finiteness preserving closure operator.

We say that a CNF of a formula A is reduced if it is a minimal CNF such that all of its propositional variables are variables occurring in A. We say that a reduced CNF of a formula A respects $C_D(\Gamma)$ if all of its clauses can be composed from members of $C_D(\Gamma)$ via the composition rules. Now to determine whether A is QC derivable from a finite Γ is simply a matter of finding a reduced CNF of A that respects $C_D(\Gamma)$. Though there is no unique reduced CNF for a formula A, it is easy to see that one of them would respect $C_D(\Gamma)$ iff all of them would. Since $C_D(\Gamma)$ is finiteness preserving, $\wp(C_D(\Gamma))$ must be finite given that Γ is finite. Hence there are only finitely many ways to generate composition trees from $C_D(\Gamma)$. The checking must terminate eventually.

4 Information Measurement

An old idea about information is that there is an inverse relation between information and possibility. In Shannon's communication theory this relation is expressed by the equation,[3]

$$I(A) \quad = \quad -\log P(A) \tag{4.1}$$

In (1), $I(A)$ is the *amount of information* conveyed by A (or the information value of A) and $P(A)$ is the probability of A occurring. Not surprisingly, the thrust of the idea is that information eliminates possibilities – the more unlikely that A occurs the more informative it is to assert A. To illustrate, consider our example in section 2. Recall that O is located in one out of nine possible locations represented by a 3×3 grid:

	q_1	q_2	q_3
p_1			
p_2			
p_3			

figure 2.

The set of all possible locations of O can be regarded as a probability space. Furthermore, we may assume that all possible location are equiprobable. Using (1), we can calculate the information of values $A = p_1$ and $A' = p_1 \wedge \neg q_2$:

$$I(p_1) = -\log\frac{3}{9} = 0.48 \qquad\qquad I(p_1 \wedge \neg q_2) = -\log\frac{2}{9} = 0.65$$

Not surprisingly, we have $I(A) < I(A')$. Even at an intuitive level it is clear that A' is more informative since A' provides the additional information that $\neg q_2$.

[3] See [17] chapter 2-3 for an overview. For a related approach to *semantic information theory* see Hintikka [11, 12, 13].

4.1 Inconsistent Information

Data, encoded as formulae in a logical language, are representations of the state of the world. For a consistent set of data each classical interpretation of the data can be regarded as a possible state of the world. Since a consistent set of formulae in finitely many propositional variables has only finitely many non-equivalent interpretations, we can treat the collection of all possible non-equivalent interpretations as a probability space and assign equal probability to each interpretation. Naturally this leads to a definition of information analogous to equation (1).

Definition 4.1 (Lozinskii [19]) Let Γ be a consistent set of formulae in n variables and let $\mathfrak{M}(\Gamma)$ denotes the collection of (equivalence classes of) models for Γ. The information value of Γ is defined by the following equation:

$$I(\Gamma) \quad = \quad \log \frac{2^n}{|\mathfrak{M}(\Gamma)|} \tag{4.2}$$

Rewriting equation (2) in base 2 we have:

$$I(\Gamma) \quad = \quad n - \log_2 |\mathfrak{M}(\Gamma)| \tag{4.3}$$

The intuitive justification of our definition is that the amount of information in a data set should be based on the logarithmic ratio between the number of non-equivalent interpretations and the number of equivalent models of the data. This is generally in agreement with the underlying idea of equation (1). If a data set allows us to exclude all interpretations except one as its model, then the data set has maximum information value. We also note that the definition applies only to data sets in finitely many variables. For sets in infinitely many variables we need to modify our definition since it is not meaningful to talk about the ratio between two infinite cardinals. For simplicity, we'll focus on sets in finitely many variables.

We should mention that Lozinskii's definition of information value is similar to the κ function defined by Gent, Prosser and Walsh [10] in their study of the *constrainedness* of combinatorial search problems. While Lozinskii's definition is defined for *single problems*, the κ function is intended to provide a quantitative measurement, for *an ensemble of search problems* (e.g. **SAT** or graph colouring problems), to determine how hard or easy it is to find solutions for these problems. One of the main interests in κ is to employ it to develop heuristics to guide search. Another interest in κ is to use it to provide a unified account of the *phase transition* phenomena in search problems (see [10] for more details). The κ function of Gent, Prosser and Walsh amounts roughly, in Lozinskii's term, to the average value of $I(\Gamma)/n$ for an ensemble of Γ's:

$$\kappa \quad = \quad 1 - \frac{\log_2 \langle sol \rangle}{n} \tag{4.4}$$

In equation (4.4), $\langle sol \rangle$ is the number of expected solutions to the problem in the ensemble, and n is the logarithm (base 2) of the size of the state space.

In the context of inconsistent data it is natural to ask for a measurement of information analogous to definition 4.1. However, unlike the approach of Aisbett and Gibbon in [1], we do not agree that inconsistent data provides no information at all. What is and what isn't informative seems to depend, at least partly, on the goal of the agent in possession of the data. For a tax auditor, inconsistencies in a taxpayer's records are useful information for detecting possible fraud. Inconsistencies may also be useful in cases where they are deployed as directives to guide learning or as indicators for faulty components in a complex system. Worse still, by assigning null information value to all inconsistent data we may incur information loss. As we mentioned earlier, an important aspect of handling inconsistencies is the ability to compare and evaluate the relative merit of different inconsistent data sets. We need to have some quantitative criteria to determine whether one data set is more inconsistent or informative then another. Thus it is desirable to have a general theoretical framework for measuring both consistent and inconsistent information. In [20] Lozinskii provides such a framework.

Definition 4.2 A subset Δ of a set Γ is maximally consistent if:(1) $\Delta \nvdash \bot$ and (2) for any $A \in \Gamma - \Delta$, $\Delta \cup \{A\} \vdash \bot$.

Definition 4.3 (Lozinskii [20]) Let Γ be a set of formulae in n variables and $M(\Gamma)$ be the set of maximal consistent subsets of Γ. For each $\Delta \in M(\Gamma)$, if $\mathfrak{M}(\Delta)$ is the collection of (equivalence classes of) models of Δ then the collection of quasi-models is defined by:

$$\mathfrak{U}(\Gamma) \quad = \quad \bigcup\{\mathfrak{M}(\Delta) : \ \Delta \in M(\Gamma)\} \tag{4.5}$$

The information value of Γ is defined by the following equation:

$$I(\Gamma) \quad = \quad n - \log_2 |\mathfrak{U}(\Gamma)| \tag{4.6}$$

Intuitively, a quasi-model represents a possible state of the world or outcome according to some maximal consistent subset of Γ. The main idea behind definition 4.3 is that the information value of a set of formulae is determined by the logarithmic ratio between the number of non-equivalent interpretations and the number of quasi-models. Clearly definition 4.3 agrees with definition 4.1 when Γ is consistent and yields a defined value for $I(\Gamma)$ when $M(\Gamma)$ is non-empty. We note that according to the new definition the information value of a data set is monotonically increasing with respect to consistent supersets, i.e. for any consistent $\Gamma' \supseteq \Gamma$, $I(\Gamma) \leq I(\Gamma')$. For inconsistent sets however, the information value is nonmonotonic when there is an increase in inconsistencies. For instance,

Example 4.4 For $\Delta = \{p \vee q,\ p \vee \neg q,\ \neg p \wedge r\}$ $\Gamma = \Delta \cup \{\neg r\}$ and $\Gamma' = \Delta \cup \{s\}$

$$I(\Gamma) < I(\Delta) \qquad\qquad I(\Gamma') > I(\Delta)$$

5 QC Logic and Information Measure

Lozinskii's new definition is problematic in two respects. The first is that the presence of tautologies will affect the value of $I(\Gamma)$. Since we are primarily interested in the amount of *empirical* information about the world, it seems reasonable to disregard tautological statements in a data set. In a more general setting, of course, we may relativize the information value of a data set by nominating a particular set of formulae to be disregarded. This is a useful generalization since, as we have already pointed out, the information value of a data set is at least partly dependent on the agent in possession of the data. Perhaps an agent has already independently confirmed A and thus it is not informative to be told A again. The second problem is that $I(\Gamma)$ is too sensitive to the syntax of the formulae in Γ and thus may produce counter-intuitive consequences. Indeed this is a general problem with any inconsistency tolerant mechanism based on maximal consistent subsets. The syntactic features of the formulae in the set determine how the set can be fragmented into consistent subsets. In [24], Wong gives the following example:

Example 5.1

	$\Gamma_1 = \{p \wedge q,\ \neg p \wedge r\}$	$\Gamma_2 = \{p \wedge q \wedge r,\ \neg p \wedge q \wedge r,\ p \wedge \neg q \wedge r\}$
$\lvert M(\Gamma)\rvert$	2	3
$\lvert\mathfrak{U}(\Gamma)\rvert$	4	3
$I(\Gamma)$	1.00	1.42

figure 3.

In this example, Γ_2 is in some sense more inconsistent than Γ_1; yet we have $I(\Gamma_2) > I(\Gamma_1)$. We should be able find a more principled way to describe the relationship between the information value and the amount of inconsistencies of a set. For an agent whose aim is to avoid inconsistencies, the information value of a highly inconsistent data set should be lower than that of a set with fewer inconsistencies. Dually, for an inconsistency seekers, e.g. an auditor, the information value of a highly inconsistent data set should be higher than that of a set with fewer inconsistencies. This suggests that we should treat the information value of a set as varying inversely to the amount of inconsistency in the set. A natural solution is to relativize the information value of a set using the decomposition closure defined in previous section; that is, we let

$$I^*(\Gamma) \quad = \quad n - \log_2 |\mathfrak{U}(C_D(\Gamma))| \tag{5.1}$$

Since C_D is a variable and finiteness preserving closure operator, replacing Γ with $C_D(\Gamma)$ in equation (4.6) has no effect on the value n. Indeed the advantage of equation (5.1) over equation (4.6) is that it provides a more discriminating way of evaluating the information value of a data set. This gives us a more realistic appraisal of the usefulness of our data. The information value of a set no longer depends on how the formulae are syntactically presented.

Example 5.2 $\Gamma = \{p \vee q,\ p \vee \neg q,\ \neg p \wedge r\}$ and $\Gamma' = \{p \vee q,\ p \vee \neg q,\ \neg p,\ r\}$

Using equation (4.6) we have $I(\Gamma) \neq I(\Gamma')$. According to equation (5.1) however we have $I^*(\Gamma) = I^*(\Gamma')$. In the extreme case when $p_i \in C_D(\Gamma)$ and $\neg p_i \in C_D(\Gamma)$ for every variable p_i occurring in Γ, we have $I^*(\Gamma) = 0$ since the number of quasi-models for Γ is exactly 2^n where n is the number of distinct variables in Γ. In one sense C_D gives us a syntactic normal form for a set of formulae. Looking at our previous example, it is easy to see that I^* provides a more appropriate information value for Γ_1 and Γ_2.

	$C_D(\Gamma_1)$	$C_D(\Gamma_2)$
$\lvert M(C_D(\Gamma))\rvert$	2	4
$\lvert\mathfrak{U}(C_D(\Gamma))\rvert$	2	4
$I^*(\Gamma)$	2.00	1.00

figure 4.

We note that we have not make full use of QC logic here. Indeed this is unnecessary and undesirable since the composition rule ∨–I allows the introduction of arbitrary new propositional variables. Clearly the introduction of new variables would interfere with the information value of a data set. In addition, we have also considered using C_D in conjunction with inference mechanisms based on maximal consistent subsets [25]. The idea there is similar in that we can first apply C_D to obtain a normal form for an inconsistent set and then use further inference mechanisms to extract conclusions from the set.

6 Application

In previous works Hunter and Nuseibeh [15, 16] have illustrated the usefulness of QC logic in the analysis of inconsistent specifications in software engineering. Hunter and Nuseibeh have pointed out that inconsistent specifications are often unavoidable during software development. They argued persuasively that during the software development cycle it is often more important to manage inconsistencies intelligently, i.e., we need to analyze and to keep track of inconsistencies rather then resolving them immediately. In the same spirit we advocate using QC logic and the definition given by equation (5.1) as a basis to analyze over-constrained problems.

6.1 Constraint Satisfaction Problems

A constraint satisfaction problem (**CSP**) involves,

1. a set of variables, $X_1, \ldots, X_n$
2. associated with each variable, X_i, is a domain, D_i of values
3. a set of constraints, $C_1, \ldots, C_m$, each defined on subset of variables a subset of the Cartesian product of the associated domains, i.e.

$$C_i(X_{i_1}, \ldots, X_{i_k}) \subseteq (D_{i_1} \times \ldots \times D_{i_k})$$

A *solution* to a **CSP** is simply an assignment of values to variables such that all constraints are satisfied. A **CSP** is a Finite Constraint Satisfaction Problem (**FCSP**) if its constraint domains are finite. Many real world problems such as optimization problems or job scheduling problems can be viewed as **CSP**s.

As is well known, there is a close relationship between **FCSP**s and logic (see [6, 21]). Any **FCSP** can be stated as an equivalent logic problem in a variety of settings. In the model checking approach for instance, a **FCSP** is taken to have a solution iff a certain propositional theory Γ is satisfiable. In fact, the solutions are just the set of models of Γ. In this scheme, the theory Γ is constructed as a set of propositional formulae in CNF such that

1. Each possible combination of values for variables is represented by a set of propositional variables, $p^{x_1}_{d_1}, \ldots, p^{x_m}_{d_n}, \ldots$, where intuitively, $p^{x_i}_{d_j}$ is the proposition which says that the variable x_i is instantiated by the value d_j. For instance, the sentence $(p^{x_i}_{d_j} \vee p^{x_i}_{d_k})$ says that the variable x_i is instantiated by at least one of values d_j and d_k.
2. A constraint is stated *negatively* in terms of values that are forbidden, e.g. the sentence $\neg p^{x_i}_{d_j}$ says that x_i are never instantiated to value d_i, the sentence $\neg p^{x_i}_{d_k} \vee \neg p^{x_j}_{d_k}$ says that x_i and x_j are never instantiated to the same value d_k. The set of all constraints is represented by a set of propositional formulae in the variables, $p^{x_1}_{d_1}, \ldots, p^{x_m}_{d_n}, \ldots$.

6.2 Over-constrained Problems

As it is with many real world problems, a **CSP** can be without a solution. A solutionless **CSP** is an over-constrained problem (**OCP**) – every assignment of values to variables fails to satisfy at least one constraint. Consider for instance,

Example 6.1 Let X, Y, Z be variables whose domain is $\{1, 2, 3\}$. Let the constraints be: $X < Y$, $Y < Z$ and $Z < X$.

Clearly, this is an **OCP** since no natural numbers can satisfy all three constraints. This example illustrates that there are two main factors which contribute to a problem being over-constrained – and thus provides two different approaches to resolving **OCP**s. The first is the domain of possible values and the second is the constraints themselves. In our example if we were to add a value w to the domain such that for some m and n, $m < n$, $n < w$, and $w < m$, then all constraints would be satisfied (w need not be a natural number), in which case we no longer have an over-constrained problem. Alternatively, we may accept a certain *partial* assignment that satisfies some but not all of the constraints as a solution. Typically, we may accept those assignments that satisfy a maximal number of constraints or variables. Given that any **FCSP** is equivalent to a model checking problem in propositional logic, the second approach to solving a finite **OCP** is equivalent to finding models for a certain subset of an inconsistent set of propositions.

Regardless of how we may resolve an **OCP**, it is sometime desirable to analyze the problem first before any further action is taken. In this respect, it is clear that QC logic is well suited to the task. According to our previous scheme, we can encode a finite **OCP** as a propositional theory Γ; Γ must be unsatisfiable and thus inconsistent. We can then apply QC logic to analyze the information value of Γ. In particular in an **OCP** not all variables may be involved in an inconsistency (i.e. being overly constrained). Thus it is desirable to identify those variables that are involved in an inconsistency. The strategy, as before, is to take the decomposition closure of Γ and then measure the value $I^*(\Gamma)$. In a highly over-constrained problem we should expect to see a lower value for $I^*(\Gamma)$ and vice versa. This gives us a relative measurement of the constrainedness or information value of **OCP**s.

We should point out that I^* is based essentially on the number, i, of inconsistent pair of literals in $C_D(\Gamma)$, since $|\mathfrak{U}(C_D(\Gamma))| \geq 2^i$. Clearly there are other alternatives. For instance, we may consider an information measurement based on the average size of proofs of an inconsistent pair of literals using the decomposition rules of QC.

Presumably this sort of measurement will give us an indication of the degree of difficulty in detecting inconsistencies in a data set. The higher the average size of these proofs, the more computation is required to identify the inconsistencies. In addition, we should also expect other parameters, e.g. the size of Γ and the length of formulae in Γ, to play a role in determining the size of these proofs.

7 Conclusion

In this paper we have argued that there are general advantages in developing practical reasoning systems that can tolerate inconsistencies. In this respect we have considered a paraconsistent logic that can avoid drawing trivial conclusions in the presence of inconsistencies. But more importantly we advocate the use of paraconsistent logic in assisting us in analyzing inconsistent data. In this light, the role of logic goes beyond capturing valid form of inferences. Logic can be seen as a tool for analysis.

References

[1] J. Aisbett and G. Gibbon. A Practical Measure of the Information in a Logic Theory. *Journal of Experiment and Theoretical Artificial Intelligence*, 11:201–217, 1999.

[2] A. E. Anderson and N. D. Belnap. *Entailment: The Logic of Relevance and Necessity Vol 1.* Princeton University Press, 1975.

[3] A. E. Anderson, N. D Belnap, and J. M. Dunn. *Entailment: The Logic of Relevance and Necessity Vol 2.* Princeton University Press, 1992.

[4] N. D. Belnap. A Useful Four-Valued Logic. In J. M. Dunn and G. Epstein, editors, *Modern Uses of Multiple-Valued Logic*, pages 8–37. D. Reidel Pub., 1975.

[5] P. Besnard and A. Hunter. Quasi-classical Logic: Non-trivializable Classical Reasoning from Inconsistent Information. In *Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, Lecture Notes in Artificial Intelligence 946, pages 44–51. Springer Verlag, 1995.

[6] W. Bibel. Constraint Satisfaction from a Deductive Viewpoint. *Artificial Intelligence*, 35:401–413, 1988.

[7] N. C. A. Da Costa. On the Theory of Inconsistent Formal System. *Notre Dame Journal of Formal Logic*, 15 (4):497–510, 1974.

[8] J. M. Dunn. A Modification of Parry's Analytic Implication. *Notre Dame Journal of Formal Logic*, 13 (2):195–205, 1972.

[9] D. M. Gabbay and A. Hunter. Making Inconsistency Respectable: a Logical Framework for Inconsistency in Reasoning, Part I - a Position Paper. *Fundamentals of Artificial Intelligence Research*, pages 19–32, 1991.

[10] I. P. Gent, P. Prosser, and T. Walsh. The Constrainedness of Search. under review for Journal of the ACM, 1999.

[11] J. Hintikka. Information, Deduction, and the a Priori. *Nous*, 4 (2):135–152, 1970.

[12] J. Hintikka. On Semantic Information. In J. Hintikka and P. Suppes, editors, *Information and Inference*, pages 3–27. D. Reidel Pub., 1970.

[13] J. Hintikka. Surface Information and Depth Information. In J. Hintikka and P. Suppes, editors, *Information and Inference*, pages 263–297. D. Reidel Pub., 1970.

[14] A. Hunter. Reasoning with Conflicting Information Using Quasi-classical Logic. to appear Journal of Logic and Computation, 1999.

[15] A. Hunter and B. Nuseibeh. Analysing Inconsistent Specifications. In *Proceedings of the Third IEEE International Symposium on Requirements Engineering (RE'97)*, pages 78–86. IEEE Computer Society Press, 1997.

[16] A. Hunter and B. Nuseibeh. Managing Inconsistent Specifications: Reasoning, Analysis and Action. *ACM Transactions on Software Engineering and Methodology*, 7 (4):335–367, 1998.

[17] G. Jumarie. *Relative Information: Theories and Applications*. Springer-Verlag, 1990.

[18] C. I. Lewis and C. H. Langford. *Symbolic Logic*. New York, 1932.

[19] E. L. Lozinskii. Information and Evidence in Logic Systems. *Journal of Experiment and Theoretical Artificial Intelligence*, 6:163–193, 1994.

[20] E. L. Lozinskii. Resolving Contradictions: a Plausible Semantics for Inconsistent Systems. *Journal of Automated Reasoning*, 12:1–31, 1994.

[21] A. K. Mackworth. The Logic of Constraint Satisfaction. *Artificial Intelligence*, 58:3–20, 1992.

[22] R. Routley, V. Plumwood, R. K. Meyer, and R. T. Brady. *Relevant Logics and Their Rivals Volume 1*. Ridgeview Publishing Co., 1982.

[23] A. Urquhart. A Semantic Theory of Analytical Implication. *Journal of Philosophical Logic*, 2:212–219, 1973.

[24] P. Wong. Paraconsistent Inference and Preservation. Workshop on Logic in Computing Science, University of Technology, Sydney, 1998.

[25] P. Wong. Reasoning with Inconsistent Information. PhD in progress, Automated Reasoning Group, Computer Science Laboratory, RSISE, Australian National University, 2000.

Received September, 2000