Representing Paraconsistent Reasoning via Quantified Propositional Logic*

Philippe Besnard[1], Torsten Schaub[2**], Hans Tompits[3], and Stefan Woltran[3]

[1] IRIT-CNRS
118, route de Narbonne, F–31062 Toulouse Cedex
besnard@irit.fr
[2] Institut für Informatik, Universität Potsdam,
Postfach 90 03 27, D–14439 Potsdam, Germany
torsten@cs.uni-potsdam.de
[3] Institut für Informationssysteme 184/3, Technische Universität Wien,
Favoritenstraße 9–11, A–1040 Vienna, Austria
[tompits,stefan]@kr.tuwien.ac.at

Abstract. Quantified propositional logic is an extension of classical propositional logic where quantifications over atomic formulas are permitted. As such, quantified propositional logic is a fragment of second-order logic, and its sentences are usually referred to as *quantified Boolean formulas* (QBFs). The motivation to study quantified propositional logic for paraconsistent reasoning is based on two fundamental observations. Firstly, in recent years, practicably efficient solvers for quantified propositional logic have been presented. Secondly, complexity results imply that there is a wide range of paraconsistent reasoning problems which can be efficiently represented in terms of QBFs. Hence, solvers for QBFs can be used as a core engine in systems prototypically implementing several of such reasoning tasks, most of them lacking concrete realisations. To this end, we show how certain paraconsistent reasoning principles can be naturally formulated or reformulated by means of quantified Boolean formulas. More precisely, we describe polynomial-time constructible encodings providing axiomatisations of the given reasoning tasks. In this way, a whole variety of a priori distinct approaches to paraconsistent reasoning become comparable in a uniform setting.

1 Introduction

Paraconsistent reasoning, that is, reasoning from inconsistent information, is a central yet rather complex task underlying the vital reasoning capacities of in-

* The third and fourth author were partially supported by the Austrian Science Foundation (FWF) under grant P15068, as well as by the European Commission under project IST-2001-33570 INFOMIX and the IST-2001-33123 CologNeT Network of Excellence.

** Affiliated with the School of Computing Science at Simon Fraser University, Burnaby, Canada.

L. Bertossi et al. (Eds.): Inconsistency Tolerance, LNCS 3300, pp. 84–118, 2004.

telligent agents. In view of our daily information feed, it even becomes more and more important every day. As opposed to neighbouring fields like database systems or nonmonotonic reasoning, whose mainstream has or is about to converge to a canonical approach, viz. relational algebra or answer-set programming, respectively, the inherent manifoldness of reasoning from inconsistent information (still) offers a whole variety of different approaches. As a consequence, there is a lack of implemented systems for paraconsistent reasoning.

In this chapter, we address paraconsistent reasoning from the perspective of *quantified propositional logic*, which is an extension of classical propositional logic where quantifications over atomic formulas are permitted. As such, quantified propositional logic is a fragment of second-order logic, and its sentences are usually referred to as *quantified Boolean formulas* (QBFs).

The motivation to study quantified propositional logic for paraconsistent reasoning is based on two fundamental observations. Firstly, in recent years, practicably efficient solvers for quantified propositional logic have been presented. Secondly, in view of results from complexity theory, a wide range of paraconsistent reasoning problems can be efficiently represented in terms of QBFs. Hence, solvers for QBFs can be used as a core engine in systems prototypically implementing several of such reasoning tasks, most of them lacking concrete realisations.

The basic contribution of this chapter is to illustrate how paraconsistent reasoning principles can be naturally formulated or reformulated by means of quantified Boolean formulas. That is to say, we are interested in *encodings* of paraconsistency in terms of QBFs. More specifically, given a paraconsistent inference relation $\vdash_{\mathrm{P}}$, we provide a mapping $\mathcal{T}_{\mathrm{P}}[\cdot;\cdot]$, assigning, to each theory T and each formula φ, a QBF $\mathcal{T}_{\mathrm{P}}[T;\varphi]$ such that

1. $T \vdash_{\mathrm{P}} \varphi$ iff $\mathcal{T}_{\mathrm{P}}[T;\varphi]$ is valid in quantified propositional logic,
2. the size of $\mathcal{T}_{\mathrm{P}}[T;\varphi]$ is polynomial in the size of T and φ, and
3. determining the validity of QBFs resulting from translation $\mathcal{T}_{\mathrm{P}}[\cdot;\cdot]$ is not computationally harder than checking inference under $\vdash_{\mathrm{P}}$.

Hence, encodings of this kind provide *axiomatisations* of the respective inference relation which are efficiently computable. In this way, a whole variety of a priori distinct approaches to paraconsistent reasoning can be compared in a uniform setting.

Our chapter is organised as follows. We start with an introduction to quantified propositional logic in Section 2, including basic intuitions, historical remarks, formal preliminaries, and complexity issues. Notably, this section introduces half a dozen basic QBF modules that can be used as building blocks for assembling axiomatisations of numerous reasoning tasks. These modules are then used in Section 3 to conduct three case-studies, demonstrating how existing approaches to paraconsistent reasoning can be axiomatised and thus implemented by means of QBFs.

2 Quantified Propositional Logic

2.1 Overview and Motivation

As mentioned previously, the language of quantified propositional logic is an extension of classical propositional logic in which formulas may contain quantifications over propositional atoms. Sentences of this language are called *quantified Boolean formulas* (QBFs), and often in the literature one identifies this term with the language of quantified propositional formulas *simpliciter*.

As in first-order logic, the quantifiers permitted in quantified propositional logic are either *existential* or *universal.* We illustrate the underlying ideas by some simple examples.

Consider the propositional formula

$$(p \to q) \land (q \to p). \tag{1}$$

Clearly, setting both p and q jointly to either true or false makes (1) true, otherwise the formula evaluates to false. Hence, (1) is *satisfiable* but not *valid.*

Imagine we want to talk about satisfiability or validity *within the logical language itself.* In other words, we want to capture the *meta-linguistic* concept of truth assignments within a suitable extension of the object language. To this end, we express a proposition of form

> "there exist truth assignments to p and q such that $(p \to q) \land (q \to p)$ evaluates to true"

in the language of QBFs, using the formula

$$\exists p \, \exists q \left((p \to q) \land (q \to p)\right). \tag{2}$$

Analogously, in order to talk about validity of a formula, say of (1), we may write

$$\forall p \, \forall q \left((p \to q) \land (q \to p)\right). \tag{3}$$

Hence, we extended the alphabet of classical propositional logic by two *quantifier symbols*, $\exists$ and $\forall$. We call $\exists$ the *existential* (*Boolean*) *quantifier symbol* and $\forall$ the *universal* (*Boolean*) *quantifier symbol.* By the intuitive meaning of quantifiers, we immediately get that QBF (2) evaluates to true, whereas QBF (3) evaluates to false.

However, using the extended language, we can construct further formulas, for instance,

$$\exists p \, \forall q \left((p \to q) \land (q \to p)\right); \text{ or} \tag{4}$$

$$\forall p \, \exists q \left((p \to q) \land (q \to p)\right). \tag{5}$$

Formula (4) can be interpreted like this:

> "Does there exist a truth assignment to p such that, for all truth assignments to q, formula (1) evaluates to true?"

By inspecting the usual truth conditions for $(p \rightarrow q) \wedge (q \rightarrow p)$, it is clear that this is not the case. On the other hand, QBF (5) evaluates to true.

QBFs of form (2)–(5) are all *closed* QBFs since each variable v occurs in the scope of a quantifier $\exists v$ or $\forall v$. *Open* formulas like

$$\exists q \Big((p \rightarrow q) \wedge (q \rightarrow p)\Big) \tag{6}$$

can be evaluated, analogously to open formulas in predicate logic, with respect to interpretations, i.e., given truth assignments for the *free* variables (in our case, p).

All formulas (1)–(6) are *well-formed QBFs*. So, each classical propositional formula is *a fortiori* a QBF. Moreover, for every atom p, we allow the unary operators $\exists p$ and $\forall p$ to appear "anywhere" in a QBF, not just at the beginning of a formula. For instance,

$$\exists p \Big(\exists q\,(p \rightarrow q) \wedge \forall q\,(q \rightarrow p)\Big)$$

is also a well-formed QBF. It is left to the reader to show that this formula evaluates to true.

In general, QBFs can be seen as a *conservative extension* of classical propositional logic, i.e., to each QBF we can assign a logically equivalent propositional formula. However, the advantage of QBFs is their compactness: to express a QBF as a logically equivalent propositional formula, one has to face an exponential increase of the formula size, in general.

In summarising, one may consider QBFs as an extension of classical propositional logic in which reasoning over truth assignments within the object language can be expressed. A different way to think of QBFs is to regard them as a subclass of second-order logic, restricting predicates to be of arity 0, and therefore to consider formulas without function symbols and object variables.

2.2 Usability of QBFs

Historically, among the first logical analyses of systems dealing with quantifiers over propositional variables are the investigations due to Russell ("theory of implication" [63]) and Łukasiewicz and Tarski ("erweiterter Aussagenkalkül" [45]), not to mention the monumental *Principia Mathematica* [70]. The particular idea of quantifying propositional variables was extended in Leśniewski's system of *protothetic logic* [42, 65] where variables whose values are *truth functions* are allowed and quantification is defined over these variables.[1]

However, it took several decades until, in the beginning of the seventies of the last century, propositional quantification got into the spotlight of computer science, in particular of the new and developing field of complexity theory [34].

[1] A more elaborate overview on these early historical aspects of propositional quantification can be found in §28 of Church's *Introduction to Mathematical Logic* [21].

Meyer and Stockmeyer [48] were the first who showed that the evaluation problem for QBFs is complete for the complexity class PSPACE—this class comprises all problems which can be decided by deterministic Turing machines with a space requirement polynomially related to the representation size of the problem. In fact, what was considered there were Boolean expressions, and the quantifiers were part of the problem description and not of the language. Already in [47], the same authors introduced the *polynomial hierarchy* [67] as an analogue to the arithmetic hierarchy of recursion theory. Starting from $\Sigma_1^P = \mathrm{NP}$ (NP comprises all problems which can be decided by nondeterministic Turing machines in polynomial time), they defined classes Σ_{k+1}^P, for $k \geq 1$,

> "as the family of sets of words accepted in nondeterministic polynomial time by Turing machines with oracles for sets Σ_k^P" [48].

In that paper, it was already shown that each member of the hierarchy possesses a complete decision problem, given by the evaluation problem of QBFs having a specific quantifier structure (viz., of QBFs being in prenex normal form[2] and such that both the leading quantifier and the number of quantifier alternations is fixed). Other classes, like Π_k^P and Δ_k^P, which are today identified as basic components of the polynomial hierarchy, first appeared in [67, 72].

In view of the above completeness results, the evaluation problem for QBFs plays the same role for the respective classes of the polynomial hierarchy as the satisfiability problem for classical propositional logic, SAT, does for the central complexity class NP. More precisely, hardness for a particular class in the polynomial hierarchy can be shown by reducing the evaluation problem for the respective class of QBFs into the problem under consideration (see [41, 66, 16] for prominent PSPACE-completeness results, or [36, 30] for complexity results for nonmonotonic logics which reside on the second level of the polynomial hierarchy). On the other hand, if we know membership for a problem in some class of the polynomial hierarchy, we are guaranteed that there must exist an efficient encoding in terms of QBFs having a restricted number of quantifier alternations.

Note that the latter observation allows us to find appropriate translation schemas into QBFs such that the resultant formulas can be employed to decide the original problem. Moreover, in many cases, satisfying truth assignments to the free variables in such QBFs correspond to solutions of the original reasoning task. Such encodings provide us thus with a *uniform axiomatisation* for all the considered problems, which leads to further insights as well as allowing the comparison of differing problems in a well-studied and common setting. In fact, this is one of the aims of this article, where different paraconsistent reasoning principles are represented as QBFs, summarising and extending previous work [12, 13]. Other application areas of this general methodology, such as expressing planning problems or different forms of nonmonotonic reasoning in terms of QBFs, is reported, e.g., in [60, 6, 69, 27, 31, 68, 25, 28, 53].

[2] The notion of a prenex normal form for QBFs is defined analogously as for formulas in first-order logic; cf. also Section 2.4 for more details.

However, the practical impact of this line of research clearly depends on the capabilites of suitable QBF-solvers which can be applied as underlying inference engines in order to solve the reduced problems. In contrast to similar methods using reductions to SAT, where impressive results have been achieved by employing sophisticated SAT-solvers (for instance in the area of planning [37, 38]), practical implementations for evaluating QBFs lagged behind for quite a long time. This changed when Kleine-Büning *et al.* [39] presented the first implemented QBF-solver, which was based on a generalisation of the resolution principle [62]. Later, an alternative—and more promising—approach was presented by Cadoli *et al.* [17] relying on an adaption of the Davis-Putnam-Logemann-Loveland (DPLL) procedure [24, 23] for propositional logic to quantified propositional logic. Starting from this seminal paper, a number of other solvers for QBFs have been developed, like, e.g., the systems described in [32, 35, 43, 61, 73], which are based on improvements of the DPLL procedure for QBFs and by adapting several methods known from propositional logic, or by introducing new methods. It is worth mentioning that one of these solvers was also designed to run on a distributed system [32, 64]. Hence, the availability of such a parallel algorithm, and the fact that we can represent a complex problem by means of QBFs faithfully, we directly obtain a distributed decision procedure for this particular problem. This convenient situation obviously avoids designing special-purposed distributed algorithms for the problem under consideration.

Recently, Ayari and Basin [6] argued that DPLL procedures need not to be the best choice in general, and an alternative approach for solving QBFs was thus put forth (a similar idea is also outlined in [54]). These new ideas promise to be very efficient, at least on some particular classes of QBFs, a situation similar to the case when *binary decision diagrams* (BDDs) [15, 49] were proposed to evaluate QBFs.

2.3 Formal Postulates of Quantified Propositional Logic

Definition 1. *The* alphabet (*or* signature) *of the language of quantified propositional logic consists of the following items:*

1. *a countable set of* propositional variables (*or* atoms)*;*
2. *the* logical constants *"$\top$" and "$\bot$";*
3. *the* logical connectives *"$\neg$", "$\vee$", "$\wedge$", "$\rightarrow$", and "$\equiv$";*
4. *the* quantification symbols *"$\exists$" and "$\forall$"; and*
5. *the auxiliary symbols "(" and ")".* □

Definition 2. *The set of* quantified Boolean formulas (*QBFs*)*, or* (well-formed) formulas of quantified propositional logic*, is inductively defined as follows:*

1. *any propositional variable and any logical constant is a QBF;*
2. *if Φ is a QBF, then $(\neg\Phi)$ is a QBF;*
3. *if Φ and Ψ are QBFs, then $(\Phi \wedge \Psi)$, $(\Phi \vee \Psi)$, $(\Phi \rightarrow \Psi)$, and $(\Phi \equiv \Psi)$ are QBFs;*

4. *if p is a propositional variable and Φ is a QBF, then $(\exists p\, \Phi)$ and $(\forall p\, \Phi)$ are QBFs;*
5. *the only QBFs are those given by 1–4.* □

We tacitly assume the usual conventions concerning the ommission of parentheses in formulas where no ambiguities can arise. Furthermore, we use upper-case Greek letters as meta-variables for QBFs, whilst lower-case Greek letters stand for propositional formulas (i.e., quantifier-free QBFs).

By a *theory* we understand a finite set of quantifier-free formulas. Often, we identify a theory, T, with the (finite) conjunction of its elements $\bigwedge_{\phi \in T} \phi$. Furthermore, for $T = \emptyset$, we define $\bigwedge_{\phi \in T} = \top$.

Let $\mathsf{Q} \in \{\exists, \forall\}$ be a quantifier symbol. For a formula $\mathsf{Q}p\,\Psi$, we call Ψ the *scope* of $\mathsf{Q}p$. Moreover, given a finite set P of atoms, $\mathsf{Q}P\,\Psi$ stands for any QBF $\mathsf{Q}p_1\mathsf{Q}p_2 \ldots \mathsf{Q}p_n\Psi$ such that the variables $p_1, \ldots, p_n$ are pairwise distinct and $P = \{p_1, \ldots, p_n\}$.

Our definition of quantified Boolean formulas is rather unrestricted in two ways: Firstly, in contrast to some formalisations of QBFs in the literature, we allow quantifiers to appear *anywhere* in a formula. Secondly, we do not stipulate any restriction on the quantification, i.e., we do not require that a quantified variable p in $\mathsf{Q}p\,\Phi$ $(\mathsf{Q} \in \{\exists, \forall\})$ occurs in the scope Φ of $\mathsf{Q}p$. For example, $(\exists p\,(q \wedge r))$ is a QBF, and so is $(\exists p\,(\forall p\,(p \to q)))$.

As usual, an occurrence of a variable p in a QBF Φ is *free* iff it does not appear in the scope of a quantifier $\mathsf{Q}p$, otherwise the occurrence of p is *bound*. If Φ contains no free variable occurrences, then Φ is *closed*, otherwise Φ is *open*. Furthermore, $\Phi[p_1/\Psi_1, \ldots, p_n/\Psi_n]$ denotes the result of uniformly substituting in Φ each free occurrence of a variable p_i by a formula Ψ_i, for $1 \leq i \leq n$.

The semantics of quantified propositional logic is based on the following notion. Let P be a non-empty set of atoms. A (*two-valued*) *interpretation*, I, (*over* P) is a function assigning to each atom from P an element from $\{t, f\}$. If $I(p) = t$, then p is *true under* I, otherwise p is *false under* I. We usually view interpretations as subsets of P such that p is true under I just in case $p \in I$. Interpretations induce truth values of general formulas recursively in the following way.

Definition 3. *Let P be a non-empty set of atoms and Φ a QBF such that all atoms occurring in Φ belong to P. The* truth value, *$v_I(\Phi)$, of Φ under an interpretation $I : P \to \{t, f\}$ is defined by the following conditions:*

1. *if $\Phi = \top$, then $v_I(\Phi) = t$, and if $\Phi = \bot$, then $v_I(\Phi) = f$;*
2. *if $\Phi = p$, for an atom p, then $v_I(\Phi) = I(p)$;*
3. *if $\Phi = \neg\Psi$, then $v_I(\Phi) = t$ if $v_I(\Psi) = f$, otherwise $v_I(\Phi) = f$;*
4. *if $\Phi = (\Phi_1 \wedge \Phi_2)$, then $v_I(\Phi) = t$ if $v_I(\Phi_1) = v_I(\Phi_2) = t$, otherwise $v_I(\Phi) = f$;*
5. *if $\Phi = (\Phi_1 \vee \Phi_2)$, then $v_I(\Phi) = t$ if $v_I(\Phi_1) = 1$ or $v_I(\Phi_2) = 1$, otherwise $v_I(\Phi) = f$;*
6. *if $\Phi = (\Phi_1 \to \Phi_2)$, then $v_I(\Phi) = t$ if $v_I(\Phi_1) = f$ or $v_I(\Phi_2) = t$, otherwise $v_I(\Phi) = f$;*
7. *if $\Phi = (\Phi_1 \equiv \Phi_2)$, then $v_I(\Phi) = t$ if $v_I(\Phi_1) = v_I(\Phi_2)$, otherwise $v_I(\Phi) = f$;*

8. *if* $\Phi = \forall p\Psi$*, then* $v_I(\Phi) = t$ *if* $v_I(\Psi[p/\top]) = v_I(\Psi[p/\bot]) = t$*, otherwise* $v_I(\Phi) = f$*;*
9. *if* $\Phi = \exists p\Psi$*, then* $v_I(\Phi) = t$ *if* $v_I(\Psi[p/\top]) = t$ *or* $v_I(\Psi[p/\bot]) = t$*, otherwise* $v_I(\Phi) = f$*.*

Observe that it obviously holds that

$$v_I(\forall p\Psi) = v_I(\Psi[p/\top] \wedge \Psi[p/\bot]) \quad \text{and}$$
$$v_I(\exists p\Psi) = v_I(\Psi[p/\top] \vee \Psi[p/\bot]).$$

We say that Φ is *true under* I if $v_I(\Phi) = t$, otherwise Φ is *false under* I. If $v_I(\Phi) = t$, then I is a *model* of Φ. If Φ possesses some model, then Φ is *satisfiable*, otherwise Φ is *unsatisfiable*. If Φ is true under every interpretation, then Φ is *valid*. As usual, we also write $\models \Phi$ to express that Φ is valid.

It is easily seen that the truth value of a QBF Φ under interpretation I depends only on the free variables in Φ. Hence, without loss of generality, for determining the truth value of QBFs, we may restrict our attention to interpretations which contain only atoms occurring free in the given QBF. In particular, closed QBFs are either true under every interpretation or false under every interpretation, i.e., they are either valid or unsatisfiable. So, for closed QBFs, there is no need to refer to particular interpretations. As well, if a closed QBF Φ is valid, we say that Φ *evaluates to true*, and, correspondingly, if Φ is unsatisfiable, we say that Φ *evaluates to false*. Two formulas (i.e., ordinary propositional formulas or QBFs) are (*logically*) *equivalent* iff they possess the same models. Thus, formulas Φ and Ψ are logically equivalent iff $\Phi \equiv \Psi$ is valid.

We also use $\models$ to refer to the semantic consequence relation between a theory (i.e., a finite set of propositional formulas) and a propositional formula, defined in the usual way. Accordingly, for a theory T, the deductive closure of T, i.e., the set of all semantic consequences of T, is given by $Cn(T) = \{\varphi \mid T \models \varphi\}$. Furthermore, $var(T)$ denotes the set of all atoms occurring in T.

Similar to classical first-order logic, there are several results concerning the shifting and renaming of quantifiers. We list some fundamental relations below and refer the interested reader to [29, 71] for a fuller discussion.

Proposition 1. *Let* p, q *be atoms and* $\mathsf{Q} \in \{\forall, \exists\}$*. Furthermore, let* Φ, Ψ *be QBFs such that* Ψ *does not contain free occurrences of* p*. Then,*

1. $\models (\neg\exists p\Phi) \equiv \forall p(\neg\Phi)$,
2. $\models (\neg\forall p\Phi) \equiv \exists p(\neg\Phi)$,
3. $\models (\Psi \circ \mathsf{Q}p\Phi) \equiv \mathsf{Q}p(\Psi \circ \Phi)$*, for* $\circ \in \{\wedge, \vee, \rightarrow\}$*, and*
4. $\models (\mathsf{Q}q\Psi) \equiv (\mathsf{Q}p\Psi[q/p])$.

2.4 Computational Complexity

We assume the reader familiar with the basic concepts of complexity theory (see, e.g., [52] for a comprehensive introduction). Relevant for our purposes are the elements of the *polynomial hierarchy* [67], introduced in [48] as a computational analogue to the arithmetic hierarchy of recursion theory, consisting of the following sequence of classes:

$$\Delta_0^P = \Sigma_0^P = \Pi_0^P = \mathrm{P},$$

and, for all $k \geq 0$,

$$\Delta_{k+1}^P = \mathrm{P}^{\Sigma_k^P}, \quad \Sigma_{k+1}^P = \mathrm{NP}^{\Sigma_k^P}, \quad \text{and} \quad \Pi_{k+1}^P = \text{co-}\Sigma_{k+1}^P.$$

Here, P is the class of all problems solvable on a deterministic Turing machine in polynomial time; NP is similarly defined but using a nondeterministic Turing machine as underlying computing model; and, for complexity classes C and A, the notation C^A stands for the *relativised version* of C, consisting of all problems which can be decided by Turing machines of the same sort and time bound as in C, only that the machines have access to an oracle for problems in A. As well, co-C is the class of all problems which are complementary to the problems in C. We note that $\mathrm{NP} = \Sigma_1^P$, co-NP $= \Pi_1^P$, and $\mathrm{P} = \Delta_1^P$.

The *cumulative polynomial hierarchy* is given by the union $\bigcup_{k=0}^{\infty} \Sigma_k^P$. We say that a problem is located *at the kth level of the polynomial hierarchy* iff it is contained in Δ_{k+1}^P and it is either Σ_k^P-hard or Π_k^P-hard.

A further relevant family of complexity classes is given by the sequence of classes D_k^P, $k \geq 1$, where each D_k^P consists of all problems expressible as the conjunction of a problem in Σ_k^P and a problem in Π_k^P. Notice that, for all $k \geq 1$, $\Sigma_k^P \subseteq \mathrm{D}_k^P \subseteq \Sigma_{k+1}^P$ holds; in fact, both inclusions are widely conjectured to be strict. Moreover, any problem in D_k^P can be solved with two Σ_k^P oracle calls, and is thus intuitively easier than a problem complete for Δ_{k+1}^P.

The classes Σ_k^P and Π_k^P are closely related to the evaluation problem of QBFs—in particular, to QBFs which are given in *prenex normal form*: A QBF Φ is in prenex normal form iff it is of the form

$$\mathsf{Q}_1 P_1 \mathsf{Q}_2 P_2 \ldots \mathsf{Q}_n P_n \, \phi,$$

where ϕ is a propositional formula, $\mathsf{Q}_i \in \{\exists, \forall\}$ such that $\mathsf{Q}_i \neq \mathsf{Q}_{i+1}$ for $1 \leq i \leq n-1$, and P_i are disjoint sets of propositional variables for $1 \leq i \leq n$. If $\mathsf{Q}_1 = \exists$, then Φ is called an $(n, \exists)$-*QBF*, and if $\mathsf{Q}_1 = \forall$, then Φ is called an $(n, \forall)$-*QBF*. Without going into details, we mention that any QBF is easily transformed into an equivalent QBF in prenex normal form (by applying, among other reduction steps, the equivalences depicted in Proposition 1).

Proposition 2. *For every $k \geq 0$, we have that*

1. *deciding the truth for closed $(k, \exists)$-QBFs is Σ_k^P-complete, and*
2. *deciding the truth for closed $(k, \forall)$-QBFs is Π_k^P-complete.*

These complexity results are central for our subsequent encodings. In particular, we are interested in representing a given paraconsistent inference relation $\vdash_{\mathrm{P}}$ via a QBF-encoding $\mathcal{T}_{\mathrm{P}}[\cdot;\cdot]$ such that

1. $\mathcal{T}_{\mathrm{P}}[\cdot;\cdot]$ is *faithful*, i.e., for each theory T and each formula φ, $T \vdash_{\mathrm{P}} \varphi$ iff $\mathcal{T}_{\mathrm{P}}[T;\varphi]$ evaluates to true,
2. $\mathcal{T}_{\mathrm{P}}[T;\varphi]$ is computable in polynomial time, for each theory T and each formula φ, and
3. determining the truth values of the QBFs resulting from $\mathcal{T}_{\mathrm{P}}[\cdot;\cdot]$ is not computationally harder than checking inference under $\vdash_{\mathrm{P}}$.

The translation $\mathcal{T}_{\mathrm{P}}[\cdot;\cdot]$ is then called an *adequate translation*. For instance, if checking $T \vdash_{\mathrm{P}} \varphi$, for a given theory T and a given formula φ, is known to be in complexity class Σ_2^P, our desired translation $\mathcal{T}_{\mathrm{P}}[T;\varphi]$ should lead, for each T and φ, to a $(2,\exists)$-QBF, i.e., a QBF with at most one quantifier alternation, whose size is polynomial in the size of T and φ.

2.5 Basic QBF-Modules

We next discuss how QBFs can be employed to express some fundamental reasoning tasks concerning the consistency of propositional theories. Computing tasks of this kind will be required frequently throughout the paper as subtasks for other problems. Hence, the "modules" discussed in this section play the role of "building blocks" for the subsequent encodings of different paraconsistent reasoning tasks.

Expressing consistency. First of all, since existential quantification refers to satisfiability, we are easily capable to decide whether a given theory W is consistent, i.e., whether $W \not\models \bot$. Indeed, simply define

$$\mathcal{C}\!ons[W] = \exists P(\bigwedge_{\psi \in W} \psi),$$

where $P = var(W)$. Hence, $\mathcal{C}\!ons[W]$ is always closed, and the following relation is easily seen:

Proposition 3. *A theory W is consistent iff $\mathcal{C}\!ons[W]$ evaluates to true.*

We now extend this simple module as follows. Assume we have given two propositional theories, W and R, and we want to identify all subsets $S \subseteq R$ such that $W \cup S$ is consistent, i.e., our task is to compute all subsets of R consistent with W.

The basic idea is to use new atoms such that the truth assignments to these atoms correspond to the possible subsets of R. More precisely, let $G = \{g_\phi \mid \phi \in R\}$ be a set of new variables, not occurring in W or R. Variables from G are called "guessing variables", since they are used to guess a certain subset of R.

Consider the following encoding:

$$\mathcal{C}\!ons^G[W;R] = \exists P\Big(\bigwedge_{\psi \in W} \psi \wedge \bigwedge_{\phi \in R} (g_\phi \rightarrow \phi)\Big),$$

where P consists of all variables occurring in R or W. Observe that we now have an *open* QBF where the guessing variables G are free. The relation between subsets of R which are consistent with W and models of $\mathcal{C}ons^G[W;R]$ is a one-to-one correspondence, as desired:

Proposition 4. *Let W and R be theories, and $G = \{g_\phi \mid \phi \in R\}$ a set of variables not occurring in W or R. Moreover, let $S \subseteq R$ and $I \subseteq G$ such that, for each $\phi \in R$, $\phi \in S$ iff $g_\phi \in I$.*

Then, $W \cup S$ is consistent iff $\mathcal{C}ons^G[W;R]$ is true under I.

Example 1. Consider $W = \{\neg p \vee \neg q\}$ and $R = \{p, q\}$. All proper subsets of R are consistent with W, but $W \cup R$ is inconsistent. For R as given, we choose

$$G = \{g_p, g_q\}$$

as corresponding set of guessing variables.

Consider now the encoding $\mathcal{C}ons^G[W;R]$, given by

$$\exists pq \Big((\neg p \vee \neg q) \wedge (g_p \rightarrow p) \wedge (g_q \rightarrow q) \Big). \tag{7}$$

It can be checked that all interpretations $I \subset G$ are models of (7), but the interpretation $I = G$ is not a model of (7). This coincides with the observation that exactly the proper subsets of R, viz. $S_1 = \emptyset$, $S_2 = \{p\}$, and $S_3 = \{q\}$, are consistent with W, while $S_4 = \{p, q\}$ is not.

Expressing maximal consistent subsets. We also require to express the *maximal* subsets of R which are consistent (with some W). For instance, in the above example, we should rule out the subset $S_1 = \emptyset$, since $S_1 \subset S_2$ (as well as $S_1 \subset S_3$) and thus S_1 is not maximal.

Formally, a subset S of R is *maximal consistent* (*with* W) iff S is consistent (with W) and each S' with $S \subset S'$ is inconsistent (with W). Due to the monotonicity of classical propositional logic, the following characterisation is equivalent:

Proposition 5. *Let W and R be theories, and $S \subseteq R$.*

Then, S is maximal consistent with W iff

1. *$W \cup S$ is consistent, and*
2. *for each $\phi \in (R \setminus S)$, $W \cup S \cup \{\phi\}$ is inconsistent.*

We express these tests as follows: For any theories W and R, let $G = \{g_\phi \mid \phi \in R\}$ be a set of variables such that $G \cap var(W \cup R) = \emptyset$. Then, define

$$\mathcal{C}ons^G_{max}[W;R] = \mathcal{C}ons^G[W;R] \wedge \bigwedge_{\phi \in R} \Big(\neg g_\phi \rightarrow \neg \mathcal{C}ons^{G \setminus \{g_\phi\}}[W \cup \{\phi\}; R \setminus \{\phi\}] \Big).$$

Intuitively, $\mathcal{C}ons^G_{max}[W;R]$ guesses a subset S of R (via atoms G). With the first conjunct $\mathcal{C}ons^G[W;R]$, it is checked whether the guess is consistent with W. The second conjunct checks maximality for S as follows: For each $\phi \in R$, if ϕ is contained in the guess (i.e., if g_ϕ is true), we are immediately done. Otherwise, $\neg \mathit{Cons}^{G\setminus\{g_\phi\}}[W \cup \{\phi\}; R \setminus \{\phi\}]$ must evaluate to true. Observe that we use the same set G in this module (except for removing ϕ, which itself is "added" to the first argument W) as in the previous test. Hence, we check whether S is *not* consistent with $W \cup \{\phi\}$. This coincides precisely with the second condition in Proposition 5.

The formal result is as follows:

Proposition 6. *Let W and R be theories, and $G = \{g_\phi \mid \phi \in R\}$ a set of variables not occurring in W or R. Moreover, let $S \subseteq R$ and $I \subseteq G$ such that, for each $\phi \in R$, $\phi \in S$ iff $g_\phi \in I$.*

Then, S is maximal consistent with W iff $\mathit{Cons}^G_{max}[W; R]$ is true under I.

Expressing minimal models. Besides the selection of maximal subsets satisfying a certain criterion, it is sometimes also necessary to characterise subsets which are *minimal* with respect to a specific condition. Indeed, a widely-used method in nonmonotonic reasoning is inference based on *minimal models*. In such an approach, the inference relation is specified not in terms of all models of a given theory but only in terms of models which are minimal with respect to a certain ordering. Following the seminal work of McCarthy [46], minimal-model reasoning can be expressed in terms of a schema of second-order logic, known as *circumscription schema* (or *circumscription* for short). However, in the propositional case, instances of the circumscription schema are actually nothing else than specific QBFs. In the following, we characterise models which are minimal with respect to a specific ordering in terms of a QBF module corresponding to propositional circumscription.

Let T be a theory and (P, Q, Z) a partition of $\mathit{var}(T)$. Assume two models I and I' of T, and define $I \leq_{P;Z} I'$ iff the following conditions are satisfied:

1. $\{q \in Q \mid v_I(q) = t\} = \{q \in Q \mid v_{I'}(q) = t\}$;
2. $\{p \in P \mid v_I(p) = t\} \subseteq \{p \in P \mid v_{I'}(p) = t\}$.

A model I of T is called $(P; Z)$-*minimal* if no model I' of T with $I' \neq I$ satisfies $I' \leq_{P;Z} I$.

Informally, the partition (P, Q, Z) can be interpreted as follows: The set P contains the variables to be minimised, Z are those variables that can vary in minimising P, and the remaining variables Q are fixed in minimising P.

For a theory T and a partition (P, Q, Z) of $\mathit{var}(T)$, where $P = \{p_1, \ldots, p_n\}$ and $Z = \{z_1, \ldots, z_m\}$, we define the QBF $\mathit{Circ}[T; P; Z]$, called the (*parallel*) *circumscription* (*schema*) *of P in T*, as

$$T \wedge \forall \tilde{P} \forall \tilde{Z} \Big(\big(T\{P/\tilde{P}, Z/\tilde{Z}\} \wedge \bigwedge_{1 \leq i \leq n} (\tilde{p}_i \rightarrow p_i) \big) \rightarrow \bigwedge_{1 \leq i \leq n} (p_i \rightarrow \tilde{p}_i) \Big),$$

where $\tilde{P} = \{\tilde{p}_1, \ldots, \tilde{p}_n\}$ and $\tilde{Z} = \{\tilde{z}_1, \ldots, \tilde{z}_m\}$ are sets of new variables corresponding to P and Z, respectively, and $T\{P/\tilde{P}, Z/\tilde{Z}\}$ results from T by uniform substitution of the variables in $\tilde{P} \cup \tilde{Z}$ for those in $P \cup Z$.

Now, the main property of $\mathit{Circ}[T; P; Z]$ is given by the following result:

Proposition 7 ([46]). *Let T be a theory, (P, Q, Z) a partition of $var(T)$, and $I \subseteq var(T)$.*

Then, I is a $(P; Z)$-minimal model of T iff I is a model of $\mathcal{C}irc[T; P; Z]$.

Derivability testing. Finally, we define further modules for expressing derivability. Recall that, for any theory T and any propositional formula φ, it holds that $T \models \varphi$ iff $T \cup \{\neg\varphi\}$ is inconsistent. We thus define

$$\mathcal{D}eriv[W; \varphi] = \neg\mathcal{C}ons[W \cup \{\neg\varphi\}]$$

and obtain the following property:

Proposition 8. *For any theory W and any formula φ, $W \models \varphi$ iff $\mathcal{D}eriv[W; \varphi]$ is valid.*

More generally, defining

$$\mathcal{D}eriv^G[W; R; \varphi] = \neg\mathcal{C}ons^G[W \cup \{\neg\varphi\}; R]$$

yields the following characterisation:

Proposition 9. *Let W and R be theories, φ a formula, and $G = \{g_\phi \mid \phi \in R\}$ a set of variables not occurring in W, R, or φ. Moreover, let $S \subseteq R$ and $I \subseteq G$ such that, for each $\phi \in R$, $\phi \in S$ iff $g_\phi \in I$.*

Then, $W \cup S \models \varphi$ iff $\mathcal{D}eriv^G[W; R; \varphi]$ is true under I.

3 QBFs for Paraconsistent Reasoning: Case Studies

In this section, we show how QBFs can be successfully used to express different families of paraconsistent inference relations, exploiting the basic QBF modules introduced above. We first deal with formalisms based on maximal-consistent subsets. Afterwards, in Section 3.2, we discuss a class of inference relations using a consistency-driven rewriting technique based on Reiter's default logic. Finally, Section 3.3 is devoted to approaches using minimal-model reasoning in many-valued logics.

3.1 Reasoning from Maximal-Consistent Subsets

A simple but very popular approach to reasoning from an inconsistent knowledge base is reasoning from consistent subsets [59, 58, 14, 51, 7, 8]. Consider an inconsistent knowledge base in the form of a theory T:

$$\phi \wedge (\psi \rightarrow \varphi); \tag{8}$$

$$\psi \wedge \neg\phi; \tag{9}$$

$$(\psi \wedge \varphi) \rightarrow \eta; \tag{10}$$

$$(\psi \wedge \neg\eta) \vee \neg\phi; \tag{11}$$

$$\phi \vee \psi \vee \varphi \vee \eta. \tag{12}$$

Clearly, this theory is inconsistent. One way to proceed is to consider the maximal consistent subsets of T, which are:

$$S = \{(8), (10), (12)\};$$
$$S' = \{(8), (11), (12)\};$$
$$S'' = \{(9), (10), (11), (12)\}.$$

Let us see what follows from these maximal consistent subsets of T:

- S entails ϕ and $\psi \rightarrow (\varphi \wedge \eta)$.
- S' entails ϕ and $\psi \wedge \varphi \wedge \neg\eta$.
- S'' entails $\neg\phi$ and ψ, as well as $\varphi \rightarrow \eta$.

Among the most cautious conclusions are those formulas that follow from the intersection of S, S', and S'':

$$\phi \vee \psi \vee \varphi \vee \eta.$$

Definition 4. *Let T be a theory. A formula φ is a* free consequence of T, *symbolically $T \vdash_{\mathrm{FREE}} \varphi$, iff φ is entailed by the intersection of all maximal consistent subsets of T.*

In the above example, $\phi \vee \psi \vee \varphi \vee \eta \vee \chi$ is a free consequence of T, for any formula χ. By contrast, $\phi \vee \psi$ is not a free consequence of T even though $\phi \vee \psi$ is entailed by S and similarly by S' as well as by S''. Thus, free consequences need not be very informative and other notions have been introduced in the literature.

According to [20], a systematic account of reasoning from consistent subsets arises from distinguishing between selection mechanisms (among consistent subsets) and reasoning principles (to be applied to the selected consistent subsets).

In the general case, T is a *prioritised theory*, which means that it comes in the form $T = T_1 \cup \cdots \cup T_n$ (possibly, $n = 1$), where each T_i is a stratum such that strata with lower index contain formulas of greater importance. We assume the T_i's to be disjoint whereas not all authors do so. Here, we use the partition requirement in order to keep things simple. A *subtheory* of a prioritised theory T is of the form $S = S_1 \cup \cdots \cup S_n$ such that $S_i = T_i \cap S$ for $i = 1, \ldots, n$. Moreover, the *level* of a subtheory of T is defined by $a(S) = \min\{i \in \{1, \ldots, n\} \mid S_i \neq T_i\}$.

Definition 5. *Given $T = T_1 \cup \cdots \cup T_n$, we define the orderings $\ll^{\mathrm{T}}$ ("subtheory-based preference"), $\ll^{\mathrm{BO}}$ ("best-out preference"), and $\ll^{\mathrm{INCL}}$ ("inclusion-based preference") as follows, where $S = S_1 \cup \cdots \cup S_n$ and $S' = S'_1 \cup \cdots \cup S'_n$ range over the set of all consistent subtheories of T:*

- *$S \ll^{\mathrm{T}} S'$ iff $S \subset S'$;*
- *$S \ll^{\mathrm{BO}} S'$ iff $a(S) < a(S')$; and*
- *$S \ll^{\mathrm{INCL}} S'$ iff there exists some $k \in \{1, \ldots, n\}$ such that $S_k \subset S'_k$ and $S_i = S'_i$, for all $i \in \{1, \ldots, k-1\}$.*

Then, a consistent subtheory of T is σ-preferred *iff it is maximal with respect to $\ll^\sigma$, where σ ranges over $\{\text{T}, \text{BO}, \text{INCL}\}$. Also, $\sigma(T)$ denotes the set of all σ-preferred subtheories of T.*

Considering that all formulas in the above example form the unique stratum of T, we get that $\sigma(T) = \{S, S', S''\}$ in all three cases for σ (i.e., T, BO, and INCL). A more interesting situation is T being stratified, e.g., as follows:

$$T_1 = \{(8), (9)\}; \qquad T_2 = \{(10)\}; \qquad T_3 = \{(11), (12)\}.$$

Clearly, introducing strata cannot alter $\ll^{\text{T}}$, and the T-preferred subtheories of T are still as above: $\{S, S', S''\}$. Although $\ll^{\text{BO}}$ depends in general on strata, it happens here that the BO-preferred subtheories of T are also the same. The INCL-preferred subtheories of T are just S and S''.

Definition 6. *Let $T = T_1 \cup \cdots \cup T_n$ be a prioritised theory, φ a propositional formula, and $\sigma \in \{\text{T}, \text{BO}, \text{INCL}\}$. Then,*

- *φ is an* EXI-σ consequence of T, *written $T \vdash_{\text{EXI-}\sigma} \varphi$, iff $\varphi \in \bigcup_{S \in \sigma(T)} Cn(S)$,*
- *φ is a* UNI-σ consequence of T, *written $T \vdash_{\text{UNI-}\sigma} \varphi$, iff $\varphi \in \bigcap_{S \in \sigma(T)} Cn(S)$, and*
- *φ is an* ARG-σ consequence of T, *written $T \vdash_{\text{ARG-}\sigma} \varphi$, iff $\varphi \in \bigcup_{S \in \sigma(T)} Cn(S)$ but $\neg\varphi \notin \bigcup_{S \in \sigma(T)} Cn(S)$.*

Considering that all formulas in our example form the unique stratum of T, we get that $\phi \vee \psi$ is a UNI-T consequence of T, whereas $\psi \wedge \neg\phi$ is an EXI-T consequence of T because $\psi \wedge \neg\phi$ is entailed by S'' even though it is neither entailed by S nor S'. However, $\psi \wedge \neg\phi$ fails to be an ARG-T consequence of T. A reason is that ϕ (from which $\neg(\psi \wedge \neg\phi)$ is classically deduced) is entailed by S, and analogously by S'. An example of an ARG-T consequence of T is ψ.

Assume now that T is equipped with the stratification given above. UNI-T consequences and UNI-BO consequences are the same as in the non-stratified case. On the other hand, $(\psi \wedge \varphi) \rightarrow \eta$ is a new UNI-INCL consequence. Moreover, $\phi \wedge \psi \wedge \neg\eta$ is no longer an EXI-INCL consequence. Accordingly, $(\psi \wedge \varphi) \rightarrow \eta$ is a new ARG-INCL consequence.

All these notions compare, by way of set-inclusion of the respective sets of consequences of a given theory, as depicted in Figure 1 (cf. also [20]).

Hence, the free consequences of a given theory T comprise the smallest set of consequences of T and the set of EXI-T consequences is the largest (apart from the classical consequences $Cn(T)$).

Other notions have been defined as well, either in the non-prioritised case or in the prioritised case, most of them technically involved.

The complexity of checking EXI-σ, UNI-σ, and ARG-σ consequences, for $\sigma \in \{\text{T}, \text{BO}, \text{INCL}\}$, was analysed in [19]. There, the following results were shown: The problem of deciding whether a formula is an EXI-σ consequence of a given theory is Σ_2^P-complete for $\sigma \in \{\text{T}, \text{BO}, \text{INCL}\}$. The corresponding problem for UNI-T and UNI-INCL consequences is Π_2^P-complete, while for UNI-BO it is known to be in Δ_2^P. As for ARG-σ, the problem is in Δ_3^P, for each $\sigma \in \{\text{T}, \text{BO}, \text{INCL}\}$.

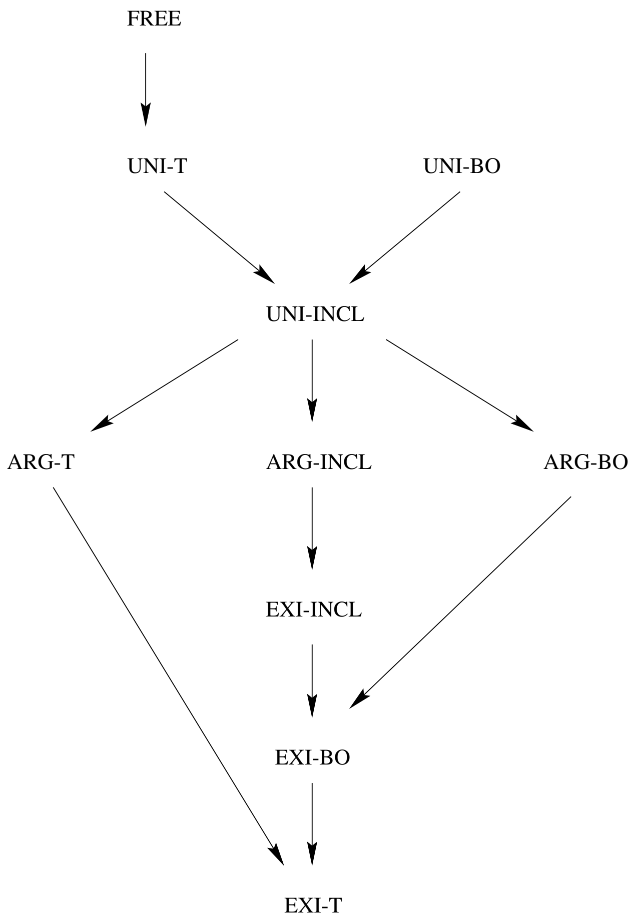

Fig. 1. Relations between different paraconsistent inference relations based on maximal subsets

Encodings. From our considerations in Section 2.5, it is quite easy to construct QBF encodings for expressing EXI-T, UNI-T, and ARG-T consequences. Indeed, it just suffices to combine the modules $\mathcal{C}ons^G_{max}[\cdot;\cdot]$ and $\mathcal{D}eriv^G[\cdot;\cdot]$ in a suitable manner, for a set G of guessing variables. More precisely, given a theory T and a formula φ, we use simultaneously the modules $\mathcal{C}ons^G_{max}[\emptyset;T]$ and $\mathcal{D}eriv^G[\emptyset;T;\varphi])$ to check whether a guess for a subset $S \subseteq T$ is maximal consistent and whether φ is entailed by S, respectively. Observe that the *same set* G of guessing variables is used for expressing both tasks. If there exists at least one interpretation $I \subseteq G$ making both $\mathcal{C}ons^G_{max}[\emptyset;T]$ and $\mathcal{D}eriv^G[\emptyset;T;\varphi])$ true, we directly get an encoding for EXI-T consequences. If under each $I \subseteq G$ which is a model of $\mathcal{C}ons^G_{max}[\emptyset;T]$, also $\mathcal{D}eriv^G[\emptyset;T;\varphi])$ is true, then we have an encoding for UNI-T consequences.

Finally, ARG-T consequences are easily encoded via two independent tests for checking EXI-T inference.

Theorem 1. *Let T be a theory, φ a formula, and $G = \{g_\phi \mid \phi \in T\}$ a set of new guessing atoms not occurring in T or φ. Then,*

1. *$T \vdash_{\text{EXI-T}} \varphi$ iff $\mathcal{T}_{\text{EXI-T}}[T;\varphi] = \exists G\big(\mathcal{C}ons^G_{max}[\emptyset;T] \land \mathcal{D}eriv^G[\emptyset;T;\varphi]\big)$ is valid,*
2. *$T \vdash_{\text{UNI-T}} \varphi$ iff $\mathcal{T}_{\text{UNI-T}}[T;\varphi] = \forall G\big(\mathcal{C}ons^G_{max}[\emptyset;T] \rightarrow \mathcal{D}eriv^G[\emptyset;T;\varphi]\big)$ is valid, and*
3. *$T \vdash_{\text{ARG-T}} \varphi$ iff $\mathcal{T}_{\text{ARG-T}}[T;\varphi] = \mathcal{T}_{\text{EXI-T}}[T,\varphi] \land \neg\mathcal{T}_{\text{EXI-T}}[T,\neg\varphi]$ is valid.*

Observe that the size of each of the above encodings is clearly polynomial in the size of T and φ. Hence, each of the encodings is computable in polynomial time. Furthermore, it is easy to check that $\mathcal{T}_{\text{EXI-T}}[T;\varphi]$ can be transformed in polynomial time into a $(2,\exists)$-QBF, whilst $\mathcal{T}_{\text{UNI-T}}[T;\varphi]$ can be transformed, likewise in polynomial time, into a $(2,\forall)$-QBF, for each T and φ. Therefore, recalling that checking EXI-T and UNI-T consequence is complete for Σ_2^P and Π_2^P, respectively, both $\mathcal{T}_{\text{EXI-T}}[\cdot;\cdot]$ and $\mathcal{T}_{\text{UNI-T}}[\cdot;\cdot]$ are adequate.

Concerning $\mathcal{T}_{\text{ARG-T}}[T;\varphi]$, since this encoding can be transformed in polynomial time into an equivalent QBF which is the conjunction of a $(2,\exists)$-QBF and a $(2,\forall)$-QBF, for each T and φ, it follows that checking ARG-T consequence is not only in Δ_3^P but actually in the easier class D_2^P.

Next, we consider the notion of free consequence. To this end, we call, for a given theory T, the set of all $\phi \in T$ which are a UNI-T consequence of T the *free base* of T.

The following property is also observed in [7].

Proposition 10. *Let T be a theory. Then, a formula φ is a free consequence of T iff φ is classically entailed by the free base of T.*

Note that, by definition, the free base of T is given by $T \cap \bigcap_{S\in\mathrm{T}(T)} Cn(S)$. Hence, Proposition 10 expresses that $T \vdash_{\text{FREE}} \varphi$ just in case $T \cap \bigcap_{S\in\mathrm{T}(T)} Cn(S) \models \varphi$. By the properties of $Cn(\cdot)$, this in turn entails that $T \vdash_{\text{FREE}} \varphi$ only if $\varphi \in \bigcap_{S\in\mathrm{T}(T)} Cn(S)$, which rephrases the relation that every free consequence of T is a UNI-T consequence of T, as depicted in Figure 1.

Theorem 2. *Let T be a theory, φ a formula, and $G = \{g_\phi \mid \phi \in T\}$ a set of new guessing variables.*

Then, $T \vdash_{\text{FREE}} \varphi$ iff

$$\mathcal{T}_{\text{FREE}}[T;\varphi] = \forall G\Big(\bigwedge_{\phi\in T} \big(\mathcal{T}_{\text{UNI-T}}[T,\phi] \rightarrow g_\phi\big) \rightarrow \mathcal{D}eriv^G[\emptyset;T;\varphi]\Big)$$

is valid.

We now turn our attention to the other approaches considered, where prioritised theories are used to realise a more fine-grained selection mechanism among consistent subsets. As it turns out, the basic reasoning principles EXI-σ, UNI-σ, and ARG-σ are encoded along the lines of Theorem 1, but we have to replace the module $\mathcal{C}ons^G_{max}[T]$ in an appropriate way.

We start with the consequence relations based on best-out preference. The encoding relies on the following proposition.

Proposition 11. *Let $S = S_1 \cup \ldots \cup S_n$ be a consistent subset of a prioritised theory $T = T_1 \cup \ldots \cup T_n$, with $S_i = S \cap T_i$.*

Then, S is BO-preferred iff $T_1 \cup \ldots \cup T_{a(S)}$ is inconsistent or $S = T$.

This motivates the subsequent encoding, which works as follows. First, $\mathcal{C}ons^G[\emptyset; T]$ yields all consistent subsets of T via the guessing variables G. By the above result, we have that T is BO-preferred whenever T is consistent. Hence, if each $g_\phi \in G$ is assigned to true, we are done. Otherwise, for a guessed subset $S \subset T$, the encoding checks, for $i = 1, \ldots, n$, that whenever i is the level, $a(S)$, of S, then $T_1 \cup \ldots \cup T_{a(S)}$ is inconsistent. Recall that the level of a subtheory S of T is defined by $a(S) = \min\{j \in \{1, \ldots, n\} \mid S_j \neq T_j\}$.

Lemma 1. *Let $T = T_1 \cup \ldots \cup T_n$ be a prioritised theory and $G = G_1 \cup \ldots \cup G_n = \{g_\phi \mid \phi \in T\}$ a set of corresponding guessing variables. Moreover, let $S \subseteq T$ and $I \subseteq G$ such that, for each $\phi \in T$, $\phi \in S$ iff $g_\phi \in I$.*

Then, S is BO-preferred iff $\mathcal{BO}^G[T]$, given by

$$\mathcal{C}ons^G[\emptyset; T] \wedge \Big(\neg G \rightarrow \bigwedge_{i=1,\ldots,n} \big((G_1 \wedge \ldots \wedge G_{i-1} \wedge \neg G_i) \rightarrow \neg\mathcal{C}ons[T_1 \cup \ldots \cup T_i]\big)\Big),$$

is true under I.

In accord to Theorem 1, we obtain the following encodings for expressing the relations $\vdash_{\text{EXI-BO}}$, $\vdash_{\text{UNI-BO}}$, and $\vdash_{\text{ARG-BO}}$, respectively, by replacing the module $\mathcal{C}ons^G_{max}[\emptyset; T]$ by $\mathcal{BO}^G[T]$ in the corresponding translations.

Theorem 3. *Let T be a prioritised theory, φ a formula, and $G = \{g_\phi \mid \phi \in T\}$ a set of new guessing atoms not occurring in T or φ. Then,*

1. *$T \vdash_{\text{EXI-BO}} \varphi$ iff $\mathcal{T}_{\text{EXI-BO}}[T; \varphi] = \exists G\big(\mathcal{BO}^G[T] \wedge \mathcal{D}eriv^G[\emptyset; T; \varphi]\big)$ is valid,*
2. *$T \vdash_{\text{UNI-BO}} \varphi$ iff $\mathcal{T}_{\text{UNI-BO}}[T; \varphi] = \forall G\big(\mathcal{BO}^G[T] \rightarrow \mathcal{D}eriv^G[\emptyset; T; \varphi]\big)$ is valid, and*
3. *$T \vdash_{\text{ARG-BO}} \varphi$ iff $\mathcal{T}_{\text{ARG-BO}}[T; \varphi] = \mathcal{T}_{\text{EXI-BO}}[T, \varphi] \wedge \neg\mathcal{T}_{\text{EXI-BO}}[T, \neg\varphi]$ is valid.*

Similar as for ARG-T consequences, the encoding $\mathcal{T}_{\text{ARG-BO}}[\cdot; \cdot]$ yields that checking ARG-BO consequences lies in the easier subclass D^P_2 of Δ^P_3. Moreover, although the encoding $\mathcal{T}_{\text{EXI-BO}}[\cdot; \cdot]$ is adequate, $\mathcal{T}_{\text{UNI-BO}}[\cdot; \cdot]$ is not because checking UNI-BO consequences is in Δ^P_2 but $\mathcal{T}_{\text{UNI-BO}}[T; \varphi]$ can be transformed in polynomial time into an equivalent $(2, \forall)$-QBF, for any T and φ. However, we can simplify $\mathcal{T}_{\text{UNI-BO}}[\cdot; \cdot]$ using the following observation:

Proposition 12. *For a prioritised theory $T = T_1 \cup \ldots \cup T_n$ and a formula φ, we have that $T \vdash_{\text{UNI-BO}} \varphi$ iff there exists some $i \in \{0, \ldots, n\}$ such that $T_1 \cup \ldots \cup T_i$ is consistent and $T_1 \cup \ldots \cup T_i \models \varphi$.*

Observe that the case $i = 0$ is required for dealing with the case where T_1 is already inconsistent. We thus obtain the following optimised encoding for checking UNI-BO consequences, avoiding explicit quantifier alternations:

Theorem 4. *Let $T = T_1 \cup \ldots \cup T_n$ be a prioritised theory, and φ a formula. Then, $T \vdash_{\text{UNI-BO}} \varphi$ iff $\bigvee_{i=0,\ldots,n} \big(\mathcal{C}ons[T_1 \cup \ldots \cup T_i] \wedge \mathcal{D}eriv[T_1 \cup \ldots \cup T_i; \varphi]\big)$ is valid.*

Finally, we define a module for expressing the INCL-preferred subsets of a given prioritised theory. The following result is the basis for this module, albeit other characterisations are also possible.

Proposition 13. *Given a consistent subtheory $S = S_1 \cup \ldots \cup S_n$ of a prioritised theory $T = T_1 \cup \ldots \cup T_n$, it holds that S is INCL-preferred iff, for each $i \in \{1, \ldots, n\}$ and each $\phi \in T_i \setminus S$, $S_1 \cup \ldots \cup S_i \cup \{\phi\}$ is inconsistent.*

This leads to the following encoding:

Lemma 2. *Let $T = T_1 \cup \ldots \cup T_n$ be a prioritised theory and $G = G_1 \cup \ldots \cup G_n = \{g_\phi \mid \phi \in T\}$ a set of corresponding guessing atoms. Moreover, let $S \subseteq T$ and $I \subseteq G$ such that, for each $\phi \in T$, $\phi \in S$ iff $g_\phi \in I$.*

Then, S is INCL-preferred iff

$$\mathcal{I}ncl^G[T] = \mathcal{C}ons^G[\emptyset; T] \wedge \bigwedge_{i=1,\ldots,n,\phi \in T_i} \big(\neg g_\phi \rightarrow \neg \mathcal{C}ons^{G^i_\phi}[\{\phi\}, T^i_\phi]\big)$$

is true under I, where $G^i_\phi = (G_1 \cup \ldots \cup G_i) \setminus \{g_\phi\}$ and $T^i_\phi = (T_1 \cup \ldots \cup T_i) \setminus \{\phi\}$.

Again, the encodings for checking EXI-INCL, UNI-INCL, and ARG-INCL consequences, respectively, follow the same pattern as for the previous variants.

Theorem 5. *Let T be a prioritised theory, φ a formula, and $G = \{g_\phi \mid \phi \in T\}$ a set of new guessing atoms not occurring in T or φ. Then,*

1. *$T \vdash_{\text{EXI-INCL}} \varphi$ iff $\mathcal{T}_{\text{EXI-INCL}}[T; \varphi] = \exists G\big(\mathcal{I}ncl^G[T] \wedge \mathcal{D}eriv^G[\emptyset; T; \varphi]\big)$ is valid,*
2. *$T \vdash_{\text{UNI-INCL}} \varphi$ iff $\mathcal{T}_{\text{UNI-INCL}}[T; \varphi] = \forall G\big(\mathcal{I}ncl^G[T] \rightarrow \mathcal{D}eriv^G[\emptyset; T; \varphi]\big)$ is valid, and*
3. *$T \vdash_{\text{ARG-INCL}} \varphi$ iff $\mathcal{T}_{\text{ARG-INCL}}[T; \varphi] = \mathcal{T}_{\text{EXI-INCL}}[T; \varphi] \wedge \neg \mathcal{T}_{\text{EXI-INCL}}[T; \neg\varphi]$ is valid.*

Analogous to the previous encodings we have that $\mathcal{T}_{\text{EXI-INCL}}[\cdot;\cdot]$ and $\mathcal{T}_{\text{UNI-INCL}}[\cdot;\cdot]$ are adequate, whilst $\mathcal{T}_{\text{ARG-INCL}}[\cdot;\cdot]$ exhibits that checking ARG-INCL consequences lies actually in D^P_2.

3.2 Signed Systems

The basic idea behind the approach taken by signed systems [11] is as follows. An inconsistent theory is transformed into a consistent one by renaming all literals occurring in the theory. Then, some of the original contents of the theory is restored by introducing progressively formal equivalences linking the original literals to their renamings. This is done as long as consistency is preserved. The overall approach provides us with a family of paraconsistent consequence relations.

For illustration, consider a theory containing the four statements

$$p,\ \neg p,\ q, (\neg q \vee r). \tag{13}$$

Clearly, this theory is inconsistent. We start with transforming the theory by renaming all of its literals:

$$p^+,\ p^-,\ q^+,\ (q^- \vee r^+).$$

The renamings indicate what renamed literals were denials of each other—making explicit whether the renamed literals were "positive" or "negative". In this way, we obtain a *signed* theory. Then, we restore some of the original contents of the theory by progressively introducing formal equivalences of the form $p^+ \equiv \neg p^-$, linking the original literals to their renamings. We do this up to the point where introducing any further equivalence would reinstate inconsistency. As a result, we can apply classical logic to reason from this signed theory extended with increasingly many equivalences (actually, the equivalences we use are slightly different because we deal at once with the signed and unsigned language). Then, a later interpretation of the signed formulas gets us back to the original language, classical inferences having thus been turned into seemingly paraconsistent ones.

The primary technical means for dealing with "signed theories" is *default logic* [57], whose central concepts are *default rules* along with their induced *extensions* of an initial set of premises. A default rule (or *default* for short) $\frac{\alpha:\beta}{\gamma}$ has two types of antecedents: a *prerequisite* α which is established if α is derivable and a *justification* β which is established if β is consistent. If both conditions hold, the *consequent* γ is concluded by default. For convenience, we denote the prerequisite of a default δ by $\mathit{prereq}(\delta)$, its justification by $\mathit{justif}(\delta)$, and its consequent by $\mathit{conseq}(\delta)$. Accordingly, for a set D of defaults, we define $\mathit{prereq}(D) = \{\mathit{prereq}(\delta) \mid \delta \in D\}$, $\mathit{justif}(D) = \{\mathit{justif}(\delta) \mid \delta \in D\}$, and $\mathit{conseq}(D) = \{\mathit{conseq}(\delta) \mid \delta \in D\}$.

A *default theory* is a pair (D, T) where D is a set of default rules and T is a set of propositional formulas. A set E of formulas is an *extension* of (D, T) iff $E = \bigcup_{n\in\omega} E_n$, where $E_1 = T$ and, for $n \geq 1$, $E_{n+1} = Cn(E_n) \cup \{\gamma \mid \frac{\alpha:\beta}{\gamma} \in D, \alpha \in E_n, \neg\beta \notin E\}$. We refer the reader for further details on default logic to the literature [57, 9].

The formal approach behind signed systems can then be described as follows. We start with a finite set of propositional formulas (i.e., a theory) T. Then, we proceed as follows. First, we transform T into conjunctive normal form (CNF).[3] This is a conjunction of disjunctions of literals, or simply a set of clauses. In this

[3] Such a transformation is not strictly necessary; see [11] on how this is avoided by distinguishing among positive and negative formula occurrences.

way, T is transformed into a finite set of clauses. It is worth noticing that this transformation does not affect the logical contents of the original theory.

Next, we rename the propositions in T as follows. Let φ be a formula in CNF. Then, we define $\varphi^{\pm}$ as the formula obtained from φ by replacing each occurrence of $\neg p$ by p^- and by replacing all remaining occurrences of p by p^+. In this way, we turn the initial theory T into the *consistent* theory $T^{\pm} = \{\phi^{\pm} \mid \phi \in T\}$. This is so because each formula ϕ of T is substituted by a formula $\phi^{\pm}$ which is always a *positive* formula.

Finally, we consider the default theory comprised of $T^{\pm}$ and a set of default rules $D_P = \{\delta_p \mid p \in P\}$, where P is a suitably chosen set of propositional atoms and

$$\delta_p = \frac{: p^+ \equiv \neg p^-}{(p \equiv p^+) \wedge (\neg p \equiv p^-)}, \tag{14}$$

for each $p \in P$. Intuitively, such default rules provide means for closing the gap between $T^{\pm}$ and T. That is, by checking whether the justification $p^+ \equiv \neg p^-$ is consistent, we test whether or not we can reintroduce the "law of (non-)contradiction" for the proposition p without getting an inconsistent theory. If this is the case, we "restore" the original meaning of the propositions p^+ and p^- by adding the equivalences $p \equiv p^+$ and $\neg p \equiv p^-$. Considering in turn each propositional letter p, we are thus gradually restoring the original contents of the theory—except that we stop at the borderline of inconsistency by leaving blank all propositions involved in genuine contradictions.

Consider the theory

$$\{p, \neg p, q, (q \to r)\}. \tag{15}$$

Transforming the elements of this theory into CNF yields the theory given in (13), i.e., $\{p, \neg p, q, (\neg q \vee r)\}$. Next, we rewrite this set of clauses into the consistent theory

$$\{p^+, p^-, q^+, (q^- \vee r^+)\} \tag{16}$$

by substituting $\neg p, \neg q$ by p^-, q^- and p, q, r by p^+, q^+, r^+, respectively.

We then proceed by adding, for each propositional atom occurring in the original theory, a corresponding default rule as defined in (14). This yields three default rules δ_p, δ_q, and δ_r, since the original theory is built from the propositional atoms p, q, and r. In full detail, $\delta_p, \delta_q, \delta_r$ have the following form:

$$\frac{: p^+ \equiv \neg p^-}{(p \equiv p^+) \wedge (\neg p \equiv p^-)}, \quad \frac{: q^+ \equiv \neg q^-}{(q \equiv q^+) \wedge (\neg q \equiv q^-)}, \quad \frac{: r^+ \equiv \neg r^-}{(r \equiv r^+) \wedge (\neg r \equiv r^-)}.$$

Consider the default theory obtained from theory (16) along with the three latter default rules:

$$\left(\{\delta_p, \delta_q, \delta_r\}, \{p^+, p^-, q^+, (q^- \vee r^+)\}\right). \tag{17}$$

Clearly, the first default rule is inapplicable, since its justification $p^+ \equiv \neg p^-$ is inconsistent in the presence of p^+ and p^-. In contrast, the second and the third default rule are applicable and consequently restore the original meaning of q^+, q^-, r^+ , and r^-.[4] Accordingly, we obtain a single extension containing the propositions q and r (from the alphabet of our inconsistent initial theory) along with p^+, p^-, q^+, r^+.

Using this definition, we define the first family of paraconsistent consequence relations based on signed theories:

Definition 7. *Let T be a theory, φ a propositional formula, and $\mathcal{E}$ the set of all extensions of $(D_P, T^{\pm})$. Moreover, for each set S of formulas and signed formulas, let $\Pi_S = \{\mathit{conseq}(\delta_p) \mid p \in P, \neg \mathit{justif}(\delta_p) \notin S\}$. Then,*

- *φ is a* credulous unsigned[5] consequence of T*, symbolically written as $T \vdash_c \varphi$, iff $\varphi \in \bigcup_{E \in \mathcal{E}} Cn(T^{\pm} \cup \Pi_E)$,*
- *φ is a* skeptical unsigned consequence of T*, symbolically written as $T \vdash_s \varphi$, iff $\varphi \in \bigcap_{E \in \mathcal{E}} Cn(T^{\pm} \cup \Pi_E)$, and*
- *φ is a* prudent unsigned consequence of T*, symbolically written as $T \vdash_p \varphi$, iff $\varphi \in Cn(T^{\pm} \cup \bigcap_{E \in \mathcal{E}} \Pi_E)$.*

For illustration, consider the inconsistent theory $T = \{p, q, \neg p \vee \neg q\}$. For obtaining the above paraconsistent consequence relations, T is turned into the default theory $(D_P, T^{\pm}) = \big(\{\delta_p, \delta_q\}, \{p^+, q^+, p^- \vee q^-\}\big)$. We obtain two extensions, viz. $Cn(T^{\pm} \cup \{\mathit{conseq}(\delta_p)\})$ and $Cn(T^{\pm} \cup \{\mathit{conseq}(\delta_q)\})$. The following relations show how the different consequence relations behave: on the one hand, we have $T \vdash_c p$, $T \not\vdash_s p$, and $T \not\vdash_p p$, but, on the other hand, for instance, it holds that $T \vdash_c p \vee q$, $T \vdash_s p \vee q$, and $T \not\vdash_p p \vee q$.

For a complement, the following "signed" counterparts are defined.

Definition 8. *Given the prerequisites of Definition 7, we say that*

- *φ is a* credulous signed consequence of T*, symbolically written as $T \vdash_c^{\pm} \varphi$, iff $\varphi^{\pm} \in \bigcup_{E \in \mathcal{E}} Cn(T^{\pm} \cup \Pi_E)$,*
- *φ is a* skeptical signed consequence of T*, symbolically written as $T \vdash_s^{\pm} \varphi$, iff $\varphi^{\pm} \in \bigcap_{E \in \mathcal{E}} Cn(T^{\pm} \cup \Pi_E)$, and*
- *φ is a* prudent signed consequence of T*, symbolically written as $T \vdash_p^{\pm} \varphi$, iff $\varphi^{\pm} \in Cn(T^{\pm} \cup \bigcap_{E \in \mathcal{E}} \Pi_E)$.*

As shown in [11], these relations compare to each other in the following way:

Proposition 14. *Let $C_i(T) = \{\varphi \mid T \vdash_i \varphi\}$ and similarly $C_i^{\pm}(T) = \{\varphi \mid T \vdash_i^{\pm} \varphi\}$, for $i \in \{p, s, c\}$. Then, we have*

1. *$C_i(T) \subseteq C_i^{\pm}(T)$, and*

[4] Notice that the contribution of a default rule like δ_r to the theory formation process is in no way sufficient for deriving r, even though it is a necessary condition. Applying δ_r merely re-establishes the original meaning of r and $\neg r$ from r^+ and r^-, respectively. In our example, r is derived from q and $\neg q \vee r$ due to the preceding restoration of q and r.

[5] The term "unsigned" indicates that only unsigned formulas are taken into account.

2. $C_p(T) \subseteq C_s(T) \subseteq C_c(T)$ *and* $C^{\pm}_p(T) \subseteq C^{\pm}_s(T) \subseteq C^{\pm}_c(T)$.

That is, signed derivability gives more conclusions than unsigned derivability, and within each series of consequence relations the strength of the relation is increasing. For a detailed formal elaboration, along with further refined consequence relations, we refer the reader to [11].

Encodings. In [12], it was shown that, given a theory T, the outcome of the different paraconsistent consequence relations solely depends on those defaults δ_p from D_P where p occurs in T. With a slight abuse of notation, in what follows we write D_T to denote this particular set of defaults for a given T.

The next result is of importance, since it leads us to a simple appealing encoding to compute the extensions of the kind of default theories under consideration.

Proposition 15. *Let T be a theory, $(D_T, T^{\pm})$ its corresponding default theory, and $C \subseteq D_T$.*

Then, $Cn(T^{\pm} \cup conseq(C))$ is an extension of $(D_T, T^{\pm})$ iff $justif(C)$ is maximal consistent with $T^{\pm}$.

Reconsider our example theory $T = \{p, \neg p, q, \neg q \vee r\}$ and its corresponding default theory (17), having $justif(D_T) = \{p^+ \equiv \neg p^-, q^+ \equiv \neg q^-, r^+ \equiv \neg r^-, \}$. It is quite easy to see that $T^{\pm} = \{p^+, p^-, q^+, q^- \vee r^+\}$ is not consistent with $p^+ \equiv \neg p^-$, but with $\{q^+ \equiv \neg q^-, r^+ \equiv \neg r^-\}$. Thus, $justif(\{\delta_q, \delta_r\})$ is the maximal subset of $justif(D)$ consistent with T. We thus get as single extension the deductive closure of $T^{\pm} \cup conseq(\{\delta_q, \delta_r\}) = T^{\pm} \cup \{q \equiv q^+, \neg q \equiv q^-, r \equiv r^+, \neg r \equiv r^-\}$ yielding $Cn(p^+, p^-, q^+, q, r^+, r)$.

Indeed, Theorem 15 gives us a suitable basis for the desired QBF-encodings which represent a more compact axiomatics than the encodings given in [27] for arbitrary default theories.

Theorem 6. *Let T be a theory, $(D_T, T^{\pm})$ its corresponding default theory, and $G = \{g_\delta \mid \delta \in D_T\}$ a set of new guessing variables. Moreover, let $C \subseteq D_T$ and $I \subseteq G$ such that, for each $\delta \in D_T$, $\delta \in C$ iff $g_\delta \in I$.*

Then, the set $Cn(T^{\pm} \cup conseq(C))$ is an extension of $(D_T, T^{\pm})$ iff the QBF $\mathcal{C}ons^G_{max}[T^{\pm}; justif(D_T)]$ is true under I.

Having a characterisation of the extensions in terms of models of QBFs, it is quite easy to decide the respective paraconsistent consequence relations. In particular, encodings for the relations $\vdash_c$, $\vdash^{\pm}_c$, $\vdash_s$, and $\vdash^{\pm}_s$ are obtained by combining, in a suitable way, the above encoding with the module for expressing derivability.

Theorem 7. *Let T be a theory, φ a formula, and $(D_T, T^{\pm})$ as before. Moreover, let $G = \{g_\delta \mid \delta \in D_T\}$ be a set of guessing variables. Then,*

1. *$T \vdash_c \varphi$ iff*

$$\mathcal{T}_c[T; \varphi] = \exists G\big(\mathcal{C}ons^G_{max}[T^{\pm}; justif(D_T)] \wedge \mathcal{D}eriv^G[T^{\pm}; conseq(D_T); \varphi]\big)$$

is valid,

2. *$T \vdash_s \varphi$ iff*

$$\mathcal{T}_s[T;\varphi] = \forall G\big(\mathcal{C}ons^G_{max}[T^{\pm}; justif(D_T)] \rightarrow \mathcal{D}eriv^G[T^{\pm}; conseq(D_T);\varphi]\big)$$

is valid,

3. *$T \vdash^{\pm}_c \varphi$ iff $\mathcal{T}^{\pm}_c[T;\varphi] = \mathcal{T}_c[T;\varphi^{\pm}]$ is valid, and*
4. *$T \vdash^{\pm}_s \varphi$ iff $\mathcal{T}^{\pm}_s[T;\varphi] = \mathcal{T}_s[T;\varphi^{\pm}]$ is valid.*

Observe that the sets D_T, $justif(D_T)$, and $conseq(D_T)$ have the same cardinality. Hence, in the above result, one set of guessing variables, G, is sufficient.

It remains to deal with the prudent consequence relations. To begin with, as pointed out in [11], the inference relation $\vdash_p$ captures the notion of free consequence. Hence, $\mathcal{T}_{\text{FREE}}[\cdot;\cdot]$ can be used as encoding for $\vdash_p$. However, for a more direct encoding of prudent consequence, we can show the following property:

Lemma 3. *Let T be a theory, $(D_T, T^{\pm})$ its corresponding default theory, and φ a formula.*

Then, the following conditions are equivalent:

1. *$T \vdash_p \varphi$;*
2. *for each $C \subseteq D_T$, if, for each $\delta \in D_T$, $T \vdash_s conseq(\delta)$ only if $\delta \in C$, then $T^{\pm} \cup conseq(C) \models \varphi$.*

This leads to the following encoding:

Theorem 8. *Let T be a theory, $(D_T, T^{\pm})$ its corresponding default theory, φ a formula, and $G = \{g_\delta \mid \delta \in D_T\}$ a set of new guessing variables.*

Then, $T \vdash_p \varphi$ iff

$$\forall G\big(\bigwedge_{\delta \in D_T} \big(\mathcal{T}_s[T^{\pm}; conseq(\delta)] \rightarrow g_\delta\big) \rightarrow \mathcal{D}eriv^G[T^{\pm}; conseq(D_T);\varphi]\big)$$

is valid.

An encoding for $T \vdash^{\pm}_p \varphi$ is easily obtained by replacing φ by $\varphi^{\pm}$ in the above encoding.

Similar to the paraconsistent inference relations based on maximal subsets, the complexity of the signed and unsigned inference relations is located at the second level of the polynomial hierarchy. This was shown in [12] on the basis of the above encodings, by inspecting the quantifier order of the resultant QBFs.[6] As well, the respective encodings are adequate.

3.3 Multi-Valued Approaches

The idea underlying the three-valued approaches to paraconsistent reasoning is to counterbalance the effect of contradictions by providing a third truth value,

---

[6] Incidentally, the complexity results for $\vdash_s$ and $\vdash^{\pm}_s$ have independently been obtained by Coste-Marquis and Marquis as well [22].

accounting for contradictory propositions. As already put forth in [55], this provides us with inconsistency-tolerating three-valued models. However, this approach turns out to be rather weak in that it invalidates certain classical inferences, even if there is no contradiction. Intuitively, this is because there are too many three-valued models, in particular those assigning the inconsistency-tolerating truth-value to propositions that are unaffected by contradictions. For instance, the three-valued logic LP [55] denies inference by disjunctive syllogism. That is, ψ is not derivable from the (consistent!) premise $(\phi \vee \psi) \wedge \neg\phi$. As pointed out in [22], this deficiency applies also to the closely related paraconsistent systems J_3 [26], L [44], and RP [33]. As a consequence, none of the aforementioned systems coincides with classical logic when reasoning from consistent premises.

The pioneering work to overcome this deficiency was done by Priest [56]. The key idea is to restrict the set of three-valued models by taking advantage of some preference criterion that aims at "minimising inconsistency". In this way, a "maximum" of a classically inconsistent knowledge base should be recovered. While minimisation is understood in Priest's seminal work [56], proposing his logic LP_m, as preferring three-valued models as close as possible to two-valued interpretations, the overall approach leaves room for different preference criteria. Another criterion is postulated in [10] by giving more importance to the given knowledge base. In this approach, one prefers three-valued models that are as similar as possible to two-valued models of the knowledge base in the sense that those models assign *true* to as many items of the knowledge base as possible. Furthermore, [40] considers cardinality-based versions of the last two preference criteria. Even more criteria are conceivable by distinguishing symbols having different importance.

Syntactically, we use propositional formulas in the standard way, but adopt the semantics as follows. A *three-valued interpretation*, M, is a function assigning to each atom a truth-value from $\{t, f, o\}$. Intuitively, the truth value o takes care for contradictory propositions. In general, the assignment of truth values to arbitrary formulas, given a three-valued interpretation M, is realised by means of a function $v_M(\cdot)$, which is specified according to the following truth tables, under the usual condition that $v_M(p) = M(p)$, for any atom p:

$$
\begin{array}{|c||c|}\hline \bot & \\ \hline & f \\ \hline \end{array}
\quad
\begin{array}{|c||c|}\hline \top & \\ \hline & t \\ \hline \end{array}
\quad
\begin{array}{|c||c|}\hline \neg & \\ \hline t & f \\ \hline f & t \\ \hline o & o \\ \hline \end{array}
\quad
\begin{array}{|c||c|c|c|}\hline \wedge & t & f & o \\ \hline t & t & f & o \\ \hline f & f & f & f \\ \hline o & o & f & o \\ \hline \end{array}
\quad
\begin{array}{|c||c|c|c|}\hline \vee & t & f & o \\ \hline t & t & t & t \\ \hline f & t & f & o \\ \hline o & t & o & o \\ \hline \end{array}
\quad
\begin{array}{|c||c|c|c|}\hline \rightarrow & t & f & o \\ \hline t & t & f & o \\ \hline f & t & t & t \\ \hline o & t & f & o \\ \hline \end{array}
\tag{18}
$$

We sometimes leave an interpretation M implicit and simply write $\phi : x$ instead of $v_M(\phi) = x$, for $x \in \{t, f, o\}$. Also, with a slight abuse of notation, an interpretation may be specified as a finite set of expressions of form $p : x$, where p is an atom and x is as before, containing only the relevant elements and omitting the implicit part.

A *three-valued model* of a formula ϕ is an interpretation that assigns either t or o to ϕ. Modelhood extends to sets of formulas in the standard way. Accordingly, given a set T of formulas and a formula ϕ, we define $T \models \phi$ if each model of T is a model of ϕ. Whenever necessary, we write $\models_3$ and $\models_2$ to distinguish three-valued from two-valued entailment.

Note that the truth value of $\phi \to \psi$ differs from that of $\neg\phi \vee \psi$ only in the case of a three valued interpretation M with $v_M(\phi) = o$ and $v_M(\psi) = f$, resulting in $v_M(\phi \to \psi) = f$ and $v_M(\neg\phi \vee \psi) = o$. This difference is prompted by the fact that t and o indicate modelhood, which motivates the assignment of the same truth values to $\phi \to \psi$ no matter whether we have $\phi : t$ or $\phi : o$. This has actually to do with the difference between *modus ponens* and *disjunctive syllogism*: The latter yields ψ from $\phi \wedge \neg\phi \wedge \neg\psi$ because $\phi \vee \psi$ follows from ϕ. The overall inference seems wrong because, in the presence of $\phi \wedge \neg\phi$, $\phi \vee \psi$ is satisfied (by $\phi : o$) with no need for ψ to be t. As pointed out in [40], one may actually view the connective $\to$ as

> "the 'right' generalisation of classical implication because $\to$ is the internal implication connective [5] for the defined inference relation in the sense that a deduction (meta)theorem holds for it: $T \cup \{\phi\} \models_3 \psi$ iff $T \models_3 \phi \to \psi$."

On the other hand, a formula composed of the connectives $\neg, \vee$, and $\wedge$ can never be inconsistent; that is, each such formula has at least one three-valued model [18]. Finally, we mention that the entailment problem for $\models_3$ is co-NP-complete, no matter whether $\to$ is included or not [50, 18, 22].

As mentioned previously, Priest's logic LP_m [56] was conceived to overcome the failure of disjunctive syllogism in LP [55]. LP amounts to the three-valued logic obtained by restricting the language to formulas in which only the connectives $\neg, \vee$, and $\wedge$ are permitted (and defining $\phi \to \psi$ as $\neg\phi \vee \psi$). In LP_m, modelhood is then limited to models containing a minimal number of *propositional variables* being assigned o. This allows for drawing

> "all classical inferences except where inconsistency makes them doubtful anyway" [56].

Formally, the consequence relation of LP_m can be defined as follows.

Definition 9. *For three-valued interpretations M and N, define the partial ordering $M \leq_m N$ iff, for each atom p, $v_M(p) = o$ implies $v_N(p) = o$. Then, $T \models_m \varphi$ iff every three-valued model of T that is minimal with respect to $\leq_m$ is a three-valued model of φ.*

Unlike this, the approach of Besnard and Schaub [10] prefers three-valued models that assign the truth value t to as many items of the knowledge base T as possible:

Definition 10. *For three-valued interpretations M and N, define the partial ordering $M \leq_n N$ iff $\{\phi \in T \mid v_M(\phi) = o\} \subseteq \{\phi \in T \mid v_N(\phi) = o\}$. Then, $T \models_n \varphi$ iff every three-valued model of T that is minimal with respect to $\leq_n$ is a three-valued model of φ.*

The major difference between the two approaches defined above is that the restriction of modelhood in LP_m focuses on models as close as possible to two-valued *interpretations*, whilst the approach of Definition 10 aims at models next to two-valued *models* of the considered premises. According to [10], the effects of making the formula select its preferred models can be seen by looking at $T = \{p, \neg p, (\neg p \vee q)\}$: While LP_m yields two $\leq_m$-preferred models, $\{p : o, q : t\}$ and $\{p : o, q : f\}$, from which one obtains $p \wedge \neg p$, the second approach yields q as additional conclusion. In fact, $\{p : o, q : t\}$ is the only $\leq_n$-preferred model of the premises $\{p, \neg p, (\neg p \vee q)\}$; it assigns t to $(\neg p \vee q)$, while this premise is attributed o by the second $\leq_m$-preferred model $\{p : o, q : f\}$. Hence, the latter is not $\leq_n$-preferred. So, while $T \not\models_m q$ and $T \models_n q$, we note that $T \cup \{(p \vee \neg q)\} \not\models_l q$ for $l = m, n$. On the other hand, $\models_n$ is clearly more syntax-dependent than $\models_m$ since the items within the knowledge base are used for distinguishing $\leq_n$-preferred models.

In fact, both inference relations $\models_m$ and $\models_n$ amount to their classical (two-valued) counterpart, whenever the set of premises is classically consistent. Also, it is shown in [22] that deciding entailment for $\models_m$ and $\models_n$ is Π_2^P-complete, no matter whether $\rightarrow$ is included or not. A logical analysis of both relations can be found in [40] and in the original literature [56, 10].

Encodings. We start with an encoding of the underlying three-valued logic introduced above by means of classical propositional logic.

To this end, we introduce, for each atom p, a globally new atom p' and define $P' = \{p' \mid p \in P\}$ for a given set P of atoms.

Let M be a three-valued interpretation over a set P of atoms. We define the *associated two-valued interpretation*, a_2^M, over $P \cup P'$ by setting

$$\begin{array}{rl} \mathsf{a}_2^M(p) = \mathsf{a}_2^M(p') = t & \text{if } M(p) = t, \\ \mathsf{a}_2^M(p) = \mathsf{a}_2^M(p') = f & \text{if } M(p) = f, \text{ and} \\ \mathsf{a}_2^M(p) = f \text{ and } \mathsf{a}_2^M(p') = t & \text{if } M(p) = o, \end{array}$$

for any atom $p \in P$. Conversely, for a given two-valued interpretation $I \subseteq P \cup P'$ satisfying $v_I(p \rightarrow p') = t$, for any $p \in P$, we define the *associated three-valued interpretation*, a_3^I, by setting

$$\mathsf{a}_3^I(p) = \begin{cases} I(p) & \text{if } I(p) = I(p'), \\ o & \text{if } I(p) = f \text{ and } I(p') = t, \end{cases}$$

for any $p \in P$.

Moreover, we need the following parameterised translation:

Definition 11. *For any atom p and any propositional formula ϕ and ψ, we define*

1. (a) $\tau[p; t] = p$,
   (b) $\tau[p; f] = \neg p'$,
   (c) $\tau[p; o] = \neg p \wedge p'$,
2. (a) $\tau[\neg\phi; t] = \tau[\phi; f]$,

   (*b*) $\tau[\neg\phi; f] = \tau[\phi; t]$,
   (*c*) $\tau[\neg\phi; o] = \tau[\phi; o]$,
3. (*a*) $\tau[\phi \wedge \psi; t] = \tau[\phi; t] \wedge \tau[\psi; t]$,
   (*b*) $\tau[\phi \wedge \psi; f] = \tau[\phi; f] \vee \tau[\psi; f]$,
   (*c*) $\tau[\phi \wedge \psi; o] = \neg\tau[\phi \wedge \psi; f] \wedge \neg\tau[\phi \wedge \psi; t]$,
4. (*a*) $\tau[\phi \vee \psi; t] = \tau[\phi; t] \vee \tau[\psi; t]$,
   (*b*) $\tau[\phi \vee \psi; f] = \tau[\phi; f] \wedge \tau[\psi; f]$,
   (*c*) $\tau[\phi \vee \psi; o] = \neg\tau[\phi \vee \psi; t] \wedge \neg\tau[\phi \vee \psi; f]$,
5. (*a*) $\tau[\phi \to \psi; t] = \tau[\phi; f] \vee \tau[\psi; t]$,
   (*b*) $\tau[\phi \to \psi; f] = \neg\tau[\phi; f] \wedge \tau[\psi; f]$,
   (*c*) $\tau[\phi \to \psi; o] = \neg\tau[\phi; f] \wedge \tau[\psi; o]$.

For computing the three-valued models of a set T of formulas, we use

$$\mathcal{N}[T] = \bigwedge_{\phi \in T} \neg\tau[\phi; f].$$

For example, consider $T = \{p, \neg p, (\neg p \vee q)\}$. We get:

$$\begin{aligned}
\mathcal{N}[T] &= \neg\tau[p; f] \wedge \neg\tau[\neg p; f] \wedge \neg\tau[(\neg p \vee q); f] \\
&= \neg\neg p' \wedge \neg\tau[p; t] \wedge \neg(\tau[\neg p; f] \wedge \tau[q; f]) \\
&= \neg\neg p' \wedge \neg p \wedge \neg(\tau[p; t] \wedge \neg q') \\
&= \neg\neg p' \wedge \neg p \wedge \neg(p \wedge \neg q').
\end{aligned}$$

Now, the latter formula is equivalent to $p' \wedge \neg p \wedge (\neg p \vee q')$, which is in turn equivalent to $p' \wedge \neg p$ by absorption. Hence, $\mathcal{N}[T]$ possesses four two-valued models (over $\{p, p', q, q'\}$), viz.

$$I_1 = \{p'\}, \quad I_2 = \{p', q\}, \quad I_3 = \{p', q'\}, \quad \text{and} \quad I_4 = \{p', q, q'\}.$$

In order to establish a correspondence among the four two-models of $\mathcal{N}[T]$ and the three three-valued models of T, assigning o to p and varying on q, the relation between the underlying sets of atoms $P = \{p, q\}$ and $P' = \{p', q'\}$ must be fixed. In fact, this is accomplished by adding $r \to r'$ for every $r \in P$. Observe that in the above example, I_2 does not have a corresponding three-valued interpretation.

In this way, we obtain the following result.

Theorem 9. *Let φ be a formula with $P = var(\varphi)$, let $P' = \{p' \mid p \in P\}$, and let $x \in \{t, f, o\}$.*

Then, the following conditions hold:

1. *For any three-valued interpretation M over P, if $v_M(\varphi) = x$, then $\bigwedge_{p \in P}(p \to p') \wedge \tau[\varphi; x]$ is true under a_2^M, the associated two-valued interpretation of M.*
2. *For any two-valued interpretation I over $P \cup P'$, if $\bigwedge_{p \in P}(p \to p') \wedge \tau[\varphi; x]$ is true under I, then $v_{\mathsf{a}_3^I}(\varphi) = x$, where a_3^I is the associated three-valued interpretation of I.*

Since the formula $\tau[\varphi; t] \vee \tau[\varphi; f] \vee \tau[\varphi; o]$ is clearly a tautology of classical logic, we immediately get the following relation between the three-valued models of a theory and the two-valued models of the corresponding encoding:

Corollary 1. *Let T be a theory with $P = var(T)$, and let $P' = \{p' \mid p \in P\}$.*

Then, there is a one-to-one correspondence between the three-valued models of T and the two-valued models of the formula

$$\bigwedge_{p \in P} (p \rightarrow p') \wedge \mathcal{N}[T], \tag{19}$$

with $\mathcal{N}[T] = \bigwedge_{\phi \in T} \neg\tau[\phi; f]$.

In particular, the three-valued model of T corresponding to a two-valued model I of (19) is given by the associated three-valued interpretation a_3^I of I.

For illustration, consider $T = \{p, \neg p, (\neg p \vee q)\}$ along with

$$(p \rightarrow p') \wedge (q \rightarrow q') \wedge \mathcal{N}[T],$$

which is equivalent to

$$(p \rightarrow p') \wedge (q \rightarrow q') \wedge (p' \wedge \neg p).$$

Unlike above, we obtain now as two-valued models I_1, I_3, and I_4 being in a one-to-one correspondence with the three three-valued models, $\{p : o, q : t\}$, $\{p : o, q : o\}$, and $\{p : o, q : f\}$, of T, respectively.

Before dealing with the reductions for the inference relations $\models_m$ and $\models_n$, it is instructive to see that the results developed so far already allow for a straightforward encoding of three-valued entailment, and, in particular, inference in logic LP [55]:

Theorem 10. *Let T be a theory with $var(T) = P$, and let φ be a formula.*

Then, $T \models_3 \varphi$ iff $\mathcal{D}eriv[\bigwedge_{p \in P}(p \rightarrow p') \wedge \mathcal{N}[T]; \neg\tau[\varphi; f]]$ is valid.

To be precise, we obtain (original) inference in LP [55] when restricting T and φ to formulas whose connectives are among $\neg$, $\wedge$, and $\vee$ only.

Let us now turn to Priest's logic LP_m [56]. For this, we must, roughly speaking, enhance the encoding of LP in order to account for the principle of "minimising inconsistency" used in LP_m. This is accomplished by means of the QBF module expressing propositional circumscription, as defined in Section 2.5.

Theorem 11. *Let T be a theory with $P = var(T)$, and let φ be a propositional formula. Furthermore, let $G = \{g_p \mid p \in var(T)\}$ be a set of new variables, and let $Q = P \cup P' \cup G \cup var(\varphi)$.*

Then, $T \models_m \varphi$ iff

$$\forall Q\Big(\mathcal{C}irc[(\textstyle\bigwedge_{p \in P}\big((p \rightarrow p') \wedge (g_p \equiv \tau[p; o])\big) \wedge \mathcal{N}[T]; G; P \cup P'] \rightarrow \neg\tau[\varphi; f]\Big)$$

is valid.

To be precise, we obtain (original) inference in LP_m [56] when restricting T and φ to formulas whose connectives are among $\neg$, $\wedge$, and $\vee$ only.

We obtain an axiomatisation of Besnard and Schaub's approach [10] in a completely analogous fashion:

Theorem 12. *Let T, P, and φ be as in Theorem 11, let $G = \{g_\phi \mid \phi \in T\}$ be a set of new guessing variables, and let $Q = P \cup P' \cup G \cup var(\varphi)$.*

Then, $T \models_n \varphi$ iff

$$\forall Q\Big(\mathcal{C}irc[(\textstyle\bigwedge_{p\in P}(p \to p') \wedge \bigwedge_{\phi\in T}(g_\phi \equiv \tau[\phi; o]) \wedge \mathcal{N}[T]; G; P \cup P'] \to \neg\tau[\varphi; f]\Big)$$

is valid.

It is a straightforward matter to check that the encodings given in the above theorems are adequate with respect to checking the corresponding inference relations. We also mention that alternative translations of the considered three-valued paraconsistent logics into QBFs are given in [13], based on different QBF modules for expressing the minimisation principles employed in the relations $\models_m$ and $\models_n$, respectively. Furthermore, although we do not detail it here, we stress that other multi-valued paraconsistent logics can analogously be treated in terms of reductions to QBFs. As a case in point, similar to the characterisations given in Theorems 11 and 12, [3] describes in effect axiomatisations of various four-valued paraconsistent logics into two-valued quantified propositional logic based on specific forms of propositional circumscription.

4 Conclusion

In this chapter, we discussed how differing approaches to paraconsistent reasoning can be expressed in a uniform framework by means of quantified propositional logic. We have started by introducing basic formulas that are used as building blocks for modeling advanced reasoning tasks. To a turn, we have demonstrated, by means of three case-studies, how specific paraconsistent inference problems can be mapped onto decision problems of QBFs.

The overall approach has several benefits. To begin with, it allows us to compare distinct approaches by looking at their axiomatisation as QBFs. Moreover, this axiomatisation provides an executable specification that can be given to existing QBF-solvers. In view of the considerable sophistication offered nowadays by these solvers, we obtain prototypical implementations with a relatively efficient performance.

The idea of encoding paraconsistent formalisms by means of QBFs is also investigated in [2]; interestingly, this approach uses signed formulas, as described in Section 3.2, for expressing inferences while preferences are expressed by QBFs. The idea of signed systems has recently been applied to database repair [4]. In this context, it is an interesting question in how far approaches to database repair and consistent query answering using annotated logics [1] (as a form of multi-valued logics) can be encoded by means of QBFs.

References

1. M. Arenas, L. Bertossi, and J. Chomicki. Consistent query answers in inconsistent databases. In *Proceedings of the Eighteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems* (*PODS '99*), pages 68–79. ACM Press, 1999.
2. O. Arieli. Paraconsistent preferential reasoning by signed quantified Boolean formulae. In *Proceedings of the 16th European Conference on Artificial Intelligence* (*ECAI 2004*), 2004. To appear.
3. O. Arieli and M. Denecker. Reducing preferential paraconsistent reasoning to classical entailment. *Journal of Logic and Computation*, 13(4):557–580, 2003.
4. O. Arieli, M. Denecker, B. V. Nuffelen, and M. Bruynooghe. Database repair by signed formulae. In *Proceedings of the Third Conference on Foundations of Information and Knowledge Systems* (*FoIKS '04*), volume 2942 of *Lecture Notes in Computer Science*, pages 14–30. Springer-Verlag, 2004.
5. A. Avron. Simple consequence relations. *Information and Computation*, 92:105–139, 1991.
6. A. Ayari and D. Basin. QUBOS: Deciding quantified Boolean logic using propositional satisfiability solvers. In M. Aagaard and J. O'Leary, editors, *Proceedings of the Fourth International Conference on Formal Methods in Computer-Aided Design* (*FMCAD 2002*), volume 2517 of *Lecture Notes in Computer Science*, pages 187–201. Springer-Verlag, 2002.
7. S. Benferhat, D. Dubois, and H. Prade. Argumentative inference in uncertain and inconsistent knowledge bases. In *Proceedings of the Ninth Conference on Uncertainty in Artificial Intelligence* (*UAI '93*), pages 411–419, 1993.
8. S. Benferhat, D. Dubois, and H. Prade. Some syntactic approaches to the handling of inconsistent knowledge bases: A comparative study. Part 1: The flat case. *Studia Logica*, 58(1):17–45, 1997.
9. P. Besnard. *An Introduction to Default Logic*. Springer-Verlag, 1989.
10. P. Besnard and T. Schaub. Circumscribing inconsistency. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence* (*IJCAI '97*), pages 150–155. Morgan Kaufmann Publishers, 1997.
11. P. Besnard and T. Schaub. Signed systems for paraconsistent reasoning. *Journal of Automated Reasoning*, 20:191–213, 1998.
12. P. Besnard, T. Schaub, H. Tompits, and S. Woltran. Paraconsistent reasoning via quantified Boolean formulas, I: Axiomatising signed systems. In S. Flesca, S. Greco, N. Leone, and G. Ianni, editors, *Proceedings of the Eighth European Conference on Logics in Artificial Intelligence* (*JELIA '02*), volume 2424 of *Lecture Notes in Computer Science*, pages 320–331. Springer-Verlag, 2002.
13. P. Besnard, T. Schaub, H. Tompits, and S. Woltran. Paraconsistent reasoning via quantified Boolean formulas, II: Circumscribing inconsistent theories. In *Proceedings of the Seventh European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty* (*ECSQARU '03*), volume 2711 of *Lecture Notes in Computer Science*, pages 528–539. Springer-Verlag, 2003.
14. G. Brewka. Preferred subtheories: An extended logical framework for default reasoning. In N. S. Sridharan, editor, *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence* (*IJCAI '89*), pages 1043–1048. Morgan Kaufmann Publishers, 1989.
15. R. E. Bryant. Graph-based algorithms for Boolean function manipulation. *IEEE Transactions on Computers*, C-35(8):677–691, 1986.

16. T. Bylander. The computational complexity of propositional STRIPS planning. *Artificial Intelligence*, 69(1–2):165–204, 1994.
17. M. Cadoli, A. Giovanardi, and M. Schaerf. An algorithm to evaluate quantified Boolean formulae. In *Proceedings of the 15th National Conference on Artificial Intelligence* (*AAAI '98*), pages 262–267. AAAI Press/MIT Press, 1998.
18. M. Cadoli and M. Schaerf. On the complexity of entailment in propositional multivalued logics. *Annals of Mathematics and Artificial Intelligence*, 18:29–50, 1996.
19. C. Cayrol, M. Lagasquie-Schiex, and T. Schiex. Nonmonotonic reasoning: From complexity to algorithms. *Annals of Mathematics and Artificial Intelligence*, 22(3–4):207–236, 1998.
20. C. Cayrol and M.-C. Lagasquie-Schiex. Non-monotonic syntax-based entailment: A classification of consequence relations. In C. Froidevaux and J. Kohlas, editors, *Proceedings of the Third European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty* (*ECSQARU '95*), volume 946 of *Lecture Notes in Computer Science*, pages 107–114. Springer-Verlag, 1995.
21. A. Church. *Introduction to Mathematical Logic, Volume I.* Princeton University Press, 1956.
22. S. Coste-Marquis and P. Marquis. Complexity results for paraconsistent inference relations. In D. Fensel, F. Giunchiglia, D. McGuiness, and M. Williams, editors, *Proceedings of the Eighth International Conference on Principles of Knowledge Representation and Reasoning* (*KR '02*), pages 61–72. Morgan Kaufmann Publishers, 2002.
23. M. Davis, G. Logemann, and D. Loveland. A machine program for theorem proving. *Communications of the ACM*, 5(7):394–397, 1962.
24. M. Davis and H. Putman. A computing procedure for quantification theory. *Journal of the ACM*, 7(3):201–215, 1960.
25. J. Delgrande, T. Schaub, H. Tompits, and S. Woltran. On computing solutions to belief change scenarios. In S. Benferhat and P. Besnard, editors, *Proceedings of the Sixth European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty* (*ECSQARU '01*), volume 2143 of *Lecture Notes in Computer Science*, pages 510–521. Springer-Verlag, 2001.
26. I. D'Ottaviano and N. da Costa. Sur un problème de Jaśkowski. In *Comptes Rendus de l'Académie des Sciences de Paris*, volume 270, pages 1349–1353, 1970.
27. U. Egly, T. Eiter, H. Tompits, and S. Woltran. Solving advanced reasoning tasks using quantified Boolean formulas. In *Proceedings of the 17th National Conference on Artificial Intelligence* (*AAAI 2000*), pages 417–422. AAAI Press/MIT Press, 2000.
28. U. Egly, R. Pichler, and S. Woltran. On deciding subsumption problems. In *Proceedings of the Fifth International Symposium on the Theory and Applications of Satisfiability Testing* (*SAT 2002*), pages 89–97, 2002.
29. U. Egly, H. Tompits, and S. Woltran. On quantifier shifting for quantified Boolean formulas. In *Proceedings of the SAT 2002 Workshop on Theory and Applications of Quantified Boolean Formulas* (*QBF 2002*), pages 48–61, 2002.
30. T. Eiter and G. Gottlob. On the computational cost of disjunctive logic programming: Propositional case. *Annals of Mathematics and Artificial Intelligence*, 15(3–4):289–323, 1995.
31. T. Eiter, V. Klotz, H. Tompits, and S. Woltran. Modal nonmonotonic logics revisited: Efficient encodings for the basic reasoning tasks. In U. Egly and C. Fermüller, editors, *Proceedings of the Eleventh International Conference on Automated Reasoning with Analytic Tableaux and Related Methods* (*TABLEAUX 2002*), volume 2381 of *Lecture Notes in Computer Science*, pages 100–114. Springer-Verlag, 2002.

32. R. Feldmann, B. Monien, and S. Schamberger. A distributed algorithm to evaluate quantified Boolean formulas. In *Proceedings of the 17th National Conference on Artificial Intelligence* (*AAAI 2000*), pages 285–290. AAAI Press/MIT Press, 2000.
33. A. Frisch. Inference without chaining. In J. McDermott, editor, *Proceedings of the Tenth International Joint Conference on Artificial Intelligence* (*IJCAI '87*), pages 515–519. Morgan Kaufmann Publishers, 1987.
34. M. R. Garey and D. S. Johnson. *Computers and Intractability.* W. H. Freeman, 1979.
35. E. Giunchiglia, M. Narizzano, and A. Tacchella. QuBE: A system for deciding quantified Boolean formulas satisfiability. In R. Goré, A. Leitsch, and T. Nipkow, editors, *Proceedings of the First International Joint Conference on Automated Reasoning* (*IJCAR 2001*), volume 2083 of *Lecture Notes in Computer Science*, pages 364–369. Springer-Verlag, 2001.
36. G. Gottlob. Complexity results for nonmonotonic logics. *Journal of Logic and Computation*, 2(3):397–425, 1992.
37. H. Kautz, D. McAllester, and B. Selman. Encoding plans in propositional logic. In L. Aiello, J. Doyle, and S. Shapiro, editors, *Proceedings of the Fifth International Conference on Principles of Knowledge Representation and Reasoning* (*KR '96*), pages 374–384. Morgan Kaufmann Publishers, 1996.
38. H. Kautz and B. Selman. Planning as satisfiability. In B. Neumann, editor, *Proceedings of the Tenth European Conference on Artificial Intelligence* (*ECAI '92*), pages 359–363. John Wiley & Sons, 1992.
39. H. Kleine Büning, M. Karpinski, and A. Flögel. Resolution for quantified Boolean formulas. *Information and Computation*, 117(1):12–18, 1995.
40. S. Konieczny and P. Marquis. Three-valued logics for inconsistency handling. In S. Flesca, S. Greco, N. Leone, and G.Ianni, editors, *Proceedings of the Eighth European Conference on Logics in Artificial Intelligence* (*JELIA '02*), volume 2424 of *Lecture Notes in Computer Science*, pages 332–344. Springer-Verlag, 2002.
41. R. E. Ladner. The computational complexity of provability in systems of modal propositional logic. *SIAM Journal on Computing*, 6(3):467–480, 1977.
42. S. Leśniewski. Grundzüge eines neuen System der Grundlagen der Mathematik. *Fundamenta Mathematica*, 14:1–81, 1929.
43. R. Letz. Lemma and model caching in decision procedures for quantified Boolean formulas. In U. Egly and C. Fermüller, editors, *Proceedings of the Eleventh International Conference on Automated Reasoning with Analytic Tableaux and Related Methods* (*TABLEAUX 2002*), volume 2381 of *Lecture Notes in Computer Science*, pages 160–175. Springer-Verlag, 2002.
44. H. Levesque. A knowledge-level account of abduction. In N. S. Sridharan, editor, *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence* (*IJCAI '89*), pages 1061–1067. Morgan Kaufmann Publishers, 1989.
45. J. Łukasiewicz and A. Tarski. Untersuchungen über den Aussagenkalkül. *Comptes Rendus Séances Société des Sciences et Lettres Varsovie*, 23(Cl. III):30–50, 1930.
46. J. McCarthy. Circumscription - A form of nonmonotonic reasoning. *Artificial Intelligence*, 13:27–39, 1980.
47. A. R. Meyer and L. J. Stockmeyer. The equivalence problem for regular expressions with squaring requires exponential space. In *13th Annual Symposium on Switching and Automata Theory*, pages 125–129, 1972.
48. A. R. Meyer and L. J. Stockmeyer. Word problems requiring exponential time. In *ACM Symposium on Theory of Computing* (*STOC '73*), pages 1–9. ACM Press, 1973.

49. S. Minato. *Binary Decision Diagrams and Applications for VLSI CAD*. Kluwer, 1996.
50. D. Mundici. Satisfiability in many-valued sentential logic is NP-complete. *Theoretical Computer Science*, 52(1-2):145–153, 1987.
51. B. Nebel. Belief revision and default reasoning: Syntax-based approaches. In J. Allen, R. Fikes, and E. Sandewall, editors, *Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning* (*KR '91*), pages 417–428. Morgan Kaufmann Publishers, 1991.
52. C. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.
53. D. Pearce, H. Tompits, and S. Woltran. Encodings for equilibrium logic and logic programs with nested expressions. In P. Brazdil and A. Jorge, editors, *Proceedings of the Tenth Portuguese Conference on Artificial Intelligence* (*EPIA '01*), volume 2258 of *Lecture Notes in Computer Science*, pages 306–320. Springer-Verlag, 2001.
54. D. Plaisted, A. Biere, and Y. Zhu. A satisfiability procedure for quantified Boolean formulae. *Discrete Applied Mathematics*, 130:291–328, 2003.
55. G. Priest. Logic of paradox. *Journal of Philosophical Logic*, 8:219–241, 1979.
56. G. Priest. Reasoning about truth. *Artificial Intelligence*, 39:231–244, 1989.
57. R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 13(1–2):81–132, 1980.
58. N. Rescher. *Plausible Reasoning*. Van Gorcum, Amsterdam, 1976.
59. N. Rescher and R. Manor. On inference from inconsistent premises. *Theory and Decision*, 1:179–219, 1970.
60. J. Rintanen. Constructing conditional plans by a theorem prover. *Journal of Artificial Intelligence Research*, 10:323–352, 1999.
61. J. Rintanen. Improvements to the evaluation of quantified Boolean formulae. In T. Dean, editor, *Proceedings of the 16th International Joint Conference on Artificial Intelligence* (*IJCAI '99*), pages 1192–1197. Morgan Kaufmann Publishers, 1999.
62. J. A. Robinson. A machine-oriented logic based on the resolution principle. *Journal of the ACM*, 12(1):23–41, 1965.
63. B. Russell. The theory of implication. *American Journal of Mathematics*, 28(2):159–202, 1906.
64. S. Schamberger. Ein paralleler Algorithmus zum Lösen von Quantifizierten Boole'schen Formeln. Master's thesis, Universität Gesamthochschule Paderborn, 2000.
65. J. Srzednicki and Z. Stachniak, editors. *Lesniewski's Systems Protothetic*. Dordrecht, 1998.
66. R. Statman. Intuitionistic propositional logic is polynomial-space complete. *Theoretical Computer Science*, 9:67–72, 1979.
67. L. J. Stockmeyer. The polynomial-time hierarchy. *Theoretical Computer Science*, 3(1):1–22, 1976.
68. H. Tompits. Expressing default abduction problems as quantified Boolean formulas. *AI Communications*, 16:89–105, 2003.
69. H. Turner. Polynomial-length planning spans the polynomial hierarchy. In S. Flesca, S. Greco, N. Leone, and G. Ianni, editors, *Proceedings of the Eighth European Conference on Logics in Artificial Intelligence* (*JELIA '02*), volume 2424 of *Lecture Notes in Computer Science*, pages 111–124. Springer-Verlag, 2002.
70. A. N. Whitehead and B. Russell. *Principia Mathematica*, volume 1–3. Cambridge University Press, 1910–13.
71. S. Woltran. *Quantified Boolean Formulas – From Theory to Practice*. PhD thesis, Technische Universität Wien, Institut für Informationssysteme, 2003.

72. C. Wrathall. Complete sets and the polynomial-time hierarchy. *Theoretical Computer Science*, 3(1):23–33, 1976.
73. L. Zhang and S. Malik. Towards a symmetric treatment of satisfaction and conflicts in quantified Boolean formula evaluation. In P. V. Hentenryck, editor, *Proceedings of the Eighth International Conference on Principles and Practice of Constraint Programming (CP 2002)*, volume 2470 of *Lecture Notes in Computer Science*, pages 200–215. Springer-Verlag, 2002.