Computational Aspects of Quasi-Classical Entailment

Pierre Marquis
CRIL/Université d'Artois
62307 Lens Cedex
France
marquis@cril.univ-artois.fr

Nadège Porquet
CRIL/Université d'Artois
62307 Lens Cedex
France
porquet@cril.univ-artois.fr

ABSTRACT. Quasi-classical logic is a propositional logic for reasoning under inconsistency pointed out recently in the literature [3] [21]. Compared with several other paraconsistent logics, it has the nice feature that no special attention needs to be paid to a special form of premises (the connectives behave in a classical manner). However, only few is known about its computational behaviour up to now. In this paper, we fill this gap by pointing out a linear time translation that maps every instance of the quasi-classical entailment problem for CNF formulas into an instance of the classical entailment problem. This enables us to (1) show that quasi-classical entailment for CNF formulas is "only" coNP*-complete, (2) take advantage of all existing automated reasoning techniques for classical entailment in order to improve quasi-classical inference from the practical side. Especially, tractable classes of formulas for quasi-classical entailment are pointed out.*

1 Introduction

Handling inconsistencies is an important issue for practical reasoning from actual, large and complex, knowledge bases. While classical logic is an unmatched formalism for reasoning from consistent knowledge bases, it suffers from trivialization in presence of contradiction. Indeed, every formula γ is considered as a classical consequence of an inconsistent knowledge base Σ (this is referred to as *ex falso quodlibet sequitur*). Since this is far from satisfying, a variety of approaches to handle inconsistency have been pointed out so far (see [4] for a survey). Some of them aim at *preventing* inconsistencies from being generated;

for instance, belief revision aims at avoiding any contradiction to occur when a consistent piece of information is incorporated into a consistent base. Other approaches aim at *removing* inconsistencies, for instance by performing tests (knowledge-gathering actions) so as to find erroneous pieces of information [22]. Complementary to them, many approaches *deal with* inconsistencies; let us mention, among others, the coherence-based approaches [32] [2] (some preferred consistent subbases of Σ are selected, then inference amounts to classical inference from some of them) and various paraconsistent logics [20] (a subset of the classical proofs are retained in order to avoid trivialization).

In this paper, we focus on *quasi-classical logic*, a propositional logic for reasoning under inconsistency pointed out recently in the literature [3] [21]. In this logic, only a subset of the classical proofs is considered: those for which the application of an elimination rule never follows the application of an introduction rule. Accordingly, quasi-classical logic is *infraclassical*: every formula γ that is quasi-classically entailed by a knowledge base Σ is a logical consequence of it; stated otherwise, quasi-classical entailment $\models_{QC}$ viewed as a relation over formulas is weaker than classical entailment $\models$: we have $\models_{QC} \subseteq \models$. The converse does not hold in the general case, especially when the knowledge base Σ is inconsistent. In this situation, while $\Sigma \models \gamma$ holds for every formula γ, we do **not** have $\Sigma \models_{QC} \gamma$ for every γ. Trivialization is avoided, and quasi-classical logic is *paraconsistent*.

Compared with several other paraconsistent logics (like C_ω logic [9]), quasi-classical logic has the nice feature that no special attention needs to be paid to a special form of premises (the connectives behave in a classical manner). Indeed, every knowledge base (as a set of formulas) is interpreted as the conjunction of its elements in quasi-classical logic; this deeply contrasts with coherence-based approaches to inconsistency handling in the general case. Accordingly, we could say that quasi-classical logic is (loosely speaking) less syntax-sensitive than many challenging approaches[1]. Anyway, the fact that connectives behave classically is a good point since it enables an easy grasp of the contents of a knowledge base, and permits it to be normalized in various ways (and this is important in practice from a computational point of view).

All these features show quasi-classical logic as a valuable candidate for reasoning under inconsistency. However, only few is known about its computational behaviour up to now. In informal terms, [20] states that *"it is obviously an advantage of quasi-classical logic to dispense with the costly consistency checks that are needed in all approaches to reasoning from consistent subsets"* but the exact complexity of deciding quasi-classical entailment is not identified. Actually, at that time, quasi-classical entailment is only known as decidable (Proposition 4.8 in [21]). Of course, this is not sufficient to adhere to this

[1] Clearly enough, every valuable approach to deal with inconsistency within a knowledge base is syntax-sensitive since if it were not the case, every inconsistent Σ and the knowledge base $\{x \wedge \neg x\}$ would have the same consequences (so the approach would be useless). What is counter-intuitive here is that the notion of independence of the syntax refers to classical semantics, which is definitely not suited to the inconsistency situation.

logic for practical situations; as advocated by many AI researchers (following the seminal work of [6]), the computational dimension of inference cannot be neglected when applications are to be considered!

In this paper, we contribute to fill this gap by pointing out a linear time translation that maps every instance of the quasi-classical entailment problem for CNF formulas into an instance of the classical entailment problem.

On the one hand, this enables us to show that deciding quasi-classical entailment for CNF formulas is "only" coNP-complete. Because deciding inference relations in coherence-based approaches is typically harder (often at the second level of the polynomial hierarchy) [30], this is a decisive advantage of quasi-classical logic over such approaches.

On the other hand, our translation to classical entailment enables us to take advantage of all sophisticated automated reasoning techniques for classical entailment in order to improve quasi-classical clausal inference. Specifically, we show that two of the major tractable classes of formulas for classical entailment (namely, Krom formulas and renamable Horn k-CNF formulas) are tractable as well for quasi-classical entailment: deciding whether a given CNF formula is a quasi-classical consequence of a knowledge base that belongs to one of these two classes can be achieved in linear time.

The rest of this paper is organized as follows. In Section 2, some formal preliminaries are given. In Section 3, quasi-classical logic is presented. In Section 4, our translation from quasi-classical entailment for CNF formulas to classical entailment is presented; as a by-product, the complexity of deciding quasi-classical entailment is identified. In Section 5, we show how the translation can be exploited to mechanize quasi-classical inference in the general case and in some restricted cases. Section 6 concludes the paper. Proofs of propositions are reported in an appendix.

2 Formal Preliminaries

Given a denumerable set PS of propositional symbols, $PROP_{PS}$ is the propositional language built up from PS and the connectives $\neg$, $\vee$, $\wedge$ in the usual way. $\Rightarrow$ can also be introduced as a connective, where $\alpha \Rightarrow \beta$ is a short for $\neg\alpha \vee \beta$. The boolean constants $true$ and $false$ are not explicitly given in the language. L_{PS} is the set of all literals that can be generated from symbols of PS. For every formula Σ of $PROP_{PS}$, $Var(\Sigma)$ is the set of propositional symbols of PS that occur in Σ, and $|\Sigma|$ is the size of Σ (the number of symbols used to write it).

A knowledge base is a finite, conjunctively-interpreted, set of formulas from $PROP_{PS}$. A NNF (Negation Normal Form) formula is a formula from $PROP_{PS}$ s.t. the scope of any occurrence of $\neg$ in the formula is a propositional symbol. A literal is a propositional symbol from PS (positive literal) or a negated one (negative literal). A clause is a finite and **non empty** disjunction of literals from L_{PS}, also viewed as the set of its literals when it is convenient. A positive clause contains only positive literals. A CNF formula is a (finite) conjunction of clauses, also viewed as a set of clauses when it is convenient. A k-CNF formula is a CNF formula in which every clause contains no more than k literals (where k is a non-negative integer). A Krom formula is a CNF formula in which every clause contains at most two literals. A Horn (resp. reverse Horn) CNF formula is a CNF formula in which each clause contains at most one positive (resp. negative) literal. A formula Σ is renamable Horn CNF iff there exists a substitution σ from L_{PS} to L_{PS} s.t. $\sigma(l) = l$ for every literal l of L_{PS} except those of a set L, and for every literal l of L, $\sigma(l) = \neg l$ and $\sigma(\neg l) = l$, and $\sigma(\Sigma)$ is a Horn CNF formula.

Classical interpretations I over PS are defined in the standard way, as total functions from PS to $\{0, 1\}$. Interpretations are represented by sets of variables from PS, exactly those variables that are mapped to 1 by the interpretation. As usual, $\models$ denotes classical entailment and $\equiv$ denotes classical equivalence.

Given a propositional symbol x from PS, we associate it with two new propositional symbols noted $+x$ and $-x$. We note PS^+_- the set of propositional symbols $\{+x \mid x \in PS\} \cup \{-x \mid x \in PS\}$. In order to simplify notations, for every literal $l \in L_{PS}$, we use $+l$ as a short for $+x$ when $l = x$ is a positive literal, and $+l$ as a short for $-x$ when $l = \neg x$ is a negative literal; dually, we use $-l$ as a short for $-x$ when $l = x$ is a positive literal, and $-l$ as a short for $+x$ when $l = \neg x$ is a negative literal.

We assume that the reader is familiar with basic notions of complexity theory, especially NP-completeness (see [19] [31] for details).

3 Quasi-Classical Logic

As evoked in the introduction, there are many ways to prevent inference from trivialization in presence of inconsistency. One of them consists in throwing out some of the standard inference rules for classical logic, like well-known *resolution*:

$$\frac{\alpha \vee \beta, \neg\beta \vee \gamma}{\alpha \vee \gamma}\ \texttt{resolution}$$

or *disjunction introduction*:

$$\frac{\alpha}{\alpha \vee \beta}\ \texttt{disj.intro.}$$

Example 1 *Let us consider the inconsistent CNF knowledge base $\Sigma = \{a, \neg a, b, \neg b \vee c\}$. If resolution is forbidden while disjunction introduction is kept, there is no way to derive d from this CNF formula.*

Levesque's limited inference [24] relies on this idea. However, rejecting resolution is not always a good thing; on the previous example, this would prevent from generating the expected consequence c as well. A less drastic approach consists in allowing the resolution rule to be applied, except when it would produce an inconsistency [2]. Another way to avoid trivialization consists in forbidding the disjunction introduction rule. On the previous example, d cannot be generated from Σ if disjunction introduction is not allowed (even if resolution is kept). Unfortunately, the desired consequence $b \vee d$ of Σ is not reachable if disjunction introduction is systematically rejected.

Actually, none of these two inference rules is solely responsible from trivialization. Indeed, *ex falso quodlibet sequitur* is a consequence of their interaction:

$$\dfrac{\dfrac{\alpha}{\alpha \vee \beta}\ \texttt{disj.intro}, \neg\alpha}{\beta}\ \texttt{resolution}$$

Example 2 (continued) *This combination can be used to derive d as a logical consequence of Σ:*

$$\dfrac{\dfrac{a}{a \vee d}\ \texttt{disj.intro}, \neg a}{d}\ \texttt{resolution}$$

As this example suggests it, throwing out one of the two rules (resolution, disjunction introduction) would be unnecessarily destructive: it is sufficient here to restrict the way they can interact.

Based on this idea, quasi-classical logic has been pointed out in [3] and extended in [21]. In this logic, an application of the resolution rule can never follow an application of the disjunction introduction rule; moreover, applications of resolution that would produce an inconsistency (i.e., the empty clause) are rejected.

Since intuitions have now been put forward, we can present quasi-classical logic in a more formal way.

The language of quasi-classical propositional logic is $PROP_{PS}$. For the sake of simplicity, we consider only the subset of $PROP_{PS}$ consisting of **all CNF formulas** as a representation language.

Quasi-classical logic has a simple proof theory; when considering CNF formulas only, it is sufficient to consider the resolution rule and the disjunction introduction rule (disjunction and conjunction being implicitely taken as associative and commutative). Starting from a CNF formula Σ, a CNF Φ is derivable iff each clause γ of Φ is derivable, and a clause γ is derivable iff it has a resolution proof from Σ in which the empty clause does not occur and no applications of the resolution rule follow an application of the disjunction introduction rule.

[2] Another approach would consist in restricting the set of literals that can be resolved upon; the extension of Levesque's limited inference given in [33] relies on this idea; however, there is no guarantee that inconsistency is prevented this way [29].

Example 3 (continued) *Here is a proof of c from Σ:*

$$\frac{b, \neg b \vee c}{c}\ \texttt{resolution}$$

Here is a proof of $c \vee d$ from Σ:

$$\frac{\dfrac{b, \neg b \vee c}{c}\ \texttt{resolution}}{c \vee d}\ \texttt{disj.intro.}$$

Contrastingly, neither the empty clause nor d are derivable from Σ in quasi-classical logic.

Because resolution and disjunction introduction are sound inference rules (w.r.t. classical logic) but some classical proofs are not admitted any longer, quasi-classical logic is (strictly) infraclassical. Moreover, since trivialization is avoided this way, quasi-classical logic is paraconsistent.

Quasi-classical logic also has a semantics w.r.t. which the proof theory is sound and complete. In order to define the quasi-classical logic entailment relation $\models_{QC}$, we first need to make precise the underlying notion of *QC interpretation*, as well as the notions of *strong satisfaction* and *weak satisfaction*:

Definition 1 (QC interpretation) *A* QC interpretation *over PS is any subset I_{QC} of $\{+x \mid x \in PS\} \cup \{-x \mid x \in PS\}$.*

QC interpretation I_{QC} gives the following meaning to literals; let $x \in PS$:

- $+x \in I_{QC}$ means that x is "satisfied" in I_{QC};
- $-x \in I_{QC}$ means that $\neg x$ is "satisfied" in I_{QC};
- $+x \notin I_{QC}$ means that x is not "satisfied" in I_{QC};
- $-x \notin I_{QC}$ means that $\neg x$ is not "satisfied" in I_{QC}.

Accordingly, this semantics can also be regarded as giving one of the four truth values $B(oth)$, $T(rue)$, $F(alse)$, $N(either)$ to symbols from PS, as in Belnap's four valued logic [1]:

- x is B if both x and $\neg x$ are "satisfied";
- x is T if x is "satisfied" and $\neg x$ is not "satisfied";
- x is F if x is not "satisfied" and $\neg x$ is "satisfied";
- x is N if neither x nor $\neg x$ is "satisfied".

Once this is stated, one can define two notions of satisfaction of a formula by a QC interpretation I_{QC}. Because we focus on CNF formulas only, it is sufficient to our purpose to define strong and weak satisfaction for such formulas.

Definition 2 (strong satisfaction) *The notion of* strong satisfaction *of a propositional CNF formula* Σ *by a QC interpretation* I_{QC}*, noted* $I_{QC} \models_s \Sigma$*, is defined inductively as follows:*

- *If* $\alpha \in PS$*, then* $I_{QC} \models_s \alpha$ *if* $+\alpha \in I_{QC}$*;*
- *If* $\alpha \in PS$*, then* $I_{QC} \models_s \neg\alpha$ *if* $-\alpha \in I_{QC}$*;*
- $I_{QC} \models_s \alpha \wedge \beta$ *iff* $I_{QC} \models_s \alpha$ *and* $I_{QC} \models_s \beta$*;*
- $I_{QC} \models_s \alpha \vee \beta$ *iff (*$I_{QC} \models_s \alpha$ *or* $I_{QC} \models_s \beta$*) and (if* $I_{QC} \models_s \neg\alpha$ *then* $I_{QC} \models_s \beta$*) and (if* $I_{QC} \models_s \neg\beta$ *then* $I_{QC} \models_s \alpha$*).*

When $I_{QC} \models_s \Sigma$ holds, I_{QC} is said a *strong model* of Σ.

Example 4 (continued) $\{+a, -a, +b, +c\}$, $\{+a, -a, +b, -b, +c\}$, *and* $\{+a, -a, +b, -b, +c, -c\}$ *are the strong models of* $\Sigma = \{a, \neg a, b, \neg b \vee c\}$ *(over* $Var(\Sigma)$*).*

The notion of strong satisfaction can be extended to the less demanding notion of weak satisfaction defined by:

Definition 3 (weak satisfaction) *The notion of* weak satisfaction *of a propositional CNF formula* Σ *by a QC interpretation* I_{QC}*, noted* $I_{QC} \models_w \Sigma$*, is defined inductively as follows:*

- *If* $\alpha \in PS$*, then* $I_{QC} \models_w \alpha$ *if* $+\alpha \in I_{QC}$*;*
- *If* $\alpha \in PS$*, then* $I_{QC} \models_w \neg\alpha$ *if* $-\alpha \in I_{QC}$*;*
- $I_{QC} \models_w \alpha \wedge \beta$ *iff* $I_{QC} \models_w \alpha$ *and* $I_{QC} \models_w \beta$*;*
- $I_{QC} \models_w \alpha \vee \beta$ *iff* $I_{QC} \models_w \alpha$ *or* $I_{QC} \models_w \beta$*.*

When $I_{QC} \models_w \Sigma$ holds, I_{QC} is said a *weak model* of Σ. Obviously, every strong model of Σ is a weak model of Σ but the converse does not hold in the general case.

Example 5 (continued) $\{+a, -a, +b, -b\}$, $\{+a, -a, +b, -b, +c\}$, $\{+a, -a, +b, -b, -c\}$, $\{+a, -a, +b, -b, +c, -c\}$, $\{+a, -a, +b, +c\}$, *and* $\{+a, -a, +b, +c, -c\}$ *are the weak models of* $\Sigma = \{a, \neg a, b, \neg b \vee c\}$ *(over* $Var(\Sigma)$*).*

It is not very difficult to show that every CNF formula Σ has a strong model (hence a weak one as well), even if it is classically inconsistent. Indeed, $\bigcup_{x \in Var(\Sigma)} \{+x, -x\}$ is always a strong model of Σ (this is a direct consequence of Proposition 3.1 from [21]).

We are now ready to define quasi-classical entailment.

Definition 4 (QC entailment) *Let* Σ *and* Φ *be two CNF formulas from* $PROP_{PS}$*. We have* $\Sigma \models_{QC} \Phi$ *iff for every QC interpretation* I_{QC}*, if* $I_{QC} \models_s \Sigma$*, then* $I_{QC} \models_w \Phi$*.*

Example 6 (continued) *We have:*

- $\Sigma \models_{QC} c$*; $+c$ belongs to each strong model of* Σ.
- $\Sigma \models_{QC} c \vee d$ *(for the same reason as above).*
- $\Sigma \models_{QC} a$*; $+a$ belongs to each strong model of* Σ.
- $\Sigma \models_{QC} a \vee \neg a$ *(for the same reason as above).*
- $\Sigma \models_{QC} a \vee d$ *(for the same reason as above).*
- $\Sigma \models_{QC} \neg a$*; $-a$ belongs to each strong model of* Σ.
- $\Sigma \models_{QC} \neg a \vee d$ *(for the same reason as above).*
- $\Sigma \not\models_{QC} d$*; $\{+a, -a, +b, +c\}$ is a strong model of Σ over $\{a, b, c, d\}$ but $+d$ does not belong to it.*
- $\Sigma \not\models_{QC} d \vee \neg d$*; $\{+a, -a, +b, +c\}$ is a strong model of Σ over $\{a, b, c, d\}$ but neither $+d$ nor $-d$ belongs to it.*

As this example illustrates it, valid formulas are not always quasi-classical consequences of a knowledge base.

In [21], the notions of strong and weak satisfaction are extended to any formula of $PROP_{PS}$ (and not only the CNF ones). Interestingly, de Morgan laws, distribution of disjunction over conjunction and double negation elimination are satisfied in the sense that applying any of them to a formula changes neither its strong models nor its weak models (see Definitions 5.1 and 5.2, and Proposition 5.1 from [21]). Accordingly, it is always possible to turn any formula from $PROP_{PS}$ into a CNF one that has the same strong models and the same weak models. Thus, our assumption of considering only CNF formulas can be made without any loss of expressivity.

A thorough analysis of the logical properties of quasi-classical inference is reported in [21]; in a nutshell, while quasi-classical entailment satisfies reflexivity, monotonicity, and-introduction, or-elimination and consistency preservation, none of supraclassicality, right modus ponens, conditionalization, deduction, cut, unit cumulativity, right weakening and left logical equivalence are satisfied.

4 Reducing Quasi-Classical Entailment for CNF Formulas to Classical Entailment

Let us now show how to reduce quasi-classical entailment for CNF formulas to classical entailment. First of all, as a straightforward consequence of the definition of weak satisfaction, determining whether $\Sigma \models_{QC} \Phi$ holds when Φ is a CNF formula amounts to determine whether $\Sigma \models_{QC} \gamma$ holds for every clause

γ of Φ. Subsequently, from now on, we will assume that every query γ is a clause (unless stated otherwise).

Our translation from quasi-classical logic into classical logic is based on a propositional encoding of the semantics of quasi-classical logic. Clearly enough, every QC interpretation I_{QC} over PS can be viewed as a classical interpretation over PS^{+}_{-} s.t. $+x \in I_{QC}$ iff $I_{QC}(+x) = 1$ and $-x \in I_{QC}$ iff $I_{QC}(-x) = 1$. This mapping is clearly bijective (the power set of $\{+x \mid x \in PS\} \cup \{-x \mid x \in PS\}$ and the set of all total functions from PS^{+}_{-} to $\{0, 1\}$ are similar).

In order to show how $\Sigma \overset{?}{\models}_{QC} \gamma$ can be translated in linear time to an instance of the decision problem for classical entailment, we associate:

- the CNF formula Σ with a formula $\mathcal{S}(\Sigma)$ from $PROP_{PS^{+}_{-}}$ s.t. the (classical) models of $\mathcal{S}(\Sigma)$ represent exactly the strong models of Σ.
- the clause γ with a clause $\mathcal{W}(\gamma)$ from $PROP_{PS^{+}_{-}}$ s.t. the (classical) models of $\mathcal{W}(\gamma)$ represent exactly the weak models of γ.

Once this is done, $\Sigma \models_{QC} \gamma$ is equivalent to $\mathcal{S}(\Sigma) \models \mathcal{W}(\gamma)$; in other words, quasi-classical entailment is reduced to classical entailment.

Let us first present a simple clausal encoding of all weak models of a given clause.

Definition 5 *Let $\gamma = l_1 \vee \ldots \vee l_n$ ($n \geq 1$) be a clause from $PROP_{PS}$. $\mathcal{W}(l_1 \vee \ldots \vee l_n)$ is the clause from $PROP_{PS^{+}_{-}}$ defined by $\bigvee_{i=1}^{n} +l_i$.*

By construction, $\mathcal{W}(\gamma)$ is a clause of $PROP_{PS^{+}_{-}}$ (containing only positive literals) and it can be computed in linear time from γ. Interestingly, this clause captures all the weak models of γ.

Proposition 1 (encoding weak models) *Let γ be a clause from $PROP_{PS}$. A QC interpretation I_{QC} weakly satisfies γ iff I_{QC} (viewed as a classical interpretation) is a model of $\mathcal{W}(\gamma)$.*

Example 7 (continued) *We have $\mathcal{W}(\Sigma) = +a \wedge -a \wedge +b \wedge (-b \vee +c)$. Accordingly, the weak models of Σ contains $+a$, $-a$, $+b$ and ($-b$ or $+c$). Conversely, every QC interpretation which satisfies this requirement is a weak model of Σ.*

Actually, such an encoding of the weak models of the query by a classical formula can be extended to the case where the query is any formula Φ from $PROP_{PS}$. $\mathcal{W}(\Phi)$ can be computed in time linear in $|\Phi|$ by first putting Φ into NNF (we only need de Morgan's laws and double negation elimination to do it, and as evoked above, such laws preserve the weak models of the formula), then computing recursively $\mathcal{W}(\Phi)$ as follows:

- if $\Phi = l \in L_{PS}$, then $\mathcal{W}(\Phi) = +l$;

- if $\Phi = \alpha \wedge \beta$, then $\mathcal{W}(\Phi) = \mathcal{W}(\alpha) \wedge \mathcal{W}(\beta)$;
- if $\Phi = \alpha \vee \beta$, then $\mathcal{W}(\Phi) = \mathcal{W}(\alpha) \vee \mathcal{W}(\beta)$.

Again, we have $I_{QC} \models_w \Phi$ iff $I_{QC} \models \mathcal{W}(\Phi)$ (the proof by structural induction on Φ is easy using Definition 5.2 from [21]).

Let us now turn to strong satisfaction and show how to derive an encoding of all strong models of a CNF formula.

Definition 6 *Given a CNF formula Σ from $PROP_{PS}$, $\mathcal{S}(\Sigma)$ is the formula from $PROP_{PS^{\pm}}$ defined by:*

- $\mathcal{S}(\Sigma) = \bigwedge_{\gamma \ clause \ \in \ \Sigma} \mathcal{S}(\gamma)$;
- $\mathcal{S}(l_1 \vee \ldots \vee l_n) = \bigvee_{i=1}^{n}(+l_i \wedge \neg - l_i) \vee \bigwedge_{i=1}^{n}(+l_i \wedge -l_i)$.

Clearly enough, $\mathcal{S}(\Sigma)$ can be computed in time linear in $|\Sigma|$. Note that $\mathcal{S}(\Sigma)$ is typically **not** a CNF formula. Nevertheless, $\mathcal{S}(\Sigma)$ captures exactly all strong models of Σ.

Proposition 2 (encoding strong models) *Let Σ be a CNF formula from $PROP_{PS}$. A QC interpretation I_{QC} strongly satisfies Σ iff I_{QC} (viewed as a classical interpretation) is a model of $\mathcal{S}(\Sigma)$.*

Example 8 (continued) *We have* $\mathcal{S}(\Sigma) = ((+a \wedge \neg - a) \vee (+a \wedge -a)) \wedge ((-a \wedge \neg + a) \vee (-a \wedge +a)) \wedge ((+b \wedge \neg - b) \vee (+b \wedge -b)) \wedge ((-b \wedge \neg + b) \vee (+c \wedge \neg - c) \vee (-b \wedge +b \wedge -c \wedge +c))$. *$\mathcal{S}(\Sigma)$ is equivalent to the formula $+a \wedge -a \wedge +b \wedge +c \wedge (\neg - c \vee -b)$. Accordingly, every strong model of Σ contains $+a$, $-a$, $+b$, $+c$ but not $-c$ unless it contains $-b$ as well. Conversely, every QC interpretation that satisfies this requirement is a strong model of Σ.*

In contrast to the weak models situation, there is no easy way to extend the proposition above to the case where Σ is not a CNF formula. Indeed, the distribution laws for strong satisfaction are not consequences of the way $\wedge$ and $\vee$ are interpreted, but are primarily required by the definition of strong satisfaction itself (see Definition 5.1 from [21]).

Now, as a direct consequence of our translation, we obtain that deciding $\models_{QC}$ for CNF formulas is in coNP. Actually, this decision problem is complete for the class:

Proposition 3 (complexity of $\models_{QC}$) *The following decision problem:*

- **Input:** *A CNF formula Σ and a CNF formula Φ from $PROP_{PS}$.*
- **Question:** *Does $\Sigma \models_{QC} \Phi$ hold?*

is coNP*-complete.*

Since $\mathcal{W}(\Phi)$ can be computed in linear time even in the case where Φ is not a CNF formula, Proposition 3 still holds in this more general situation.

Interestingly, the complexity of quasi-classical entailment for CNF formulas is not very high, especially if we compare it with the complexity of inference relations within coherence-based approaches to inconsistency handling [30]. Moreover, as an aside of the hardness proof (see Appendix), we can observe that $\Sigma \models_{QC} \gamma$ is equivalent to $\Sigma \models \gamma$, when the CNF Σ is consistent and the clause γ is not a tautology. Accordingly, quasi-classical inference misses only trivial consequences from a consistent knowledge base (and such valid clauses can be easily recognized in polynomial time). This shows $\models_{QC}$ very close to $\models$ when the knowledge base is consistent, and this valuable feature is not shared by many paraconsistent logics.

5 Mechanizing Quasi-Classical Inference

Thanks to our translation, we can take advantage of all the existing machinery for classical entailment so as to decide quasi-classical clausal entailment. Especially, we can directly exploit the well-known Davis/Logemann/Loveland search procedure for testing consistency [11] (and all of its recent improvements, like C-SAT [17] and SATZ [8]) in order to test whether $\Sigma \models_{QC} \gamma$ holds; indeed, the latter amounts to check the consistency of a conjunctive normal form of $\mathcal{S}(\Sigma) \wedge \neg\mathcal{W}(\gamma)$[3]. While the time complexity of Davis/Logemann/Loveland search procedure and its variants is exponential in the input size in the worst case, such decision procedures exhibit a good computational behaviour in many situations. We can also take advantage of local search algorithms for consistency checking (see [35]) in order to design incomplete but fast algorithms for quasi-classical entailment.

Another advantage offered by our translation is the possibility to take advantage of *knowledge compilation techniques* (see [7] for a survey) in order to circumvent the intractability of quasi-classical clausal entailment in the general case. The key idea of compilation is pre-processing of the fixed part of the inference problem, namely the knowledge base Σ; once this is done, the compiled form of Σ is used to answer queries on-line. Provided that Σ does not often change and that answering queries from its compiled form is computationally easier than answering them from Σ, the compilation time can be balanced over a sufficient number of queries. Several knowledge compilation techniques have been proposed so far (see, among others, [12] [27] [34] [5] [10]). While none of them can ensure the objective of enhancing inference to be reached in the worst case (because the size of the compiled form can be exponentially larger than the size of the original knowledge base), experiments have shown such approaches valuable in many practical situations. All the approaches listed above can be

[3] While $\mathcal{S}(\Sigma)$ is not a CNF formula in the general case, it is well-known that every propositional formula can be turned in linear time into a CNF formula, while preserving consistency (the idea consists in incorporating new propositional variables to name subformulas; see e.g., the reduction from CIRCUIT-SAT to SAT given in [31] (pp. 163); see also [37]).

used to possibly circumvent the intractability of quasi-classical entailment; it is sufficient to compile $\mathcal{S}(\Sigma)$ during the off-line phase[4]. Interestingly, *approximate* knowledge compilation techniques [13] [36] can also be used; indeed, in our translation from $\Sigma \models_{QC} \gamma$ into $\mathcal{S}(\Sigma) \models \mathcal{W}(\gamma)$, $\mathcal{W}(\gamma)$ always is a *positive clause*. Accordingly, what is important is not to compile $\mathcal{S}(\Sigma)$ as such but to determine all the *positive clauses* that are logical consequences of $\mathcal{S}(\Sigma)$. Stated otherwise, it is sufficient to focus on a specific target language[5] (consisting of all positive clauses). Many consequence-finding algorithms can be used to this end (see [28] for a survey); especially, there are recent algorithms based on kernel resolution [14] [15] that achieve this goal in a satisfying way.

Last but not least, imposing some restrictions over Σ enables to render tractable the decision problem for quasi-classical entailment. For instance, assume that Σ is a Krom formula. While our translation function $\mathcal{S}$ does not preserve CNF, it is not hard to prove that, for every unary clause l, we have $\mathcal{S}(l) \equiv +l$, and for every binary clause $l_1 \vee l_2$, we have $\mathcal{S}(l_1 \vee l_2) \equiv (+l_1 \vee +l_2) \wedge (\neg - l_1 \vee +l_2) \wedge (+l_1 \vee \neg - l_2)$ (just distribute the disjunction connective over the conjunction one, and simplify). As a consequence, for every Krom formula Σ, a Krom formula from $PROP_{PS^{+}_{-}}$ that is equivalent to $\mathcal{S}(\Sigma)$ can be generated in time linear in $|\Sigma|$.

Example 9 (continued) *We have* $\mathcal{S}(\Sigma) \equiv +a \wedge -a \wedge +b \wedge (-b \vee +c) \wedge (\neg + b \vee +c) \wedge (-b \vee \neg - c)$. *This Krom formula can still be simplified to* $+a \wedge -a \wedge +b \wedge +c \wedge (-b \vee \neg - c)$.

Because determining whether a clause is a logical consequence of a Krom formula can be done in time linear in the input size [18], we obtain the following proposition:

Proposition 4 (a tractable class) *Let Σ be a Krom formula and Φ be a CNF formula. Determining whether $\Sigma \models_{QC} \Phi$ holds can be achieved in time linear in the input size.*

Another interesting candidate for tractability is the Horn (or the reverse Horn) CNF class. Indeed, it is tractable for clausal query answering [16]. By a straightforward induction on the clause length n, it is easy to prove that for every clause $l_1 \vee \ldots \vee l_n$, $\mathcal{S}(l_1 \vee \ldots \vee l_n)$ can be turned into an equivalent CNF formula from $PROP_{PS^{+}_{-}}$ as the conjunction of $2^n - 1$ clauses containing n literals:

$$*l_1 \vee \ldots \vee *l_n$$

where each $*l_i$ ($i \in 1 \, .. \, n$) stands for $+l_i$ or $\neg - l_i$, except that the clause $\neg - l_1 \vee \ldots \vee \neg - l_n$ is rejected.

[4] Using such knowledge compilation techniques directly on Σ would not give the expected result since both approaches trivialize when Σ is inconsistent.

[5] Also called a production field in [37].

As a consequence, a CNF formula equivalent to $\mathcal{S}(\Sigma)$ can be generated in time linear in $|\Sigma|$ when Σ is a k-CNF formula (where k is a constant).

Example 10 *Let Σ be the 3-CNF formula $(\neg a \vee \neg b \vee c) \wedge (a \vee b \vee c)$. $\mathcal{S}(\Sigma)$ is equivalent to the following CNF formula from $PROP_{PS^{+}_{-}}$:*

$$((-a \vee -b \vee +c) \wedge (\neg + a \vee -b \vee +c) \wedge (-a \vee \neg + b \vee +c) \wedge (-a \vee -b \vee \neg - c) \wedge (\neg + a \vee \neg + b \vee +c) \wedge (\neg + a \vee -b \vee \neg - c) \wedge (-a \vee \neg + b \vee \neg - c)) \wedge ((+a \vee +b \vee +c) \wedge (\neg - a \vee +b \vee +c) \wedge (+a \vee \neg - b \vee +c) \wedge (+a \vee +b \vee \neg - c) \wedge (\neg - a \vee \neg - b \vee +c) \wedge (\neg - a \vee +b \vee \neg - c) \wedge (+a \vee \neg - b \vee \neg - c)).$$

Unfortunately, as the previous example shows it, this translation does not preserve Horn CNF formulas (non Horn clauses can be generated); indeed, while the clause $\neg a \vee \neg b \vee c$ is Horn, we have:

$$\mathcal{S}(\neg a \vee \neg b \vee c) \equiv ((-a \vee -b \vee +c) \wedge (\neg + a \vee -b \vee +c) \wedge (-a \vee \neg + b \vee +c) \wedge (-a \vee -b \vee \neg - c) \wedge (\neg + a \vee \neg + b \vee +c) \wedge (\neg + a \vee -b \vee \neg - c) \wedge (-a \vee \neg + b \vee \neg - c)),$$

and this CNF formula is not Horn (since e.g., the clause $-a \vee -b \vee +c$ is not).

Nevertheless, the fact that $\mathcal{S}(\Sigma)$ is not turned into a Horn CNF formula does not prevent from making it tractable for clausal query answering. Indeed, applying the translation described above to Σ so as to reach a k-CNF formula leads to a *renamable Horn* k-CNF formula, from which clausal query answering is tractable [25]. Furthermore, it is not necessary that Σ is a Horn k-CNF formula to reach it; indeed, it is sufficient that Σ is a renamable Horn k-CNF formula.

Example 11 (continued) *Let Σ be the 3-CNF formula $(\neg a \vee \neg b \vee c) \wedge (a \vee b \vee c)$. Σ is not a Horn CNF formula, nor a reverse Horn CNF formula. However, Σ is a renamable Horn 3-CNF formula (substituting b by $\neg b$ and c by $\neg c$ in Σ results in a Horn formula). The formula equivalent to $\mathcal{S}(\Sigma)$ given above is a renamable Horn 3-CNF formula (substituting in it $-a$ by $\neg - a$, $+b$ by $\neg + b$ and $+c$ by $\neg + c$ results in a Horn formula).*

Hence, we obtain:

Proposition 5 (another tractable class) *Let Σ be a renamable Horn k-CNF formula and Φ be a CNF formula. Determining whether $\Sigma \models_{QC} \Phi$ holds can be achieved in time linear in the input size.*

Interestingly, the class of renamable Horn k-CNF formulas is quite large; especially, it contains all the Horn k-CNF formulas and all the reverse Horn k-CNF formulas[6]. Accordingly, we have at our disposal a quite large class of propositional CNF formulas from which quasi-classical clausal entailment can be decided very efficiently, and this is a good point w.r.t. applications.

[6] It also contains the consistent Krom formulas, but this is less interesting for us since quasi-classical logic is typically of interest when the knowledge base Σ under consideration is inconsistent.

6 Conclusion

In this paper, we investigated some computational aspects of quasi-classical logic. Encoding the semantics of quasi-classical logic into a propositional language enabled us to exhibit a (simple) linear time translation from the quasic-classical entailment decision problem (for CNF formulas) to the classical entailment decision problem. We argue that this is an important step towards the practical application of this non-classical logic. Indeed, based on the proposed translation, we showed that deciding quasi-classical entailment for CNF formulas is "only" coNP-complete. This complexity is typically lower than the complexity of inference within challenging approaches to inconsistency handling. From the practical side, the translation enables to take advantage of all the existing machinery for classical entailment. We also pointed out two classes of propositional formulas from which clausal quasi-classical entailment can be decided in linear time.

Acknowledgements

We are very grateful to an anonymous reviewer for her helpful comments. Many thanks as well to the Région Nord / Pas-de-Calais, the IUT de Lens and the European Communities for their support. Nadège Porquet is granted by the Région Nord / Pas-de-Calais.

Appendix

Proof of Proposition 1:
By definition, I_{QC} is a model of $\mathcal{W}(\gamma)$ iff there exists a literal l of $\mathcal{W}(\gamma)$ s.t. $I_{QC}(l) = 1$. Since every literal l of $\mathcal{W}(\gamma)$ is of the form $+x$ or $-x$, this is equivalent to state that there exists a literal x of γ s.t. $+x \in I_{QC}$ or a literal $\neg x$ of γ s.t. $-x \in I_{QC}$. By definition of weak satisfaction, this is equivalent to state that I_{QC} weakly satisfies γ. ∎

Proof of Proposition 2:
Like classical interpretations, a QC interpretation strongly satisfies a conjunctive formula iff it satisfies every conjunct of it. Hence, it is sufficient to consider the case where Σ is a clause γ. Now, Proposition 3.1 from [21] shows that for every clause $l_1 \vee \ldots \vee l_n$ and every QC interpretation I_{QC}, we have $I_{QC} \models_s l_1 \vee \ldots \vee l_n$ iff for some $i \in 1 \mathrel{..} n$, we have $(+l_i \in I_{QC}$ and $-l_i \notin I_{QC})$ or for every $i \in 1 \mathrel{..} n$, we have $(+l_i \in I_{QC}$ and $-l_i \in I_{QC})$. What we do is just encoding the semantics within a propositional language: I_{QC} can be viewed as a standard interpretation I over PS^+_- s.t. for every variable x of PS, $+x \in I_{QC}$ iff $I_{QC}(+x) = 1$ and $-x \in I_{QC}$ iff $I_{QC}(-x) = 1$. As a consequence, Proposition 3.1 from [21] can be stated as $I_{QC} \models_s l_1 \vee \ldots \vee l_n$ iff $I_{QC} \models \bigvee_{i=1}^{n}(+l_i \wedge \neg - l_i) \vee \bigwedge_{i=1}^{n}(+l_i \wedge -l_i)$.

■

Proof of Proposition 3:

- Membership to coNP. This is a straightforward consequence of our linear-time translation of the quasi-entailment decision problem. Indeed, we have $\Sigma \models_{QC} \gamma$ iff $\mathcal{S}(\Sigma) \models \mathcal{W}(\gamma)$, the latter decision problem is in coNP, and coNP is closed under polynomial time reduction.

- coNP-hardness. Let $\Sigma = \gamma_1 \wedge \ldots \wedge \gamma_n$ be a CNF formula and $new \in PS \setminus Var(\Sigma)$ be a new variable. Σ is inconsistent iff $\Sigma \vee new \equiv new$ holds iff $(\gamma_1 \vee new) \wedge \ldots \wedge (\gamma_n \vee new) \models new$ holds. Now, the CNF formula $\Sigma_{new} = (\gamma_1 \vee new) \wedge \ldots \wedge (\gamma_n \vee new)$ is consistent (whatever Σ) since interpreting new as 1 is sufficient to satisfy it. Subsequently, we have $\Sigma_{new} \models new$ iff new is a prime implicate of Σ_{new}. As a consequence of Lee's subsumption theorem [23], new is a prime implicate of the consistent CNF formula Σ_{new} iff there is a resolution proof of it from Σ_{new} (see Corollary to Proposition 7 in [26]) iff new is derivable from Σ_{new} in the proof theory of quasi-classical logic. To sum up, we have Σ inconsistent iff $\Sigma_{new} \models_{QC} new$. Since Σ_{new} can be computed in time linear in $|\Sigma|$, this is a polynomial time reduction from UNSAT to the quasi-classical entailment problem. This shows it coNP-complete (even in the restricted case where the query is a positive literal and the knowledge base is consistent).

■

Proof of Proposition 4:
This is a direct consequence of the fact that for every Krom formula Σ, we can compute in linear time a Krom formula equivalent to $\mathcal{S}(\Sigma)$. Indeed, for each clause $l_1 \vee l_2$, we have $\mathcal{S}(l_1 \vee l_2) \equiv (+l_1 \vee +l_2) \wedge (\neg - l_1 \vee +l_2) \wedge (+l_1 \vee \neg - l_2)$. Now, for every CNF formula Φ, we have $\Sigma \models_{QC} \Phi$ iff every clause γ of Φ is a quasi-classical consequence of Σ, and for every clause γ, we have $\Sigma \models_{QC} \gamma$ iff $\mathcal{S}(\Sigma) \models \mathcal{W}(\gamma)$ holds iff $\mathcal{S}(\Sigma) \wedge \neg\mathcal{W}(\gamma)$ is inconsistent. Since $\neg\mathcal{W}(\gamma)$ is a conjunction of literals (that can be computed in linear time from γ), the formula $\mathcal{S}(\Sigma) \wedge \neg\mathcal{W}(\gamma)$ is Krom. Hence, its consistency can be tested in time linear in $|\Sigma| + |\gamma|$. ■

Proof of Proposition 5:
It is well-known that a CNF formula Σ is renamable Horn iff there exists an interpretation I s.t. for every clause $l_1 \vee \ldots \vee l_n$ of Σ, I satisfies every literal l_i ($i \in 1 \,..\, n$) except at most one [25]. In the CNF encoding of $\mathcal{S}(\Sigma)$ that we point out, every clause $l_1 \vee \ldots \vee l_n$ is associated to $2^n - 1$ clauses $*l_1 \vee \ldots \vee *l_n$, which is a constant number of clauses when n is bounded. In these clauses, $*l_i$ ($i \in 1 \,..\, n$) stands for $+l_i$ or $\neg - l_i$, except that the clause $\neg - l_1 \vee \ldots \vee \neg - l_n$ is not authorized. Let us associate to such an interpretation I over PS (a witness

of the Horn renamability of Σ) an interpretation I_{-}^{+} over PS_{-}^{+} s.t. for every $i \in 1 \,..\, n$, $I(l_i) = 1$ iff $I_{-}^{+}(+l_i) = 1$ and $I_{-}^{+}(-l_i) = 0$. Clearly enough, I_{-}^{+} satisfies all the literals (except at most one) from every of the $2^n - 1$ clauses $*l_1 \vee \ldots \vee *l_n$. Accordingly, our CNF encoding of $\mathcal{S}(\Sigma)$ is a renamable Horn formula whenever Σ is such. Moreover, it can be generated in time linear in $|\Sigma|$ when Σ is a k-CNF formula. Now, for every CNF formula Φ, we have $\Sigma \models_{QC} \Phi$ iff every clause γ of Φ is a quasi-classical consequence of Σ, and for every clause γ, we have $\Sigma \models_{QC} \gamma$ iff $\mathcal{S}(\Sigma) \models \mathcal{W}(\gamma)$ holds iff $\mathcal{S}(\Sigma) \wedge \neg\mathcal{W}(\gamma)$ is inconsistent. Since $\neg\mathcal{W}(\gamma)$ is a conjunction of literals (that can be computed in linear time from γ), the CNF formula $\mathcal{S}(\Sigma) \wedge \neg\mathcal{W}(\gamma)$ is renamable Horn. Hence, its consistency can be tested in time linear in $|\Sigma| + |\gamma|$. ■

References

[1] N. Belnap. A useful four-valued logic. In *Modern Uses of Multiple-Valued Logic*, Reidel, 8-37, 1977.

[2] S. Benferhat, C. Cayrol, D. Dubois, J. Lang and H. Prade. Inconsistency management and prioritized syntax-based entailment. *Proc. of the* 13^{th} *International Joint Conference on Artificial Intelligence (IJCAI'93)*, Chambéry, 640-645, 1993.

[3] Ph. Besnard and A. Hunter. Quasi-classical logic: non-trivializable classical reasoning from inconsistent information. *Proc. of the European Conference on Symbolic and Qualitative Approaches to Reasoning and Uncertainty (ECSQARU'95)*, LNCS 946, Springer-Verlag, 44-51, 1995.

[4] Ph. Besnard and A. Hunter. Introduction to actual and potential contradictions. In *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, vol. 2, Kluwer Academic Publishers, 1-11, 1998.

[5] Y. Boufkhad, E. Grégoire, P. Marquis, B. Mazure and L. Saïs. Tractable cover compilations. *Proc. of the* 15^{th} *International Joint Conference on Artificial Intelligence (IJCAI'97)*, Nagoya, 122-127, 1997.

[6] R.J. Brachman and H.J. Levesque. A fundamental tradeoff in knowledge representation and reasoning. *Readings in Knowledge Representation*, Morgan Kaufmann, 41-70, 1985.

[7] M. Cadoli and F.M. Donini. A survey on knowledge compilation. *AI Communications* 10, 137-150, 1998.

[8] Chu Min Li and Anbulagan. Heuristics based on unit propagation for satisfiability problems. *Proc. of the* 15^{th} *International Joint Conference on Artificial Intelligence (IJCAI'97)*, Nagoya, 366-371, 1997.

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

[10] A. Darwiche. Compiling knowledge into decomposable negation normal form. *Proc. of the* 16^{th} *International Joint Conference on Artificial Intelligence (IJCAI'99)*, Stockholm, 284-289, 1999.

[11] M. Davis, G. Logemann and D. Loveland. A machine program for theorem proving. *Journal of the Association for Computing Machinery* 5:394-397, 1962.

[12] A. del Val. Tractable databases: how to make propositional unit resolution complete through compilation. *Proc. of the* 4^{th} *International Conference on Principles of Knowledge Representation and Reasoning (KR'94)*, Bonn, 551-561, 1994.

[13] A. del Val. An analysis of approximate knowledge compilation. *Proc. of the* 14^{th} *International Joint Conference on Artificial Intelligence (IJCAI'95)*, Montreal, 830-836, 1995.

[14] A. del Val. A new method for consequence finding and compilation in restricted languages. *Proc. of the* 16^{th} *National Conference on Artificial Intelligence (AAAI'99)*, Orlando (FL), 259-264, 1999.

[15] A. del Val. The complexity of restricted consequence finding and abduction. *Proc. of the* 17^{th} *National Conference on Artificial Intelligence (AAAI'00)*, Austin (TX), 337-342, 2000.

[16] W.H. Dowling and J.H. Gallier. Linear-time algorithms for testing satisfiability of propositional Horn formulae. *Journal of Logic Programming* 3:267-284, 1984.

[17] O. Dubois, P. André, Y. Boufkhad and J. Carlier. SAT versus UNSAT. *Second DIMACS Challenge*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, American Mathematical Society, 415-436, 1996.

[18] S. Even, A. Itai and A. Shamir. On the complexity of timetable and multicommodity flow problems. *SIAM Journal of Computing* 20(6):1119-1127, 1976.

[19] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*, Freeman, 1979.

[20] A. Hunter. Paraconsistent logics. In *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, vol. 2, Kluwer Academic Publishers, 11-36, 1998.

[21] A. Hunter. Reasoning with contradictory information using quasi-classical logic. *Journal of Logic and Computation* 10(5):677-703, 2000.

[22] J. Lang and P. Marquis. In search of the right extension. *Proc. of the* 7^{th} *International Conference on Principles of Knowledge Representation and Reasoning (KR'00)*, Breckenridge (CO), 625-636, 2000.

[23] R.C.T. Lee. A completeness theorem and a computer program for finding theorems derivable from given axioms. Ph.D. thesis, University of California, 1967.

[24] H.J. Levesque. A knowledge-level account of abduction. *Proc. of the* 11^{th} *International Joint Conference on Artificial Intelligence (IJCAI'89)*, Detroit (MI), 1061-1067, 1989.

[25] H.R. Lewis. Renaming a set of clauses as Horn set. *Journal of the Association for Computing Machinery* 25:134-135, 1978.

[26] P. Marquis. Skeptical abduction. *International Journal on Artificial Intelligence Tools* 2(4):511-540, 1993.

[27] P. Marquis. Knowledge compilation using theory prime implicates. *Proc. of the* 14^{th} *International Joint Conference on Artificial Intelligence (IJCAI'95)*, Montreal, 837-843, 1995.

[28] P. Marquis. Consequence finding algorithms. In *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, vol. 5, Kluwer Academic Publishers, 41-145, 2000.

[29] P. Marquis and N. Porquet. Resource-bounded inference from inconsistent belief bases. *Proc. of the* 17^{th} *International Joint Conference on Artificial Intelligence (IJCAI'01)*, Seattle (WA), to appear, 2001.

[30] B. Nebel. How hard is it to revise a belief base? In *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, vol. 3, Kluwer Academic Publishers, 77-145, 1998.

[31] Ch. H. Papadimitriou. *Computational Complexity*, Addison-Wesley, 1994.

[32] G. Pinkas and R.P. Loui. Reasoning from inconsistency: a taxonomy of principles for resolving conflict. *Proc. of the* 2^{nd} *International Conference on Principles of Knowledge Representation and Reasoning (KR'92)*, Cambridge (MA), 709-719, 1992.

[33] M. Schaerf and M. Cadoli. Tractable reasoning via approximation. *Artificial Intelligence* 74(2):249-310, 1995.

[34] R. Schrag. Compilation for critically constrained knowledge bases. *Proc. of the* 13^{th} *National Conference on Artificial Intelligence (AAAI'96)*, Portland (OR), 510-515, 1996.

[35] B. Selman, H.J. Levesque and D. Mitchell. GSAT: a new method for solving hard satisfiability problems. *Proc. of the* 10^{th} *National Conference on Artificial Intelligence (AAAI'92)*, San Jose (CA), 440-446, 1992.

[36] B. Selman and H.A. Kautz. Knowledge compilation and theory approximation. *Journal of the Association for Computing Machinery* 43(2):193-224, 1996.

[37] P. Siegel. Représentation et utilisation des connaissances en calcul propositionnel. Thèse d'Etat, Université d'Aix-Marseille II, 1987.