Reasoning with Contradictory Information using Quasi-classical Logic

Anthony Hunter
Department of Computer Science
University College London
London WC1E 6BT, UK
Email: a.hunter@cs.ucl.ac.uk

July 20, 1999

Abstract

The proof theory of quasi-classical logic (QC logic) allows the derivation of non-trivializable classical inferences from inconsistent information. A non-trivializable, or paraconsistent, logic is, by necessity, a compromise, or weakening, of classical logic. The compromises on QC logic seem to be more appropriate than other paraconsistent logics for applications in computing. In particular, the connectives behave in a classical manner. Here we motivate the need for QC logic, present a proof theory, and semantics for the logic, and compare it to other paraconsistent logics.

1 Introduction

Intellectual activities usually involve reasoning with different perspectives. For example, consider negotiation, learning, or merging multiple opinions. Central to reasoning with different perspectives is the issue of handling inconsistencies. Maintaining absolute consistency is not always possible. Often it is not even desirable since this can unnecessarily constrain the intellectual activity, and can lead to the loss of important information. Indeed since the real world forces us to work with inconsistencies, we should formalize some of the usually informal or extra-logical ways of responding to them. This is not necessarily done by eradicating inconsistencies, but rather by supplying logical rules specifying how we should act on them [GH91, GH93]. In this way, we are moving away from a classical view of information being either true or false, to a view where we accept that we may have a number of perspectives on information and that these perspectives may contradict each other.

An important example of the need to reason with inconsistent information is in systems development. The development of most large and complex systems necessarily involves many people, each with their own perspectives on the system. Systems development therefore involves problems of identifying and handling inconsistencies between such perspectives. For this there is a need to tolerate inconsistencies, and more importantly to be able to act in a context-dependent way in response to inconsistency [FGH+94]. This is vital since inflexible forcing of consistency can unnecessarily constrain the development process.

Paraconsistent reasoning is important in handling inconsistent information [BdCGH97], and there have been a number of proposals for paraconsistent logics (for reviews see [EGH95, CH97, Hun98]). 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. However this results in useful proof rules such as disjunctive syllogism failing, and intuitive equivalences such as $\neg\alpha \vee \beta \equiv \alpha \rightarrow \beta$ failing. For users of logic, such as software engineers, the migration from classical logic which might be familiar and intuitive, to a paraconsistent logic where classical connectives are weakened, could be difficult.

An alternative, called quasi-classical logic (or QC logic), which we first introduced in [BH95], is to restrict the proof theory. In this restriction, compositional proof rules (for example, disjunction introduction) cannot be followed by decompositional rules (for example, resolution). Whilst this gives a logic that is weaker than classical logic, it does mean that the connectives behave classically. Furthermore, we argue that all the inferences from the logic are intuitive. We believe the logic is appealing for reasoning with inconsistencies arising in applications such as systems development [HN98, HN97], and more generally in information fusion [CH97].

QC logic has been shown to have potential in requirements engineering as it offers appropriate reasoning in the presence of inconsistency, supporting continued requirements gathering, and it provides the basis of analytical techniques for localizing and categorizing inconsistencies [HN97, HN98].

The purpose of this paper is to develop the presentation and analysis of QC logic. In the next section, we provide some basic definitions, and then in the following section, we discuss the essential idea behind QC logic. In subsequent sections, we provide a proof theory, semantics, and a comparison with other key paraconsistent logics.

2 Some basic definitions

In this section, we establish some basic definitions for discussing the language and proof theory of QC logic.

Definition 2.1 *Let $\mathcal{L}$ be the set of classical propositional formulae formed from a non-empty countable set of atoms $\mathcal{A}$, and the $\wedge, \vee, \rightarrow$ and $\neg$ connectives. The language of QC logic is $\mathcal{L}$. We restrict formulae to those containing a finite number of occurrences of atoms.*

Example 2.1 *From the propositional atoms α, β and γ, members of $\mathcal{L}$ include $\alpha, \beta \wedge \gamma$, $\neg\alpha \wedge \alpha$ and $(\alpha \wedge \beta) \rightarrow \neg\neg\gamma$.*

Definition 2.2 *For each atom $\alpha \in \mathcal{L}$, α is a literal and $\neg\alpha$ is a literal. These are the only literals. For $\alpha_1 \vee .. \vee \alpha_n \in \mathcal{L}$, $\alpha_1 \vee .. \vee \alpha_n$ is a clause iff each of $\alpha_1, .., \alpha_n$ is a literal. For $\alpha_1 \wedge .. \wedge \alpha_n \in \mathcal{L}$, $\alpha_1 \wedge .. \wedge \alpha_n$ is in a conjunctive normal form (CNF) iff each of $\alpha_1, .., \alpha_n$ is a clause. These are the only CNF formulae. Let $\mathcal{L}_{clauses} = \{\beta \mid \beta \in \mathcal{L}$ and β is a clause $\}$*

Definition 2.3 *For $\alpha \in \mathcal{L}$, and $\beta \in \mathcal{L}$, α is a conjunctive normal form (CNF) of β iff α is classically equivalent to β and α is in a CNF.*

For any $\alpha \in \mathcal{L}$, a CNF of α can be produced by the application of distributivity, arrow elimination, double negation elimination, and de Morgan laws.

Definition 2.4 *Let $\vdash$ denote the classical consequence relation.*

Definition 2.5 *Let $\alpha_1 \vee ... \vee \alpha_n$ be a clause, then $Literals(\alpha_1 \vee ... \vee \alpha_n)$ is the set of literals $\{\alpha_1, ..., \alpha_n\}$ that are in the clause.*

Definition 2.6 *Let α be a clause, and let β be a literal, such that $\beta \in Literals(\alpha)$. $Focus(\alpha, \beta)$ is just the original formula α without the disjunct β. For a literal α, let $Focus(\alpha, \alpha) = \bot$.*

Example 2.2 *Let $\alpha \vee \beta \vee \gamma$ be a clause then $Focus(\alpha \vee \beta \vee \gamma, \beta)$ is $\alpha \vee \gamma$.*

Definition 2.7 *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 complementation operation is not in the object language. We use it to make clearer definitions for the semantics of QC logic.*

We now consider issues of paraconsistency.

Definition 2.8 *A logic is defined as paraconsistent iff ex falso qoudlibet does not hold for the logic, where ex falso quodlibet can be defined as follows:*

$$\frac{\alpha, \quad \neg\alpha}{\beta}$$

Since this definition of paraconsistency is very general, we define a notion of trivial reasoning that will help us to characterize useful paraconsistent logics. For this, we require the following function.

Definition 2.9 *Let* $\alpha \in \mathcal{L}$ *and let* $\{\alpha_1, .., \alpha_n\} \in \wp(\mathcal{L})$. $Atoms(\{\alpha\})$ *is the set of atoms used in* α*, and* $Atoms(\{\alpha_1, .., \alpha_n\}) = Atoms(\{\alpha_1\}) \cup .. \cup Atoms(\{\alpha_n\})$.

Example 2.3 *Let* Δ *be* $\{(\alpha \vee \beta) \wedge \neg\gamma, \delta \rightarrow \neg\neg\gamma\}$. *Then* $Atoms(\Delta) = \{\alpha, \beta, \gamma, \delta\}$.

Definition 2.10 *Let* $\alpha \in \mathcal{L}$ *and* $\Delta \in \wp(\mathcal{L})$. α *is a trivial inference from* Δ *iff* α *is an inference from* Δ *and* $Atoms(\Delta) \cap Atoms(\{\alpha\}) = \emptyset$. α *is a non-trivial inference from* Δ *iff* α *is an inference from* Δ *and* $Atoms(\Delta) \cap Atoms(\{\alpha\}) \neq \emptyset$.

Example 2.4 *Let* $\alpha, \beta \in \mathcal{A}$ *such that* $\alpha \neq \beta$. *Let* Δ *be* $\{\alpha, \neg\alpha\}$*, and let* β *be an inference from* Δ*, then* β *is a trivial inference from* Δ.

So for any set of formulae Δ, and any inference α from Δ, if α has no propositional atom in common with Δ, then α is a trivial inference from Δ. By this definition, classical logic has trivial inferences even if $\Delta = \emptyset$, though in this case, these trivial inferences are tautologies.

3 The essential idea of QC logic

First, we use the language of QC logic to represent beliefs.

Definition 3.1 *For each* $\phi \in \mathcal{L}$, ϕ *is a* **belief**. *For each* $\phi \in \mathcal{L}$*, if* ϕ *is a literal, then* ϕ *is a* **literal belief**.

We now consider the essential idea behind QC logic. We describe it using the resolution proof rule. Resolution can be applied to clauses to generate further clauses called resolvents. For example, by resolution $\beta \vee \gamma$ is a resolvent of $\alpha \vee \beta$ and $\neg\alpha \vee \gamma$.

Given a set of clauses as assumptions, each clause in the assumptions, can be regarded as a belief, and each resolvent can be regarded as a belief. So for each clause α in the assumptions, there is a belief α derivable from the assumptions. Similarly, for each resolvent α derivable from the assumptions, there is a belief α derivable from the assumptions.

Resolution can be regarded as a process of focusing beliefs. So the clause γ is more focused than the clause $\neg\alpha \vee \gamma$. Similarly, for the pair of beliefs $\alpha \vee \beta$ and $\neg\alpha \vee \gamma$, the resolvent $\beta \vee \gamma$ is more focused. In general, a clause α is more focused than a clause β if $Atoms(\{\alpha\}) \subset Atoms(\{\beta\})$. Hence, as one or more applications of resolution decomposes a set of assumptions, it focuses the beliefs derivable from the assumptions.

A useful property of resolution is that α is a resolvent only if all the literals used in α are literals used in the set of assumptions (assuming, of course, that resolution is the only proof rule used). This means that any resolvent, and hence any belief derivable from the assumptions, is a non-trivial inference from the assumptions. This holds even if the set of assumptions is classically inconsistent. As a result, resolution can constitute the basis of useful paraconsistent reasoning.

QC logic is motivated by the need to handle beliefs rather than the need to address issues of verisimilitude for given propositions. We are aiming for a logic of beliefs in the "real world" rather than a logic of truths in the "real world". For this, we introduce the following definition for a model.

Definition 3.2 *Let $\mathcal{A}$ be the set of atoms in $\mathcal{L}$. Let $\mathcal{O}$ be the set of objects defined as follows, where $+\alpha$ is a positive object, and $-\alpha$ is a negative object.*

$$\mathcal{O} = \{+\alpha \mid \alpha \in \mathcal{A}\} \cup \{-\alpha \mid \alpha \in \mathcal{A}\}$$

We call any $X \in \wp(\mathcal{O})$ a model. So X can contain both $+\alpha$ and $-\alpha$ for some α. We assume that each object is distinct. In particular, we assume that no object is both positive and negative, and that for no α, β is $+\alpha = -\beta$.

We can consider the following meaning for positive and negative objects being in some model X.

Definition 3.3 *For each atom $\alpha \in \mathcal{L}$, and each $X \in \wp(\mathcal{O})$,*

$+\alpha \in X$ *means*	*that in X there is* **a reason for** *the belief α and that in X there is* **a reason against** *the belief $\neg\alpha$*
$-\alpha \in X$ *means*	*that in X there is* **a reason against** *the belief α and that in X there is* **a reason for** *the belief $\neg\alpha$*

So a model just contains reasons for/against beliefs — though we defer, for a page, discussion of formulae more complicated than literals. Note, the definition of a model incorporates no notion of truth or falsity. Note also, the notion of a reason for, or against, is binary. There is no way, in this framework, to say for example that there are n reasons for α. Also there is no way to denote the relative strengths of the reasons for/against any belief.

Equivalently, we can regard a model as giving a notion of satisfiability.

$+\alpha \in X$ means α is "satisfiable" in the model

$-\alpha \in X$ means $\neg\alpha$ is "satisfiable" in the model

$+\alpha \notin X$ means α is not "satisfiable" in the model

$-\alpha \notin X$ means $\neg\alpha$ is not "satisfiable" in the model

Since we can allow both an atom and its complement to be satisfiable, we have decoupled, at the level of the model, the link between a formula and its complement. In contrast, in a classical model, if a model satisfies a literal, then it is forced to not satisfy the complement of the literal.

Since we have decoupled the link between a formula and its negation at the level of the model, we have 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 the connectives given in [Bel77], but instead provide a significantly different semantics. First, we define strong satisfaction.

Definition 3.4 *Let $\models_s$ be a satisfiability relation, called strong satisfaction, such that $\models_s \subseteq \wp(\mathcal{O}) \times \mathcal{L}_{clauses}$. For $X \in \wp(\mathcal{O})$, we define $\models_s$ as follows, and $\alpha_1, ..., \alpha_n$ are literals in $\mathcal{L}$, where α is an atom in $\mathcal{L}$.*

$$X \models_s \alpha \textit{ if } +\alpha \in X$$

$$X \models_s \neg\alpha \textit{ if } -\alpha \in X$$

$$\begin{array}{l} X \models_s \alpha_1 \vee ... \vee \alpha_n \\ \quad \textit{iff } [X \models_s \alpha_1 \textit{ or } ... \textit{ or } X \models_s \alpha_n] \\ \qquad \textit{and } \forall i \textit{ s.t. } 1 \leq i \leq n \;\; [X \models_s \sim\alpha_i \textit{ implies } X \models_s Focus(\alpha_1 \vee ... \vee \alpha_n, \alpha_i)] \end{array}$$

For $\alpha \in \mathcal{L}$ and $X \in \wp(\mathcal{O})$, α is strongly satisfiable in X iff $X \models_s \alpha$. Also, for $\alpha \in \mathcal{L}$, α is strongly satisfiable iff $\exists X \in \wp(\mathcal{O})$ such that $X \models_s \alpha$.

The first two parts of this definition cover literals. The third part covers disjunction. This definition for disjunction is more restricted than the classical definition. In addition to at least one disjunct being satisfiable, there is also a notion of focusing incorporated into the definition. Essentially, for each disjunct α_i in the formula, $\alpha_1 \vee ... \vee \alpha_n$, if the model strongly satisfies the complement $\sim\alpha_i$ of that disjunct, then the model must also strongly satisfy the focused formula $Focus(\alpha_1 \vee ... \vee \alpha_n, \alpha_i)$.

So if $\alpha \vee \beta$ is a disjunction, where α and β are atoms then the above definition reduces to the following, where $Focus(\alpha \vee \beta, \beta)$ is α, and $Focus(\alpha \vee \beta, \alpha)$ is β.

$$\begin{array}{ll} X \models_s \alpha \vee \beta \textit{ iff } & [X \models_s \alpha \textit{ or } X \models_s \beta] \\ & \textit{and } [X \models_s \neg\alpha \textit{ implies } X \models_s \beta] \\ & \textit{and } [X \models_s \neg\beta \textit{ implies } X \models_s \alpha] \end{array}$$

The reason we need this definition for disjunction that is more restricted than the classical one, is that we have decoupled the link between a formula and its negation in the model. Therefore, in order to provide a meaning for resolution, we need to put the link between each disjunct, and its complement, into the definition for disjunction. As a result, to ensure a clause is strongly satisfiable, we need to ensure that if necessary, every more focused clause is also strongly satisfiable.

Example 3.1 *Let* $\Delta = \{\neg\alpha \vee \neg\beta \vee \gamma, \neg\alpha \vee \gamma, \neg\gamma\}$*, where* $\alpha, \beta, \gamma \in \mathcal{A}$*, and let* $X = \{-\alpha, -\beta, -\gamma\}$*. So* $X \models_s \neg\alpha$*,* $X \models_s \neg\beta$ *and* $X \models_s \neg\gamma$*. Also,* $X \models_s \sim\gamma$*. Hence,* $X \models_s \neg\alpha \vee \gamma$*, and* $X \models_s \neg\alpha \vee \neg\beta$*, and so,* $X \models_s \neg\alpha \vee \neg\beta \vee \gamma$*. Hence every formula in* Δ *is strongly satisfiable in* X*.*

Example 3.2 *Let* $\Delta = \{\alpha, \neg\alpha\}$*, where* $\alpha \in \mathcal{A}$*, and let* $X = \{+\alpha, -\alpha\}$*. So* $X \models_s \alpha$ *and* $X \models_s \neg\alpha$*.*

Example 3.3 *Let* $\Delta = \{\alpha, \neg\alpha \vee \neg\beta, \beta\}$*, where* $\alpha, \beta \in \mathcal{A}$*, and let* $X = \{+\alpha, +\beta, -\alpha, -\beta\}$*. So* $X \models_s \alpha$*,* $X \models_s \beta$*,* $X \models_s \neg\alpha$*,* $X \models_s \neg\beta$*, and* $X \models_s \neg\alpha \vee \neg\beta$*.*

Example 3.4 *Let* $\Delta = \{\alpha, \neg\alpha \vee \beta\}$*, where* $\alpha, \beta \in \mathcal{A}$*, and let* $X = \{+\alpha, -\alpha, +\beta\}$*. So* $X \models_s \alpha$*,* $X \models_s \neg\alpha$*,* $X \models_s \beta$*, and* $X \models_s \neg\alpha \vee \beta$*. Other models that strongly satisfy all the formulae in* Δ *include* $Y = \{+\alpha, +\beta\}$ *and* $Z = \{+\alpha, -\alpha, +\beta, +\gamma\}$*.*

The following theorem provides a slightly different view on the semantics of disjunction.

Proposition 3.1 *Let* $X \in \wp(\mathcal{O})$*, and* $\alpha_1, .., \alpha_n$ *be literals in* $\mathcal{L}$*.*

$$X \models_s \alpha_1 \vee .. \vee \alpha_n \text{ iff } \begin{array}{l} (1) \text{ for some } \alpha_i \in \{\alpha_1, .., \alpha_n\}, +\alpha_i \in X \text{ and } -\alpha_i \notin X \\ \text{or} \\ (2) \text{ for all } \alpha_i \in \{\alpha_1, .., \alpha_n\}, +\alpha_i \in X \text{ and } -\alpha_i \in X \end{array}$$

Proof We consider the simplest case of $\phi \vee \psi$, where ϕ and ψ are atoms. By Definition 3.4, we have, $X \models_s \phi \vee \psi$

$$\begin{array}{ll} \text{iff} & (X \models_s \phi \text{ or } X \models_s \psi) \\ & \text{and } (X \models_s \neg\phi \text{ implies } X \models_s \psi) \\ & \text{and } (X \models_s \neg\psi \text{ implies } X \models_s \phi) \\ \\ \text{iff} & (+\phi \in X \text{ or } +\psi \in X) \\ & \text{and } (-\phi \notin X \text{ or } +\psi \in X) \\ & \text{and } (-\psi \notin X \text{ or } +\phi \in X) \\ \\ \text{iff} & (+\phi \in X \text{ and } -\phi \notin X) \\ & \text{or } (+\psi \in X \text{ and } -\psi \notin X) \\ & \text{or } (+\phi \in X \text{ and } -\phi \in X \text{ and } +\psi \in X \text{ and } -\psi \in X) \\ \\ \text{iff} & (1) \text{ for some } \alpha_i \in \{\phi, \psi\}, +\alpha_i \in X \text{ and } -\alpha_i \notin X \\ & \text{or} \\ & (2) \text{ for all } \alpha_i \in \{\phi, \psi\}, +\alpha_i \in X \text{ and } -\alpha_i \in X \end{array}$$

We can straightforwardly generalize this for a disjunction $\alpha_1 \vee .. \vee \alpha_n$, where $n > 2$, and $\alpha_1, .., \alpha_n$ be literals. $\Box$

Strong satisfaction therefore provides a semantic account for paraconsistent reasoning using resolution. Given a set of clauses as assumptions, we can query using this paraconsistent reasoning system. So a query α is satisfied if there is a belief α derivable from the assumptions. However, a query might be less focused than the resolvents. For example, suppose $\alpha \vee \beta$ is a query, and there is no belief $\alpha \vee \beta$ derivable from the assumptions using reflexivity or resolution, though there is a belief α in the assumptions. Here, it would be reasonable to allow disjunction introduction, defined below, to be used to derive $\alpha \vee \beta$ from the α. In this way the query can be satisfied.

$$\frac{\alpha}{\alpha \vee \beta}$$

However, for paraconsistent reasoning, disjunction introduction cannot be followed by further applications of resolution. Otherwise trivial inferences would follow from inconsistent information. For example, let Δ be $\{\alpha, \neg\alpha\}$. By disjunction introduction, $\alpha \vee \beta$ is an inference. If resolution were allowed on this inference, then β is an inference, which is a trivial inference.

Proofs in this paraconsistent reasoning are now two-stage affairs. The first is decompositional, forming resolvents from clauses using resolution. The second is compositional, forming clauses from the assumptions and resolvents, using disjunction introduction. Furthermore, any clause obtained will be non-trivial inference from the assumptions.

As a result of extending the proof theory, we also need to extend the semantics. First, we require the following definition of weak satisfaction.

Definition 3.5 *Let $\models_w$ be a satisfiability relation, called weak satisfaction, such that $\models_w \subseteq \wp(\mathcal{O}) \times \mathcal{L}_{clauses}$. For $X \in \wp(\mathcal{O})$, we define $\models_w$ as follows, where $\alpha_1, ..., \alpha_n$ are literals in $\mathcal{L}$ and α is an atom in $\mathcal{L}$.*

$$X \models_w \alpha \text{ if } +\alpha \in X$$

$$X \models_w \neg\alpha \text{ if } -\alpha \in X$$

$$X \models_w \alpha_1 \vee ... \vee \alpha_n \text{ iff } [X \models_w \alpha_1 \text{ or ... or } X \models_w \alpha_n]$$

Weak satisfaction is weaker than strong satisfaction in that it does not incorporate focusing, and indeed seems closer to a classical notion of satisfaction.

In the following definition, we can see that QC entailment is of the same form as classical entailment except we use strong satisfaction for the assumptions and weak satisfaction for the inference.

Definition 3.6 *$\models_Q$ is an entailment relation iff $\models_Q \subseteq \wp(\mathcal{L}_{clauses}) \times \mathcal{L}_{clauses}$, and,*

$$\{\alpha_1, .., \alpha_n\} \models_Q \beta \text{ iff } \forall X (X \models_s \alpha_1 \text{ and } \ldots \text{ and } X \models_s \alpha_n \text{ implies } X \models_w \beta)$$

We can consider the strong satisfaction relation as capturing the decomposition of the set of assumptions. Models are acceptable to the strong satisfaction relation only if they support focusing. Strong satisfaction forces each resolvent β of a clause $\alpha \vee \beta$ to hold if $\sim\alpha$ holds. In contrast, we can consider the weak satisfaction relation as capturing the composition of formulae from resolvents, allowing disjuncts to be introduced.

Example 3.5 *Let $\Delta = \{\alpha\}$, where $\alpha \in \mathcal{A}$, and let $X1 = \{+\alpha\}$ and $X2 = \{+\alpha, -\alpha\}$. Now $X1 \models_s \alpha$, and $X2 \models_s \alpha$, whereas $X1 \models_s \alpha \vee \beta$, and $X2 \not\models_s \alpha \vee \beta$. However, $X1 \models_w \alpha \vee \beta$, and $X2 \models_w \alpha \vee \beta$, and indeed $\Delta \models_Q \alpha \vee \beta$.*

Example 3.6 *Let $\Delta = \{\alpha \vee \beta, \neg\alpha\}$, where $\alpha, \beta \in \mathcal{A}$. For all models X, if $X \models_s \alpha \vee \beta$, and $X \models_s \neg\alpha$, then $X \models_w \beta$. Hence, $\Delta \models_Q \alpha \vee \beta$, $\Delta \models_Q \neg\alpha$, and $\Delta \models_Q \beta$.*

Example 3.7 *Let $\Delta = \{\alpha, \neg\alpha\}$, where $\alpha \in \mathcal{A}$. For all models X, $X \models_s \alpha$ and $X \models_s \neg\alpha$ implies $X \models_w \alpha$, $X \models_w \neg\alpha$, and for any β, $X \models_w \alpha \vee \beta$, $X \models_w \neg\alpha \vee \beta$. Hence, $\Delta \models_Q \alpha$, $\Delta \models_Q \neg\alpha$, $\Delta \models_Q \alpha \vee \beta$, $\Delta \models_Q \neg\alpha \vee \beta$, but $\Delta \not\models_Q \beta$.*

One ramification of this definition for QC proof theory and semantics is that, in general, classical tautologies do not hold.

Example 3.8 *Let $\Delta = \emptyset$. Hence $X = \emptyset$ is a model that strongly satisfies all the formulae in Δ, but $\Delta \models_Q \alpha \vee \neg\alpha$ does not hold.*

In general, the failure of excluded middle, and other tautologies, is not a problem in applications in computing. Consider software engineers using logic for reasoning about some specification. Here tautologies tell them nothing useful for their task.

To summarize, given a set of clauses, as assumptions, a set of resolvents of these clauses can be formed by repeated application of resolution, and this set of inferences is such that each element is a non-trivial inference from the assumptions, and hence (potentially) useful. This idea can be extended to allow disjunction introduction as a proof rule, as long as it is not followed by a further application of resolution. These proof rules hold in classical logic, but the logic is weaker than classical logic. In particular, ex falso quodlibet does not hold.

The proof theory and semantics presented in this section is the basis of QC logic. Each formula in the language represents a belief, the proof theory is for deriving further beliefs from a set of beliefs, and the semantics captures satisfaction in terms of the existence of reasons for and against literal beliefs. In the following sections, we provide the full definitions for the proof theory and the semantics, and analyse the resulting logic. This will include soundness and completeness results relating the QC entailment relation with the QC consequence relation.

4 Full proof theory for QC logic

In the previous section, we presented the essential idea behind QC logic. Here, we develop the proof theory and semantics so that the logic can handle any set of formulae as assumptions and any formula as query.

4.1 QC proof rules

The QC proof theory is presented as a set of decomposition rules and a set of composition rules below. The decomposition rules apply to the assumptions and the composition rules apply to the query. All the QC proof rules hold in classical logic, but QC logic is weaker than classical logic.

Definition 4.1 *The following are the QC decomposition rules. We assume that conjunction and disjunction are commutative and associative.*

$$\frac{\alpha \wedge \beta}{\alpha} \quad \textit{[Conjunction elimination]}$$

$$\frac{\neg\neg\alpha \vee \beta}{\alpha \vee \beta} \quad \frac{\neg\neg\alpha}{\beta} \quad \textit{[Negation elimination]}$$

$$\frac{\alpha \vee \beta \quad \neg\alpha \vee \gamma}{\beta \vee \gamma} \quad \frac{\alpha \vee \beta \quad \neg\alpha}{\beta} \quad \frac{\alpha \quad \neg\alpha \vee \gamma}{\gamma} \quad \textit{[Resolution]}$$

$$\frac{\alpha \vee \alpha \vee \beta}{\alpha \vee \beta} \quad \frac{\alpha \vee \alpha}{\alpha} \quad \textit{[Disjunction contraction]}$$

$$\frac{\alpha \vee (\beta \rightarrow \gamma)}{\alpha \vee \neg\beta \vee \gamma} \quad \frac{\alpha \vee \neg(\beta \rightarrow \gamma)}{\alpha \vee (\beta \wedge \neg\gamma)} \quad \textit{[Arrow elimination]}$$

$$\frac{\beta \rightarrow \gamma}{\neg\beta \vee \gamma} \quad \frac{\neg(\beta \rightarrow \gamma)}{(\beta \wedge \neg\gamma)}$$

$$\frac{\alpha \vee (\beta \wedge \gamma)}{(\alpha \vee \beta) \wedge (\alpha \vee \gamma)} \quad \frac{(\alpha \wedge \beta) \vee (\alpha \wedge \gamma)}{\alpha \wedge (\beta \vee \gamma)} \quad \textit{[Decompositional distribution]}$$

$$\frac{\neg(\alpha \wedge \beta) \vee \gamma}{\neg\alpha \vee \neg\beta \vee \gamma} \quad \frac{\neg(\alpha \vee \beta) \vee \gamma}{(\neg\alpha \wedge \neg\beta) \vee \gamma} \quad \textit{[Decompositional de Morgan laws]}$$

$$\frac{\neg(\alpha \wedge \beta)}{\neg\alpha \vee \neg\beta} \quad \frac{\neg(\alpha \vee \beta)}{\neg\alpha \wedge \neg\beta}$$

Definition 4.2 *The following are the QC composition rules. We assume that conjunction and disjunction are commutative and associative.*

$$\frac{\alpha \quad \beta}{\alpha \wedge \beta} \quad \textit{[Conjunction introduction]} \qquad \frac{\alpha}{\alpha \vee \beta} \quad \textit{[Disjunction introduction]}$$

$$\frac{\alpha}{\neg\neg\alpha} \quad \frac{\alpha \vee \beta}{\neg\neg\alpha \vee \beta} \quad \textit{[Negation introduction]}$$

$$\frac{\neg\beta \vee \gamma}{\beta \rightarrow \gamma} \qquad \frac{\beta \wedge \neg\gamma}{\neg(\beta \rightarrow \gamma)} \qquad \textit{[Arrow introduction]}$$

$$\frac{\alpha \vee \neg\beta \vee \gamma}{\alpha \vee (\beta \rightarrow \gamma)} \quad \frac{\alpha \vee (\beta \wedge \neg\gamma)}{\alpha \vee \neg(\beta \rightarrow \gamma)}$$

$$\frac{(\alpha \vee \beta) \wedge (\alpha \vee \gamma)}{\alpha \vee (\beta \wedge \gamma)} \quad \frac{\alpha \wedge (\beta \vee \gamma)}{(\alpha \wedge \beta) \vee (\alpha \wedge \gamma)} \qquad \textit{[Compositional distribution]}$$

$$\frac{\neg\alpha \vee \neg\beta \vee \gamma}{\neg(\alpha \wedge \beta) \vee \gamma} \qquad \frac{(\neg\alpha \wedge \neg\beta) \vee \gamma}{\neg(\alpha \vee \beta) \vee \gamma} \quad \textit{[Compositional de Morgan laws]}$$

$$\frac{\neg\alpha \vee \neg\beta}{\neg(\alpha \wedge \beta)} \qquad \frac{\neg\alpha \wedge \neg\beta}{\neg(\alpha \vee \beta)}$$

The decomposition rules allow clauses to be formed from formulae of the full language by eliminating connectives. The composition rules allow formulae of the full language to be formed from clauses by introducing connectives. We explain this more fully below.

4.2 QC proofs

QC logic can be applied to any set of classical formulae (ie. any $\Delta \subseteq \mathcal{L}$) as assumptions, and any classical formula (ie. any $\alpha \in \mathcal{L}$) as a query.

Definition 4.3 *T is a decomposition tree iff T is a tree where each node is an element of $\mathcal{L}$, and any node that is not a leaf is derived by the application of a QC decomposition rule. For any such node, each parent is a premise of the decomposition rule.*

Definition 4.4 *Let $\Delta \in \wp(\mathcal{L})$. For a clause β, there is a decomposition of β from Δ iff there is a decomposition tree, where each leaf is an element of Δ, and the root is β.*

Example 4.1 *For $\Delta = \{\alpha \vee \beta \vee \gamma, \neg\alpha, \neg\beta, \neg\gamma\}$ there are decompositions of $\beta \vee \gamma$, $\alpha \vee \beta$, $\alpha \vee \gamma$, α, β, and γ from Δ.*

Definition 4.5 *T is a composition tree iff T is a tree where each node is an element of $\mathcal{L}$, and any node that is not a leaf is derived by the application of a QC composition proof rule. For any such node, each parent is a premise of the decomposition proof rule.*

Definition 4.6 *For a set of clauses $\{\delta_1, .., \delta_n\}$, and a formula β, there is a composition of β from $\{\delta_1, .., \delta_n\}$ iff there is a composition tree, where each leaf is an element of $\{\delta_1, .., \delta_n\}$, and the root is β.*

Definition 4.7 *Let $\Delta \in \wp(\mathcal{L})$, and $\alpha \in \mathcal{L}$. We define the QC consequence relation, denoted $\vdash_Q$, as follows:*

$$
\begin{array}{lll}
\Delta \vdash_Q \alpha \textit{ iff} & \exists \gamma_1, ..., \gamma_n \textit{ such that} & \\
& \quad \textit{there is a decomposition of } \gamma_1 \textit{ from } \Delta & \\
& \qquad\quad \textit{and} & \\
& \qquad\quad \vdots & \\
& \qquad\quad \textit{and} & \\
& \quad \textit{there is a decomposition of } \gamma_n \textit{ from } \Delta & \\
& \qquad\quad \textit{and} & \\
& \quad \textit{there is a composition of } \alpha \textit{ from } \{\gamma_1, .., \gamma_n\} &
\end{array}
$$

This definition captures the way that the proof theory is restricted to only allowing compositional proof rules to be used after decompositional rules.

Example 4.2 *Let $\Delta = \{\alpha, \alpha \rightarrow \beta\}$. From $\alpha \rightarrow \beta$, we get $\neg\alpha \vee \beta$ by arrow elimination. Then we get β by resolution. So there is a decomposition of β from Δ, and hence*

$$\{\alpha, \alpha \rightarrow \beta\} \vdash_Q \beta$$

Example 4.3 *Let $\Delta = \{\neg\beta, \alpha \rightarrow \beta\}$. From $\alpha \rightarrow \beta$, we get $\neg\alpha \vee \beta$ by arrow elimination. Then we get $\neg\alpha$ by resolution on $\neg\alpha \vee \beta$ and $\neg\beta$. So there is a decomposition of $\neg\alpha$ from Δ, and therefore,*

$$\{\neg\beta, \alpha \rightarrow \beta\} \vdash_Q \neg\alpha$$

Example 4.4 *Let $\Delta = \{\alpha \wedge \beta, \neg\beta \vee \gamma\}$. From $\alpha \wedge \beta$, we obtain β by conjunction elimination, and γ by resolution on β and $\neg\beta \vee \gamma$. So γ is a decomposition from Δ and $\gamma \vee \delta$ is a composition from γ. Hence,*

$$\{\alpha \wedge \beta, \neg\beta \vee \gamma\} \vdash_Q \gamma \vee \delta$$

Example 4.5 *Let $\Delta = \{\alpha, \beta, \gamma\}$. There is a decomposition of α from Δ and a decomposition of β from Δ. From α, we get $\neg\neg\alpha$ by double negation introduction, and $\neg\neg\alpha \vee \phi$ by disjunction introduction. Similarly, we get $\neg\neg\beta \vee \delta$ from β. Then by disjunction introduction, we get $(\neg\neg\alpha \vee \phi) \wedge (\neg\neg\beta \vee \psi)$. So there is a composition of $(\neg\neg\alpha \vee \phi) \wedge (\neg\neg\beta \vee \psi)$ from $\{\alpha, \beta\}$. Hence,*

$$\{\alpha, \beta, \gamma\} \vdash_Q (\neg\neg\alpha \vee \phi) \wedge (\neg\neg\beta \vee \psi)$$

Example 4.6 *Consider Δ comprising the following formulae, $\alpha \wedge (\beta \vee (\gamma \rightarrow \delta))$, $\neg\beta \wedge \neg\delta$, and $\neg\gamma \vee \neg\alpha$. For Δ, there is a decomposition of $\neg\gamma$ from Δ, and there is a decomposition of $\beta \vee \neg\gamma$ from Δ, and there is a composition of $\neg((\gamma \rightarrow \beta) \rightarrow \gamma)$ from $\{\neg\gamma, \beta \vee \neg\gamma\}$. So,*

$$\{\alpha \wedge (\beta \vee (\gamma \rightarrow \delta)), \neg\beta \wedge \neg\delta, \neg\gamma \vee \neg\alpha\} \vdash_Q \neg((\gamma \rightarrow \beta) \rightarrow \gamma)$$

Example 4.7 *For $\Delta = \{\alpha \vee \beta, \alpha \vee \neg\beta, \neg\alpha \wedge \delta\}$,*

$$\{\alpha \vee \beta, \alpha \vee \neg\beta, \neg\alpha \wedge \delta\} \vdash_Q \alpha \vee \beta$$
$$\{\alpha \vee \beta, \alpha \vee \neg\beta, \neg\alpha \wedge \delta\} \vdash_Q \alpha \vee \neg\beta$$
$$\{\alpha \vee \beta, \alpha \vee \neg\beta, \neg\alpha \wedge \delta\} \vdash_Q \alpha$$
$$\{\alpha \vee \beta, \alpha \vee \neg\beta, \neg\alpha \wedge \delta\} \vdash_Q \neg\alpha$$
$$\{\alpha \vee \beta, \alpha \vee \neg\beta, \neg\alpha \wedge \delta\} \vdash_Q \delta$$
$$\{\alpha \vee \beta, \alpha \vee \neg\beta, \neg\alpha \wedge \delta\} \not\vdash_Q \neg\delta$$
$$\{\alpha \vee \beta, \alpha \vee \neg\beta, \neg\alpha \wedge \delta\} \not\vdash_Q \gamma \vee \phi$$
$$\{\alpha \vee \beta, \alpha \vee \neg\beta, \neg\alpha \wedge \delta\} \not\vdash_Q \neg\phi \wedge \neg\psi$$

Example 4.8 *For $\Delta = \{\alpha \vee (\beta \wedge \gamma), \neg\beta\}$,*

$$\{\alpha \vee (\beta \wedge \gamma), \neg\beta\} \vdash_Q \alpha \vee \beta$$
$$\{\alpha \vee (\beta \wedge \gamma), \neg\beta\} \vdash_Q \alpha \vee \gamma$$
$$\{\alpha \vee (\beta \wedge \gamma), \neg\beta\} \vdash_Q \alpha$$
$$\{\alpha \vee (\beta \wedge \gamma), \neg\beta\} \vdash_Q \neg\beta$$

4.3 Some properties of QC proof theory

First we consider some simple results concerning the proof theory.

Definition 4.8 *Let $CNF(\Delta)$ be a minimal set of formulae such that for every $\alpha \in \Delta$, there is a $\beta \in CNF(\Delta)$, where β is a CNF of α. Let $Clauses(\Delta) = \bigcup\{\{\alpha_1, ..., \alpha_n\} \mid \alpha_1 \wedge ... \wedge \alpha_n \in CNF(\Delta)\}$. $Resolvents(\Delta)$ is the closure of $Clauses(\Delta)$ under resolution. So for any $\alpha \vee \beta$, $\neg\beta \vee \gamma \in Resolvents(\Delta)$, we have $\alpha \vee \gamma \in Resolvents(\Delta)$.*

Proposition 4.1 *For all $\alpha \in Resolvents(\Delta)$, $\alpha \in Resolvents(CNF(\Delta))$ and $\alpha \in Resolvents(Clauses(\Delta))$*

Proposition 4.2 *For any formula α, if $\beta \in Clauses(\{\alpha\})$, then there is a decomposition tree where α is the leaf and β is the root. Furthermore, the tree involves no application of resolution.*

Proof Assume $\beta \in Clauses(\{\alpha\})$. So, there is a π such that π is a CNF of β and α is a conjunct of π. Now, π is obtained by a sequence of applications of rewrites. Let us consider this rewrite process as the construction of a tree, called a CNF tree, that is defined as follows:

- Let $f : \mathcal{L} \longrightarrow \mathcal{L}$ be a function defined by the de Morgan's law, double negation elimination, arrow elimination, disjunction contraction, and distribution rules given in Definition 4.1 so that if X is the premise of a rule, and Y is the consequent, then $f(X) = Y$ holds. For example, $f(\neg\neg\phi \vee \psi)$ $= \phi \vee \psi$, and $f(\alpha \vee (\beta \wedge \gamma)) = (\alpha \vee \beta) \wedge (\alpha \vee \gamma)$. Each application of f is a rearrangement step, since each step takes one formula and returns a rearrangement of it.

- Let $g : \mathcal{L} \longrightarrow \mathcal{L}$ be a function defined as $g(X \wedge Y) = f(X) \wedge f(Y)$, where $f(X) = X$ when X is a conjunction or a literal. For example, if ϕ and ψ are atoms, $g(\neg\neg\phi \wedge \psi) = f(\neg\neg\phi) \wedge f(\psi)$, where $f(\neg\neg\phi) = \phi$, and $f(\psi)$ $= \psi$. Each application of g is a recursive step since each call to g on a conjunction is a call to f on each conjunct.

A CNF tree is obtained by repeatedly and exhaustively applying f and g. Each node denotes a formula. Each step $f(X) = Y$ is represented by an arc (X, Y) in the tree and each step $g(X) = f(Y) \wedge f(Z)$ is represented by a bifurcation in the tree with a pair of arcs (X, Y) and (X, Z). There is a one-to-one correspondence between the sequence of steps and the sequence of nodes. Each leaf of the tree is a clause, and the conjunction of the leaves is a formula that is a CNF of the formula at the root.

In the same way, a QC decomposition (excluding use of resolution) can be defined in terms of f and the function g' given below.

- Let $g' : \mathcal{L} \longrightarrow \mathcal{L}$ be a function defined as $g'(X \wedge Y) = f(X)$, where $f(X) = X$ when X is a conjunction or a literal. For example, $g'(\neg\neg\phi \wedge \psi)$ $= f(\neg\neg\phi)$. Each application of g' is also a recursive step since each call to g' on a conjunction is a call to f on one of the conjuncts.

A QC decomposition tree is obtained by repeatedly and exhaustively applying f and g'. Each node denotes a formula. Each step $f(X) = Y$ is represented by an arc (Y, X) in the tree and each step $g'(X) = f(Y)$ is represented an arc (Y, X) in the tree. There is a one-to-one correspondence between the sequence of steps and the sequence of nodes. Each leaf is a clause.

Clearly, for any CNF tree, we can form a QC decomposition tree by taking one branch from root to leaf inclusive, and reversing the direction of the arcs. So if $\beta \in Clauses(\{\alpha\})$, there is a CNF tree where the root is α, and a leaf is β, and therefore there is a QC decomposition tree where the root is β, and the leaf is α.□

Proposition 4.3 *For any $\Delta \in \wp(\mathcal{L})$, if $\beta \in Resolvents(\Delta)$, then there is a decomposition tree where the leaves are elements from Δ, and β is the root.*

Proof Follows directly from Proposition 4.2.

Proposition 4.4 *For any formula α, if $\beta_1 \wedge ... \wedge \beta_n$ is a CNF of α, then there is a composition tree where $\beta_1, .., \beta_n$ are the leaves and α is the root.*

Proof Analogous to the proof for Proposition 4.2.

The following result means that in general there are no classical tautologies that hold for the empty database for the $\vdash_Q$ relation.

Proposition 4.5 *For $\Delta = \emptyset$, there are no $\alpha \in \mathcal{L}$ such that $\Delta \vdash_Q \alpha$*

Proof Assume $\Delta = \emptyset$. Also assume $\exists \beta \in \mathcal{L}$ such that $\Delta \vdash_Q \beta$. Therefore either (option 1) there is a composition of β from $\emptyset$ or (option 2) there is a γ_1 such that there is a decomposition of γ_1 from $\emptyset$, and ...., and there is a γ_n such that there is a decomposition of γ_n from $\emptyset$, and there is a composition of β from $\{\gamma_1, ..., \gamma_n\}$. Now, (option 1) cannot hold because there are no composition proof rules that hold for an empty premise list. Similarly, (option 2) cannot hold because there are no decomposition rules that hold for an empty premise list. Hence, there is no β such that $\Delta \vdash_Q \beta$, when $\Delta = \emptyset$. □

In Section 3, we presented QC logic as a logic of beliefs, rather than as a logic of "truths". We therefore believe that failure of classical tautologies to hold in QC logic is intuitive, in general. For non-empty sets of assumptions, however, there are classical tautologies that follow. For example, let $\Delta = \{\beta\}$. So $\Delta \vdash_Q \beta \vee \neg\beta$ holds, and $\beta \vee \neg\beta$ is a classical tautology.

Clearly QC logic is paraconsistent according to the definition in Section 1. We also have the following result that ensures the reasoning is non-trivial.

Proposition 4.6 *Let $\Delta \in \wp(\mathcal{L})$, and $\alpha \in \mathcal{L}$. If $\Delta \vdash_Q \alpha$, then α is not a trivial inference from Δ. In other words, $Atoms(\Delta) \cap Atoms(\{\alpha\}) \neq \emptyset$.*

Proof In the definition of the QC consequence relation, the only proof rule that can introduce new atom symbols is disjunction introduction. However, any new atom symbol that is introduced by this proof rule is part of a disjunction where at least some of the other atom symbols occur in Δ, and the decomposition rules cannot be applied to remove these other atom symbols since no decomposition rule can be applied after any of the composition rules have been applied.□

However, if $\Delta \vdash \alpha$ and $Atoms(\Delta) \cap Atoms(\{\alpha\}) \neq \emptyset$, then it is not guaranteed that $\Delta \vdash_Q \alpha$ holds. For example, suppose $\Delta = \{\neg\beta\}$. Here, $\Delta \not\vdash_Q (\alpha \wedge \neg\alpha) \rightarrow \beta$.

Even if we restrict consideration to a non-tautological inference α that follows classically from a consistent set of formulae, we are not guaranteed that α will also follow from the QC consequence relation, as illustrated by the following example.

Example 4.9 *Let $\Delta = \{\alpha\}$. For Δ, $\beta \rightarrow (\alpha \wedge \beta)$ is a classical inference, but it is not a QC inference. To show this, $\{\alpha\} \vdash_Q \beta \rightarrow (\alpha \wedge \beta)$ iff there is a composition of $\beta \rightarrow (\alpha \wedge \beta)$ from $\{\alpha\}$ iff there is a composition tree where the root is $\beta \rightarrow (\alpha \wedge \beta)$ and the leaves are α. But there is no such composition tree. Though there is a composition tree with root $\beta \rightarrow (\alpha \wedge \beta)$ and leaves from $\{\alpha, \beta\}$. Similiarly, there is a composition tree with root $\beta \rightarrow (\alpha \wedge \beta)$ and leaves from $\{\alpha, \neg\beta\}$.*

In some important respects, QC logic does preserve classical reasoning as illustrated by the following result.

Proposition 4.7 *For $\Delta \in \wp(\mathcal{L})$, $\alpha \in \mathcal{L}$, we have the following equivalence: $\Delta \vdash_Q \alpha$ iff $\beta_1 \wedge .. \wedge \beta_n$ is a CNF of α and $\beta_1, .., \beta_n \in Resolvents(\Delta)$.*

Proof Follows from Proposition 4.3 and Proposition 4.4.

Excluded middle is not in $Resolvents(\Delta)$ for all Δ. So for any α, if $\beta_1 \wedge .. \wedge \beta_n$ is a CNF of α and there is a β_i $(1 \leq i \leq n)$ that is excluded middle, then $\Delta \vdash_Q \alpha$ is not guaranteed.

We now show that QC logic is decidable.

Lemma 4.1 *For any finite $\Delta \in \wp(\mathcal{L})$, the set $Resolvents(\Delta)$ is finite.*

Lemma 4.2 *For any finite $\Delta \in \wp(\mathcal{L})$, and for any decomposition proof tree T, if each leaf of T is an element of Δ, then T has a finite number of nodes.*

Proof (1) Each non-leaf node on each path from leaf to root is generated by a proof rule that either eliminates a connective, or pushes negation or disjunction into a subformula, and none of the proof rules can reverse this. Since for any formula in the tree, there is a finite number of connectives and of subformulae, there can only be a finite number of nodes on each path from leaf to root. (2) Since Δ is finite, there is a finite number of leaves in the tree, and hence a finite number of branches from leaf to root. As a result of (1) and (2), there is a finite number of nodes in T. □

Lemma 4.3 *For any finite set of clauses $\{\delta_1, ..., \delta_n\}$, and any $\alpha \in \mathcal{L}$ and for any composition tree T, if each leaf of T is an element from $\{\delta_1, ..., \delta_n\}$, and α is the root of T, then T has a finite number of nodes.*

Proof Analogous to proof of Lemma 4.2.

Proposition 4.8 *For any finite $\Delta \in \wp(\mathcal{L})$, and any $\alpha \in \mathcal{L}$, there is an effective procedure for determining whether $\Delta \vdash_Q \alpha$. In other words, the QC consequence relation is decidable.*

Proof Construct $Resolvents(\Delta)$. There are a finite number of items in this set (Lemma 4.1), and each item can be determined by constructing a decomposition tree, and so there is a finite number of decomposition trees that can be formed, and each of these has a finite number of nodes (Lemma 4.2). The set of all roots of all these decomposition trees is therefore $Resolvents(\Delta)$. Since $Resolvents(\Delta)$ is finite, $\wp(Resolvents(\Delta))$ is finite, and so there is a finite number of composition trees that can be formed from subsets of $Resolvents(\Delta)$, and each of these has a finite number of nodes (Lemma 4.3). So, $\Delta \vdash_Q \alpha$ holds iff α is a root of least one of these composition trees. Therefore, for any $\Delta \in \wp(\mathcal{L}), \alpha \in \mathcal{L}$, $\Delta \vdash_Q \alpha$ or $\Delta \not\vdash_Q \alpha$ can be determined in a finite number of steps. □

We finish this section by including the following definition and proposition that we will require later.

Definition 4.9 *T is a normal decomposition tree iff T is a decomposition tree and for each node that is derived by an application of resolution, every subsequent node is derived by resolution or disjunct contraction.*

Proposition 4.9 *For every decomposition tree T, where each leaf is an element of Δ and the root is β, there is a normal decomposition tree N, where each leaf is an element of Δ and the root is β.*

Proof If T has only one leaf, then T involves no applications of resolution, and so N can be formed directly from T. Now, assume a decomposition tree T where the leaves are $\alpha_1, .., \alpha_n$, and the root is β. From Proposition 4.1 for any $\tau \in Resolvents(\Delta)$, τ is also in $Resolvents(Clauses(\Delta))$. So we can form N with leaves $\alpha_1, .., \alpha_n$ and root β by defining a subtree with leaves $\beta_1, .., \beta_n$ and root β such that this subtree only contains applications of resolution and disjunct contraction. We now form N from this subtree by defining a branch from each leaf α_i to the node β_i such that each of these branches is a decomposition tree involving no application of resolution or disjunction contraction as demonstrated in the proof of Proposition 4.2. □

4.4 Characterization of the QC consequence relation

We now consider properties of the consequence relation. These properties have been much discussed in the context of non-monotonic logics [Gab85, GM93] and of relevance logics [AB75, Ten84]. At the end of this section, we discuss the relevance to QC logic of these properties and associated results.

Proposition 4.10 *The property of reflexivity, defined as follows, succeeds for the QC consequence relation, where* $\Delta \in \wp(\mathcal{L}), \alpha \in \mathcal{L}$.

$$\Delta \cup \{\alpha\} \vdash_Q \alpha$$

Proof If α is an assumption, then for each clause α_i, where $\alpha_1 \wedge ... \wedge \alpha_n$ is a CNF of α, by Proposition 4.2, there is a decomposition tree where α is the leaf and α_i is the root. Now, by Proposition 4.3, there is a composition tree where $\alpha_1, ..., \alpha_n$ are the leaves and α is the root. □

Proposition 4.11 *The property of monotonicity, defined as follows, succeeds for the QC consequence relation, where* $\Delta \in \wp(\mathcal{L})$, $\alpha \in \mathcal{L}$.

$$\Delta \vdash_Q \alpha \textit{ implies } \Delta \cup \{\beta\} \vdash_Q \alpha$$

Proof The assumption $\Delta \vdash_Q \alpha$ holds if and only if there is a set of clauses Γ such that there is a composition of α from Γ and for each $\gamma \in \Gamma$, there is a decomposition of γ from Δ. Since each decomposition of γ only requires a subset of Δ, then each decomposition of γ from Δ is a decomposition of γ from $\Delta \cup \{\beta\}$. □

Proposition 4.12 *The property of and-introduction, defined as follows, succeeds for the QC consequence relation, where* $\Delta \in \wp(\mathcal{L})$, $\alpha \in \mathcal{L}$.

$$\Delta \vdash_Q \alpha \textit{ and } \Delta \vdash_Q \beta \textit{ implies } \Delta \vdash_Q \alpha \wedge \beta$$

Proof Let Γ_α be a set of clauses such that there is a composition of α from Γ_α and for each γ in Γ_α, there is a decomposition of γ from Δ. Let Γ_β be a set of clauses formed similarly. Then $\Gamma_{\alpha\wedge\beta}$ is a set of clauses such that there is a composition of $\alpha \wedge \beta$ from $\Gamma_{\alpha\wedge\beta}$ and for each γ in $\Gamma_{\alpha\wedge\beta}$, there is a decomposition of γ from Δ. □

Proposition 4.13 *The property of or-elimination, defined as follows, succeeds for the QC consequence relation, where* $\Delta \in \wp(\mathcal{L})$, $\alpha \in \mathcal{L}$.

$$\Delta \cup \{\alpha\} \vdash_Q \gamma \textit{ and } \Delta \cup \{\beta\} \vdash_Q \gamma \textit{ implies } \Delta \cup \{\alpha \vee \beta\} \vdash_Q \gamma$$

Proof We consider the simple case where Δ is a set of clauses, and α, β, γ are clauses. First assume we can obtain $\gamma \in Resolvents(\Delta \cup \{\alpha\})$ and $\gamma \in Resolvents(\Delta \cup \{\beta\})$. Therefore $\gamma \vee \beta \in Resolvents(\Delta \cup \{\alpha \vee \beta\})$. As a result $\gamma \vee \gamma \in Resolvents(\Delta \cup \{\alpha \vee \beta\})$. By disjunct contraction, we obtain $\Delta \cup \{\alpha \vee \beta\} \vdash_Q \gamma$. Generalizing to any $\Delta \in \wp(\mathcal{L})$ and any $\alpha, \beta, \gamma \in \mathcal{L}$ is straightforward. □

Proposition 4.14 *The property of consistency preservation succeeds for the QC consequence relation, where* $\Delta \in \wp(\mathcal{L})$, $\alpha \in \mathcal{L}$.

$$\Delta \vdash_Q \alpha \wedge \neg\alpha \text{ implies } \Delta \vdash \alpha \wedge \neg\alpha$$

Proof $\Delta \vdash_Q \alpha \wedge \neg\alpha$
iff $\alpha \in Resolvents(CNF(\Delta))$ and $\neg\alpha \in Resolvents(CNF(\Delta))$
implies $\Delta \vdash \alpha \wedge \neg\alpha$. □

Proposition 4.15 *The property of supraclassicality, defined as follows, fails for the QC consequence relation, where* $\Delta \in \wp(\mathcal{L})$, $\alpha \in \mathcal{L}$.

$$\Delta \vdash \alpha \text{ implies } \Delta \vdash_Q \alpha$$

Proof Let $\Delta = \emptyset$, so $\Delta \vdash \alpha \vee \neg\alpha$, but $\Delta \not\vdash_Q \alpha \vee \neg\alpha$. □

Proposition 4.16 *The property of right modus ponens, defined as follows, fails for the QC consequence relation, where* $\Delta \in \wp(\mathcal{L})$, $\alpha \in \mathcal{L}$.

$$\Delta \vdash \alpha \text{ and } \Delta \vdash \alpha \rightarrow \beta \text{ implies } \Delta \vdash_Q \beta$$

Proof Let $\Delta = \{\alpha, \neg\alpha\}$. So $\Delta \vdash_Q \alpha$, and $\Delta \vdash_Q \alpha \rightarrow \beta$, but $\Delta \not\vdash_Q \beta$.□

Proposition 4.17 *The property of conditionalization, defined as follows, fails for the QC consequence relation, where* $\Delta \in \wp(\mathcal{L})$, $\alpha \in \mathcal{L}$.

$$\Delta \cup \{\alpha\} \vdash_Q \beta \text{ implies } \Delta \vdash_Q \alpha \rightarrow \beta$$

Proof Let $\Delta = \emptyset$. Here $\Delta \cup \{\alpha\} \vdash_Q \alpha$, but $\Delta \not\vdash_Q \alpha \rightarrow \alpha$. □

Proposition 4.18 *The property of deduction, defined as follows, fails for the QC consequence relation, where* $\Delta \in \wp(\mathcal{L})$, $\alpha \in \mathcal{L}$.

$$\Delta \vdash_Q \alpha \rightarrow \beta \text{ implies } \Delta \cup \{\alpha\} \vdash_Q \beta$$

Proof Let $\Delta = \{\neg\alpha\}$. Here $\Delta \vdash_Q \alpha \rightarrow \beta$, but $\Delta \cup \{\alpha\} \not\vdash_Q \beta$. □

Proposition 4.19 *The property of cut, defined as follows, fails for the QC consequence relation, where* $\Delta \in \wp(\mathcal{L})$, $\alpha \in \mathcal{L}$.

$$\Delta \cup \{\alpha\} \vdash_Q \beta \text{ and } \Gamma \vdash_Q \alpha \text{ implies } \Delta \cup \Gamma \vdash_Q \beta$$

Proof Consider that $\{\neg\alpha\} \cup \{\alpha \vee \beta\} \vdash_Q \beta$ and $\{\alpha\} \vdash_Q \alpha \vee \beta$, but that $\{\neg\alpha, \alpha\} \not\vdash_Q \beta$. □

Proposition 4.20 *The property of unit cumulativity* [1]*, defined as follows, fails for the QC consequence relation, where* $\Delta \in \wp(\mathcal{L})$*,* $\alpha \in \mathcal{L}$*.*

$$\Delta \vdash_Q \beta \text{ and } \Delta \cup \{\beta\} \vdash_Q \gamma \text{ implies } \Delta \vdash_Q \gamma$$

Proof Consider that $\{\neg\alpha, \alpha\} \vdash_Q \alpha \vee \beta$ and $\{\neg\alpha, \alpha\} \cup \{\alpha \vee \beta\} \vdash_Q \beta$, but $\{\neg\alpha, \alpha\} \not\vdash_Q \beta$. □

Proposition 4.21 *The property of right weakening fails for the QC consequence relation, where* $\Delta \in \wp(\mathcal{L})$*,* $\alpha \in \mathcal{L}$*.*

$$\Delta \vdash_Q \alpha \text{ and } \vdash \alpha \rightarrow \beta \text{ implies } \Delta \vdash_Q \beta$$

Proof Let $\Delta = \{\alpha\}$. Then $\{\alpha\} \vdash_Q \alpha$ holds. Consider $\vdash \alpha \rightarrow \beta \vee \neg\beta$. But $\{\alpha\} \not\vdash_Q \beta \vee \neg\beta$ holds. □

Proposition 4.22 *The property of left logical equivalence fails for the QC consequence relation, where* $\Delta \in \wp(\mathcal{L})$*,* $\alpha \in \mathcal{L}$*.*

$$\Delta \cup \{\alpha\} \vdash_Q \gamma \text{ and } \vdash \alpha \leftrightarrow \beta \text{ implies } \Delta \cup \{\beta\} \vdash_Q \gamma$$

Proof Let $\Delta = \emptyset$. Now $\{\alpha \vee \neg\alpha\} \vdash_Q \alpha \vee \neg\alpha$ and $\vdash (\alpha \vee \neg\alpha) \leftrightarrow (\beta \vee \neg\beta)$ hold. But, $\{\beta \vee \neg\beta\} \not\vdash_Q \alpha \vee \neg\alpha$. □

We summarize the properties of the QC consequence relation in Table 1. There is some debate over the minimal properties that a consequence relation should support. There are proposals that the minimal properties should be reflexivity, transitivity (cut), and monotonicity (see for example [Tar56]). However, to support some forms of non-classical reasoning, some compromise on these properties needs to be made. For non-monotonic reasoning — a form of reasoning with information that is in some sense inconsistent [BH98] — involves compromising on monotonicity [Gab85]. However, this compromise results in other properties also being compromised. There are many choices over which combinations of properties can be supported for non-monotonic reasoning [Mak94].

The properties of reflexivity and monotonicity hold for the QC consequence relation. However, the property of cut fails for the QC consequence relation, because to ensure non-trivial reasoning, decomposition rules cannot be applied

[1]The property of unit cumulativity is a consequence of the property of cut. However, the proof for the proposition is given to further illustrate reasoning for the QC consequence relation.

Property	Holds?
Reflexivity	Yes
Monotonicity	Yes
And-introduction	Yes
Or-elimination	Yes
Consistency preservation	Yes
Supraclassicality	No
Right modus ponens	No
Conditionalization	No
Deduction	No
Cut	No
Unit cumulativity	No
Right weakening	No
Left logical equivalence	No

Table 1: Summary of properties of the QC consequence relation

after composition rules in a proof. This means that proofs cannot be composed to give longer proofs, and so we can only view inference as a one-step process, not an interative process.

The rejection of cut (transitivity) has also been examined for relevance logics as a solution to the problem of Lewis' paradoxes:

$$\alpha \wedge \neg\alpha \vdash \beta$$

$$\beta \vdash \alpha \vee \neg\alpha$$

The first paradox concerns the irrelevancy arising from ex falso quodlibet. The second paradox concerns the irrelevancy arising from the tautology on the right hand side holding irrespective of the assumptions on the left hand side. In classical logic, the first paradox can hold because we can obtain $\alpha \vee \beta$ from α by disjunction introduction, and then β by disjunctive syllogism. Recognizing this, Anderson and Belnap, who were keen to retain transitivity, suggested the rejection of disjunctive syllogism [AB75]. Later Tennant [Ten84] suggested the retention of disjunctive syllogism, and the rejection of transitivity.

Tennant's paraconsistent logic is a restriction of the classical system of natural deduction: Classical proofs are restricted so as to prohibit applications of ex falso quodlibet. This gives a system in which (1) the Lewis paradoxes do not hold; (2) every unsatisfiable set of sentences can be shown to be inconsistent by the logic; (3) all non-tautological classical consequences of a set of sentences are provable in the logic; and (4) transitivity fails only when the combined assumptions form an inconsistent set.

There is a clear similarity between Tennant's paraconsistent logic and QC logic. Though, an immediate difference is that tautologies, such as $\alpha \vee \neg\alpha$, follow from the empty set using Tennant's logic whereas no tautologies hold for QC logic.

Other properties that fail for the QC consequence relation include supraclassicality, conditionalization, deduction, right weakening, and left logical equivalence. These failures are substantially due to classical tautologies not holding in general for QC logic. We compare QC logic with further paraconsistent logics in Section 8.

5 Full semantics for QC logic

We now extend the semantics presented in Section 3.

Definition 5.1 *Let $\models_s$ be the strong satisfiability relation defined in Definition 3.2 where $\models_s \subseteq \wp(\mathcal{O}) \times \mathcal{L}$. For $X \in \wp(\mathcal{O}), \alpha, \beta, \gamma \in \mathcal{L}$, we extend the definition as follows,*

$$X \models_s \alpha \wedge \beta \text{ iff } X \models_s \alpha \text{ and } X \models_s \beta$$

$$X \models_s \neg\neg\alpha \vee \gamma \text{ iff } X \models_s \alpha \vee \gamma$$

$$X \models_s \neg(\alpha \wedge \beta) \vee \gamma \text{ iff } X \models_s \neg\alpha \vee \neg\beta \vee \gamma$$

$$X \models_s \neg(\alpha \vee \beta) \vee \gamma \text{ iff } X \models_s (\neg\alpha \wedge \neg\beta) \vee \gamma$$

$$X \models_s \alpha \vee (\beta \wedge \gamma) \text{ iff } X \models_s (\alpha \vee \beta) \wedge (\alpha \vee \gamma)$$

$$X \models_s \alpha \wedge (\beta \vee \gamma) \text{ iff } X \models_s (\alpha \wedge \beta) \vee (\alpha \wedge \gamma)$$

$$X \models_s (\alpha \rightarrow \beta) \vee \gamma \text{ iff } X \models_s \neg\alpha \vee \beta \vee \gamma$$

$$X \models_s \neg(\alpha \rightarrow \beta) \vee \gamma \text{ iff } X \models_s (\alpha \wedge \neg\beta) \vee \gamma$$

Definition 5.2 *Let $\models_w$ be the weak satisfiability relation defined in Definition 3.3 where $\models_w \subseteq \wp(\mathcal{O}) \times \mathcal{L}$. For $X \in \wp(\mathcal{O}), \alpha, \beta, \gamma \in \mathcal{L}$, we extend the definition* *as follows,*

$$X \models_w \alpha \wedge \beta \text{ iff } X \models_w \alpha \text{ and } X \models_w \beta$$

$$X \models_w \alpha \vee \beta \text{ iff } X \models_w \alpha \text{ or } X \models_w \beta$$

$$X \models_w \neg\neg\alpha \vee \gamma \text{ iff } X \models_w \alpha \vee \gamma$$

$$X \models_w \neg(\alpha \wedge \beta) \vee \gamma \text{ iff } X \models_w \neg\alpha \vee \neg\beta \vee \gamma$$

$$X \models_w \neg(\alpha \vee \beta) \vee \gamma \text{ iff } X \models_w (\neg\alpha \wedge \neg\beta) \vee \gamma$$

$$X \models_w (\alpha \rightarrow \beta) \vee \gamma \text{ iff } X \models_w \neg\alpha \vee \beta \vee \gamma$$

$$X \models_w \neg(\alpha \rightarrow \beta) \vee \gamma \text{ iff } X \models_w (\alpha \wedge \neg\beta) \vee \gamma$$

The difference between between Definition 5.1 and Definition 5.2 is that Definition 5.1 includes axioms for distribution whereas Definition 5.2 includes an axiom for disjunction. As discussed in Section 3, strong satisfaction is much more restricted than weak satisfaction with regard to disjunction, and as can be seen from Proposition 5.1, distribution axioms are not required explicitly in Definition 5.2.

Definition 5.3 $\models_Q$ *is an entailment relation, called the QC entailment relation, iff* $\models_Q \subseteq \wp(\mathcal{L}) \times \mathcal{L}$, *and*

$$\{\alpha_1, .., \alpha_n\} \models_Q \beta$$
$$\text{iff for all } X \text{ if } X \models_s \alpha_1 \text{ and ... and } X \models_s \alpha_n \text{ then } X \models_w \beta$$

Example 5.1 *Let* $\Delta = \{\neg(\alpha \wedge \beta)\}$, *and let* $X_1 = \{-\alpha, -\beta\}$ *and* $X_2 = \{+\alpha, -\alpha\}$. *Now* $X_1 \models_s \neg\alpha$, $X_1 \models_s \neg\beta$, $X_2 \models_s \neg\alpha$, *and* $X_2 \not\models_s \neg\beta$. *For this* $X_1 \models_s \neg\alpha \wedge \neg\beta$, *and hence* $X_1 \models_s \neg(\alpha \vee \beta)$, *whereas* $X_2 \not\models_s \neg\alpha \wedge \neg\beta$, *and hence* $X_2 \not\models_s \neg(\alpha \vee \beta)$,

Example 5.2 *Let* $\Delta = \{\alpha \wedge \neg\alpha\}$, *where* $\alpha \in \mathcal{A}$. *For all models* X, $X \models_s \alpha \wedge \neg\alpha$, *iff* $X \models_s \alpha$ *and* $X \models_s \neg\alpha$ *iff* $+\alpha \in X$ *and* $-\alpha \in X$. *Hence,* $\Delta \models_Q \alpha \wedge \neg\alpha$

Proposition 5.1 *For any* $\alpha, \beta, \gamma \in \mathcal{L}$, *and any* $X \in \wp(\mathcal{O})$, *the following distribution properties hold.*

$$X \models_w \alpha \vee (\beta \wedge \gamma) \text{ iff } X \models_w (\alpha \vee \beta) \wedge (\alpha \vee \gamma)$$

$$X \models_w \alpha \wedge (\beta \vee \gamma) \text{ iff } X \models_w (\alpha \wedge \beta) \vee (\alpha \wedge \gamma)$$

Proof Assume $X \models_w \alpha \vee (\beta \wedge \gamma)$. So $X \models_w \alpha$ or ($X \models_w \beta$ and $X \models_w \gamma$). By distributivity of the classical connectives "or" and "and", we have ($X \models_w \alpha$ or $X \models_w \beta$) and ($X \models_w \alpha$ or $X \models_w \gamma$). Hence. $X \models_w (\alpha \vee \beta) \wedge (\alpha \vee \beta)$. The rest of the proposition follows similarly. □

We now show QC proof theory is sound.

Lemma 5.1 *Let β,$\neg\beta$,$\alpha_1, .., \alpha_m$,$\gamma_1, .., \gamma_n$ be literals. For clauses $\beta \vee \alpha_1 \vee .. \vee \alpha_m$, $\neg\beta \vee \gamma_1 \vee .. \vee \gamma_n$ in $\mathcal{L}$, and for all $X \in \wp(\mathcal{O})$, the following holds.*

If $X \models_s \beta \vee \alpha_1 \vee .. \vee \alpha_m$ and $X \models_s \neg\beta \vee \gamma_1 \vee .. \vee \gamma_n$
then $X \models_s \alpha_1 \vee .. \vee \alpha_m \vee \gamma_1 \vee .. \vee \gamma_n$

Proof Assume (A1) $X \models_s \beta \vee \alpha_1 \vee .. \vee \alpha_m$; (A2) $X \models_s \neg\beta \vee \gamma_1 \vee .. \vee \gamma_n$; and (A3) $X \not\models_s \alpha_1 \vee .. \vee \alpha_m \vee \gamma_1 \vee .. \vee \gamma_n$. We give a proof by contradiction. First consider the following iterative scheme (B_i), where Γ_i denotes a disjunction of literals.

$X \not\models_s \Gamma_i$ implies

(C_i) for all disjuncts ψ in Γ_i $X \not\models_s \psi$

or

(D_i) there is a disjunct ϕ_i in Γ_i such that
[$X \models_s \sim\phi_i$ and $X \not\models_s Focus(\Gamma_i, \phi_i)$]

Let $\Gamma_{i+1} = Focus(\Gamma_i, \phi_i)$

This scheme follows directly from Definition 3.2. For the first iteration B_1, Γ_1 is $\alpha_1 \vee .. \vee \alpha_m \vee \gamma_1 \vee .. \vee \gamma_n$. For this, C_1 contradicts with assumptions A1 and A2. D_1 gives $X \models_s \sim\phi_1$ for some disjunct ϕ_1 in Γ_1, and causes an iteration where Γ_2 is $Focus(\Gamma_1, \phi_1)$. For C_2, there is a contradiction with A1, A2 and the statement $X \models_s \sim\phi_1$, where ϕ_1 was instantiated as one of $\{\alpha_1, .., \alpha_m, \gamma_1, .., \gamma_n\}$ in D_1. As for D_1, D_2 gives $X \models_s \sim \phi_2$ for some ϕ_2 in Γ_2, and causes an iteration B_3 where Γ_3 is $Focus(\Gamma_2, \phi_2)$. For each C_i where $i > 2$, there is a contradiction with A1, A2 and the statements $X \models_s \sim \phi_1$, ..., $X \models_s \sim \phi_{i-1}$ (which were instantiated in the previous iterations). For each D_i, there is an iteration B_i. The iterations of (B_i) continue until there is an $i = m+n$ where Γ_{m+n} is a literal and $X \not\models_s \Gamma_{m+n}$. But there is a contradiction with $A1, A2$, and the statements $X \models_s \sim\phi_1$, ..., $X \models_s \sim\phi_{i-1}$ There are no other possible branches to consider, and so the assumption A3 is false. □

Example 5.3 *To illustrate the nature of Lemma 5.1, consider the following special case of it.*

$$\frac{X \models_s \alpha \vee \beta \qquad X \models_s \neg\beta \vee \gamma}{X \models_s \alpha \vee \gamma}$$

The assumption $X \models_s \alpha \vee \beta$ implies: (1) $X \models_s \alpha$ or $X \models_s \beta$; (2) $X \models_s \neg\alpha$ implies $X \models_s \beta$; and (3) $X \models_s \neg\beta$ implies $X \models_s \alpha$. The assumption $X \models_s \neg\beta \vee \gamma$ implies: (4) $X \models_s \neg\beta$ or $X \models_s \gamma$; (5) $X \models_s \beta$ implies $X \models_s \gamma$; and (6) $X \models_s \neg\gamma$ implies $X \models_s \neg\beta$. Now (3)+(6) gives $X \not\models_s \neg\gamma$ or $X \models_s \alpha$, and (2)+(5) gives $X \not\models_s \neg\alpha$ or $X \models_s \gamma$, and (1)+(5) gives $X \models_s \alpha$ or $X \models_s \gamma$. So (3)+(6) plus (2)+(5) plus (1)+(5) gives $X \models_s \alpha \vee \gamma$.

An alternative proof for Lemma 5.1 can be straightforwardly obtained from Proposition 3.1.

Lemma 5.2 *For $\Delta \in \wp(\mathcal{L})$, and a clause $\gamma_i \in \mathcal{L}$, if there is a decomposition of γ_i from Δ, then $\Delta \models_Q \gamma_i$ holds.*

Proof Let $\Delta = \{\beta_1, ..., \beta_n\}$. Assume a model X such that $X \models_s \beta_1$, ..., $X \models_s \beta_n$. So we need to show $X \models_s \gamma_i$, and hence $X \models_w \gamma_i$. For this, we need to show that each proof rule in the decomposition tree for γ_i from $\{\beta_1, ..., \beta_n\}$ is sound. In other words, for each premise π and each consequent τ, of each decomposition proof rule, if $X \models_s \pi$, then $X \models_s \tau$. This is immediate for conjunction elimination, disjunction contraction, decompositional distribution, negation elimination, arrow elimination, and decompositional de Morgan laws. This leaves resolution. Since we can assume a normal decomposition tree, by Proposition 4.9, resolution is only applied to clauses, and so by Lemma 5.1, resolution is sound. □

Lemma 5.3 *For a set of clauses $\{\gamma_1, ..., \gamma_n\} \in \wp(\mathcal{L})$, and a formula $\beta \in \mathcal{L}$, assume there is a composition of β from $\{\gamma_1, ..., \gamma_n\}$. If $X \models_w \gamma_1$ and .. and $X \models_w \gamma_n$ hold, then $X \models_w \beta$*

Proof Assume a model X such that $X \models_w \gamma_1$, ..., $X \models_w \gamma_n$. So we need to show $X \models_w \beta$. For this, we need to show that each proof rule in the composition tree for β from $\{\gamma_1, ..., \gamma_n\}$ is sound. In other words, for each premise π and each consequent τ, of each composition proof rule, if $X \models_w \pi$, then $X \models_w \tau$. This is immediate for all the composition proof rules. □

Proposition 5.2 *For $\Delta \in \wp(\mathcal{L})$, and $\alpha \in \mathcal{L}$, if $\Delta \vdash_Q \alpha$, then $\Delta \models_Q \alpha$.*

Proof Follows directly from Lemma 5.2 and Lemma 5.3.

We now show QC proof theory is complete.

Lemma 5.4 *For $X \in \wp(\mathcal{O})$, $\psi, \phi \in \mathcal{L}$, if ψ is a CNF of ϕ, then the following equivalences hold.*

$$X \models_s \psi \text{ iff } X \models_s \phi$$

$$X \models_w \psi \text{ iff } X \models_w \phi$$

Lemma 5.5 *For $X \in \wp(\mathcal{O})$, $\beta, \gamma \in \mathcal{L}$,*

$$X \models_s \beta \text{ iff } \forall \gamma \in Resolvents(\{\beta\}) X \models_s \gamma.$$

Proof ($\Rightarrow$) Follows from Lemma 5.4 and Lemma 5.1. ($\Leftarrow$) Since $Clauses(\{\beta\}) \subseteq Resolvents(\{\beta\})$, by Lemma 5.4, $X \models_s \beta$. $\Box$

Definition 5.4 *Let $\Delta \in \wp(\mathcal{L})$, where $\Delta = \{\beta_1, ..., \beta_n\}$.*

$$M(\Delta) = \{X \in \wp(\mathcal{O}) \mid X \models_s \beta_1 \wedge .. \wedge \beta_n\}$$

$$W(\Delta) = \{\alpha \in \mathcal{L} \mid \forall X \in M(\Delta)\ X \models_w \alpha\}$$

So $M(\Delta)$ is the set of models that strongly satisfy Δ, and $W(\Delta)$ is the set of formulae that are weakly satisfied in all models in $M(\Delta)$.

Lemma 5.6 *Let $\Delta \in \wp(\mathcal{L})$. $W(\Delta)$ is the closure of $Resolvents(\Delta)$ under the following equivalences:*

(E1) $\alpha \vee \beta \in W(\Delta)$ *iff* $\alpha \in W(\Delta)$ *or* $\beta \in W(\Delta)$
(E2) $\alpha \wedge \beta \in W(\Delta)$ *iff* $\alpha \in W(\Delta)$ *and* $\beta \in W(\Delta)$
(E3) $\neg\neg\alpha \vee \beta \in W(\Delta)$ *iff* $\alpha \vee \beta \in W(\Delta)$
(E4) $\neg(\alpha \wedge \beta) \vee \gamma \in W(\Delta)$ *iff* $\neg\alpha \vee \neg\beta \vee \gamma \in W(\Delta)$
(E5) $\neg(\alpha \vee \beta) \vee \gamma \in W(\Delta)$ *iff* $(\neg\alpha \wedge \neg\beta) \vee \gamma \in W(\Delta)$
(E6) $(\alpha \rightarrow \beta) \vee \gamma \in W(\Delta)$ *iff* $\neg\alpha \vee \beta \vee \gamma \in W(\Delta)$
(E7) $\neg(\alpha \rightarrow \beta) \vee \gamma \in W(\Delta)$ *iff* $(\alpha \wedge \neg\beta) \vee \gamma \in W(\Delta)$

Proof Since $M(\Delta) = M(Resolvents(\Delta))$ by Lemma 5.5, we can substitute $M(Resolvents(\Delta))$ in the definition for $W(\Delta)$, so $W(\Delta) = W(Resolvents(\Delta))$. Hence, for all $\alpha \in Resolvents(\Delta)$, $\alpha \in W(\Delta)$. Furthermore, by Definition 5.2, the only further formulae in $W(\Delta)$ are those given by the equivalences for the $\models_w$ relation. $\Box$

Definition 5.5 *Let $\Delta \in \wp(\mathcal{L})$, and let $\Phi \in \wp(\mathcal{L})$, where Φ is a set of clauses.*

$$C(\Phi) = \{\alpha \in \mathcal{L} \mid \exists \Psi \subseteq \Phi \text{ such that there is a composition of } \alpha \text{ from } \Psi\}$$

$$D(\Delta) = \{\gamma \in \mathcal{L} \mid \text{ there is a decomposition of } \gamma \text{ from } \Delta\}$$

Lemma 5.7 *For any $\Delta \in \wp(\mathcal{L})$, the following equivalence holds.*

$$W(Resolvents(\Delta)) = C(Resolvents(\Delta))$$

Proof Since each semantic rewrite from clauses, given by (E1),..,(E7), is captured by a composition proof rule in the definition of $C(\Delta)$, any formula α that holds in every model of $Resolvents(\Delta)$ by weak satisfaction, is such that there is a composition of α from some subset of $Resolvents(\Delta)$. □

Proposition 5.3 *For $\Delta \in \wp(\mathcal{L})$, and $\alpha \in \mathcal{L}$, if $\Delta \models_Q \alpha$ then $\Delta \vdash_Q \alpha$.*

Proof Let $\Delta = \{\beta_1, .., \beta_n\}$.

$\Delta \models_Q \alpha$	
implies $\forall X[X \models_s \beta_1 \wedge .. \wedge \beta_n$ implies $X \models_w \alpha]$	by Definition 5.3
implies $\forall X \in M(\Delta)\ X \models_w \alpha$	by Definition 5.4
implies $\alpha \in W(\Delta)$	by Definition 5.4
implies $\alpha \in W(Resolvents(\Delta))$	by Lemma 5.6
implies $\alpha \in C(Resolvents(\Delta))$	by Lemma 5.7
implies $\alpha \in C(D(\Delta))$	by Proposition 4.3
implies $\Delta \vdash_Q \alpha$	by Definitions 4.7 and 5.5 □

We now consider two complexity results.

Proposition 5.4 *Let $\Delta \in \wp(\mathcal{L})$. Checking whether Δ has a model takes constant time.*

Proof Assume $\Delta = \{\beta_1, .., \beta_n\}$. For all $\alpha \in \mathcal{L}$, there is an $X \in \wp(\mathcal{O})$ such that $X \models_s \alpha$. Hence, there is an $X \in \wp(\mathcal{O})$ such that $X \models_s \beta_1 \wedge .. \wedge \beta_n$, and so every $\Delta \in \wp(\mathcal{L})$ has a model. □.

Proposition 5.5 *Let $\Delta \in \wp(\mathcal{L})$ be a set of clauses. Identifying an $X \in \wp(\mathcal{O})$, such that for all $\alpha \in \Delta, X \models_s \alpha$, takes time linear in the number of disjuncts in the set of clauses.*

Proof First form $Resolvents(\Delta)$. Since resolution can be applied at most $n/2$ times, where n is the number of disjuncts in the set of clauses, this takes time linear in n. Now make a disjunct in each clause in $Resolvents(\Delta)$ hold in the model. Since, there at most n clauses $Resolvents(\Delta)$, this also takes time linear in n. □.

6 Comparison with other approaches

QC logic exhibits the nice feature that no attention needs to be paid to a special form that the formulae in a set of premises should have, as long as each formula in the set is individually consistent and not a tautology. This is in contrast with other paraconsistent logics where two formulae identical by definition of a connective in classical logic may not yield the same set of conclusions. From an example given earlier in this paper, in QC logic, β is a conclusion of both $\{(\neg\alpha \rightarrow \beta), \neg\alpha\}$ and $\{\alpha \vee \beta, \neg\alpha\}$. QC logic is much better behaved in this respect than other paraconsistent logics.

In this section, we compare QC logic with C_ω logic and four-valued logic. Below we give a presentation of C_ω which was proposed da Costa [dC74]. All the schemata in the logic C_ω are schemata in classical logic.

Definition 6.1 *The logic C_ω is defined by the following axiom schemata together with the modus ponens proof rule.*

$$\alpha \rightarrow (\beta \rightarrow \alpha)$$
$$(\alpha \rightarrow \beta) \rightarrow ((\alpha \rightarrow (\beta \rightarrow \gamma)) \rightarrow (\alpha \rightarrow \gamma))$$
$$\alpha \wedge \beta \rightarrow \alpha$$
$$\alpha \wedge \beta \rightarrow \beta$$
$$\alpha \rightarrow (\beta \rightarrow \alpha \wedge \beta)$$
$$\alpha \rightarrow \alpha \vee \beta$$
$$\beta \rightarrow \alpha \vee \beta$$
$$(\alpha \rightarrow \gamma) \rightarrow ((\beta \rightarrow \gamma) \rightarrow (\alpha \vee \beta \rightarrow \gamma))$$
$$\alpha \vee \neg\alpha$$
$$\neg\neg\alpha \rightarrow \alpha$$

As with QC logic, the properties of reflexivity, and, consistency preservation, or, and monotonicity hold for the C_ω relation. However, cut, conditionalization, and deduction, also hold for C_ω [Hun96]. For both QC and C_ω, supraclassicality, left logical equivalence, and right weakening fail.

The four-valued logic of Belnap [Bel77] provides an interesting alternative to C_ω in that it has an illuminating and intuitive semantic characterization to complement its proof theory. We just provide the proof theory here.

Definition 6.2 *The language for four-valued logic is a subset of classical logic. Let $\mathcal{P}$ be the usual set of formulae of classical logic that is formed using the connectives $\neg, \wedge$ and $\vee$. Then the set of formulae of the language, denoted Q, is $\mathcal{P} \cup \{\alpha \rightarrow \beta \mid \alpha, \beta \in \mathcal{P}\}$, and hence implication is not nestable.*

Definition 6.3 *Let $\alpha, \beta, \gamma \in \mathcal{L}$. The following are the proof rules for the four-valued consequence relation. Let $\alpha \leftrightarrow \beta$ signify that α and β are semantically equivalent, and can be intersubstituted in any context.*

$$\alpha_1 \wedge ... \wedge \alpha_m \rightarrow \beta_1 \vee ... \vee \beta_n \textit{ provided some } \alpha_i \textit{ is some } \beta_j$$
$$(\alpha \vee \beta) \rightarrow \gamma \textit{ iff } \alpha \rightarrow \gamma \textit{ and } \beta \rightarrow \gamma$$
$$\alpha \rightarrow (\beta \wedge \gamma) \textit{ iff } \alpha \rightarrow \beta \textit{ and } \alpha \rightarrow \gamma$$
$$\alpha \rightarrow \beta \textit{ iff } \neg\beta \rightarrow \neg\alpha$$
$$\alpha \rightarrow \beta \textit{ and } \beta \rightarrow \gamma \textit{ implies } \alpha \rightarrow \gamma$$
$$\alpha \rightarrow \beta \textit{ iff } \alpha \leftrightarrow (\alpha \wedge \beta) \textit{ iff } \beta \leftrightarrow (\alpha \vee \beta)$$

In addition, the following extends the definition of the four-valued consequence relation.

$$\alpha \vee \beta \leftrightarrow \beta \vee \alpha$$
$$\alpha \wedge \beta \leftrightarrow \beta \wedge \alpha$$
$$\alpha \vee (\beta \vee \gamma) \leftrightarrow (\alpha \vee \beta) \vee \gamma$$
$$(\alpha \wedge \beta) \wedge \gamma \leftrightarrow \alpha \wedge (\beta \wedge \gamma)$$
$$\alpha \wedge (\beta \vee \gamma) \leftrightarrow (\alpha \wedge \beta) \vee (\alpha \wedge \gamma)$$
$$\alpha \vee (\beta \wedge \gamma) \leftrightarrow (\alpha \vee \beta) \wedge (\alpha \vee \gamma)$$
$$\neg\neg\alpha \leftrightarrow \alpha$$
$$\neg(\alpha \wedge \beta) \leftrightarrow \neg\alpha \vee \neg\beta$$
$$\neg(\alpha \vee \beta) \leftrightarrow \neg\alpha \wedge \neg\beta$$
$$\alpha \leftrightarrow \beta \textit{ and } \beta \leftrightarrow \gamma \textit{ implies } \alpha \leftrightarrow \gamma$$

.

As with QC logic, reflexivity, consistency preservation, and monotonicity hold for the four-valued consequence relation [Hun96]. In addition, cut holds for four-valued logic. However, four-valued logic does not adhere to and-introduction, supraclassicality, or-elimination, left logicial equivalence, deduction, conditionalization, or right weakening.

The QC consequence relation offers many more non-tautological inferences from data than either the weakly-negative or four-valued logics. For example, via disjunctive syllogism, QC logic gives β from $\{\neg\alpha, \alpha \vee \beta\}$, whereas neither the weakly-negative logic C_ω nor four-valued logic gives β.

Proposition 6.1 *For $\Delta \in \wp(\mathcal{L})$, let $C_\omega(\Delta)$ denote the set of C_ω consequences from Δ, and let $C_Q(\Delta)$ denote the set of QC consequences from Δ. For this $C_Q(\Delta) \not\subseteq C_\omega(\Delta)$ and $C_\omega(\Delta) \not\subseteq C_Q(\Delta)$ hold.*

Proof For any Δ, $\alpha \vee \neg\alpha$ is in $C_\omega(\Delta)$, but not in $C_Q(\Delta)$. For $\Delta = \{\neg\alpha, \alpha \vee \beta\}$, β is in $C_Q(\Delta)$, but not in $C_\omega(\Delta)$. □

This proposition follows from the classical tautologies from the empty set not being derivable in QC logic. However, if we exclude consideration of these tautologies, then we see that QC logic is stronger than C_ω.

Proposition 6.2 *For $\Delta \in \wp(\mathcal{Q})$, let $C_{FV}(\Delta)$ denote the set of four-valued consequences from Δ, and let $C_Q(\Delta)$ denote the set of QC consequences from Δ. For this $C_Q(\Delta) \not\subseteq C_{FV}(\Delta)$ and $C_{FV}(\Delta) \not\subseteq C_Q(\Delta)$ hold.*

Proof For any Δ, $\alpha \vee \neg\alpha$ is in $C_{FV}(\Delta)$, but not in $C_Q(\Delta)$. For $\Delta = \{\neg\alpha, \alpha \vee \beta\}$, β is in $C_Q(\Delta)$, but not in $C_{FV}(\Delta)$. □

QC logic can also be more appropriate than various approaches to reasoning from consistent subsets of inconsistent sets of formulae (for example consistency-based logics [BDP93, EGH95] and truth maintenance systems [Doy79, Kle86, MS88]). In particular, QC logic does not suffer from the limitation due to splitting sets of formulae into compatible subsets: QC logic can make use of the contents of the formulas without being constrained by a consistency check. Moreover, it is obviously an advantage of QC logic to dispense with the costly consistency checks that are needed in all approaches to reasoning from consistent subsets.

Whilst QC logic constitutes an interesting alternative to other paraconsistent logics for practical applications, the compromises include tautologies from an empty set of assumptions being non-derivable, (though this is not usually a problem for applications), and at the meta-level, ie at the level of the consequence relation, some classical properties, including transitivity, do not hold.

7 Discussion

Developing a non-trivializable, or paraconsistent logic, necessitates some compromise, or weakening, of classical logic. The compromises imposed to give QC logic seem to be more appropriate than other paraconsistent logics for applications in computing. QC logic provides a means to obtain all the non-trivial resolvents from a set of formulae, without the problem of trivial clauses also following.

QC logic is being developed for applications — in particular, for reasoning about requirements specifications that might be inconsistent [HN97, HN98]. QC logic may also offer an interesting alternative to classical logic in non-monotonic reasoning. Consider default logic with a default theory (D, W). Here the default rules in D are used to extend the classical reasoning from W under a proviso that the extension is consistent. Now, if QC reasoning is used, instead of classical reasoning, then an alternative mechanism and interpretation of default reasoning is possible.

Acknowledgements

The author wishes to thank the anonymous referees for many valuable suggestions for improving the paper, and to thank an anonymous referee of a related paper [HN98] for conjecturing Proposition 3.1.

References

[AB75] A Anderson and N Belnap. *Entailment: The Logic of Relevance and Necessity*. Princeton University Press, 1975.

[BdCGH97] Ph Besnard, L Farinas del Cerro, D Gabbay, and A Hunter. Logical handling of default and inconsistent information. In Ph Smets and A Motro, editors, *Uncertainty Management in Information Systems*. Kluwer, 1997.

[BDP93] S Benferhat, D Dubois, and H Prade. Argumentative inference in uncertain and inconsistent knowledge bases. In *Proceedings of the 9th Conference on Uncertainty in Artificial Intelligence*, pages 411–419. Morgan Kaufmann, 1993.

[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 C Froidevaux and J Kohlas, editors, *Symbolic and Quantitative Approaches to Uncertainty*, volume 946 of *Lecture Notes in Computer Science*, pages 44–51, 1995.

[BH98] Ph Besnard and A Hunter. Introduction to actual and potential contradictions. In *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, volume 2, pages 1–9. Kluwer, 1998.

[CH97] L Cholvy and A Hunter. Information fusion in logic: A brief overview. In D Gabbay, R Kruse, A Nonnengart, and H-J Ohlbach, editors, *Qualitative and Quantitative Practical Reasoning*, volume 1244 of *Lecture Notes in Computer Science*. Springer, 1997.

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

[Doy79] J Doyle. A truth maintenance system. *Artificial Intelligence*, 12:231–272, 1979.

[EGH95] M Elvang-Gøransson and A Hunter. Argumentative logics: Reasoning from classically inconsistent information. *Data and Knowledge Engineering*, 16:125–145, 1995.

[FGH+94] A Finkelstein, D Gabbay, A Hunter, J Kramer, and B Nuseibeh. Inconsistency handling in multi-perspective specifications. *Transactions on Software Engineering*, 20(8):569–578, 1994.

[Gab85] D Gabbay. Theoretical foundations of non-monotonic reasoning in expert systems. In K Apt, editor, *Logics and Models of Concurrent Systems*. Springer, 1985.

[GH91] D Gabbay and A Hunter. Making inconsistency respectable 1: A logical framework for inconsistency in reasoning. In Ph Jorrand and J Kelemen, editors, *Fundamentals of Artificial Intelligence*, volume 535 of *Lecture Notes in Computer Science*, pages 19–32. Springer, 1991.

[GH93] D Gabbay and A Hunter. Making inconsistency respectable 2: Meta-level handling of inconsistent data. In M Clarke, R Kruse, and S Moral, editors, *Symbolic and Qualitative Approaches to Reasoning and Uncertainty*, volume 747 of *Lecture Notes in Computer Science*, pages 129–136. Springer, 1993.

[GM93] P Gardenförs and D Makinson. Non-monotonic inference based on expectations. *Artificial Intelligence*, 65:197–246, 1993.

[HN97] A Hunter and B Nuseibeh. Analysing inconsistent specifications. In *IEEE Third Symposium on Requirements Engineering*, pages 78–86. IEEE Computer Society Press, 1997.

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

[Hun96] A Hunter. Some results on paraconsistent logics. Technical report, Department of Computing, Imperial College, London, 1996.

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

[Kle86] J De Kleer. An assumption-based TMS. *Artificial Intelligence*, 28:127–162, 1986.

[Mak94] D Makinson. General patterns of non-montonic reasoning. In D Gabbay, C Hogger, and J Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 3, pages 35–110. Oxford Univeristy Press, 1994.

[MS88] J Martins and S Shapiro. A model of belief revision. *Artificial Intelligence*, 35:25–79, 1988.

[Tar56] A Tarski. *Logic, Semantics, Metamathematics*. Oxford University Press, 1956.

[Ten84] N Tennant. Perfect validity, entailment, and paraconsistency. *Studia Logica*, 43:179–198, 1984.