Knowledge compilation for closed word reasoning and circumscription *

Sylvie Coste-Marquis[†⋆] and Pierre Marquis[⋆]
[⋆]CRIL [†]IUT de Lens / Université d'Artois
rue de l'Université – S.P. 16
F–62307 Lens Cedex – France
Email: {coste, marquis}@cril.univ-artois.fr

Abstract

This paper presents new complexity results for propositional closed world reasoning (CWR) from tractable knowledge bases (KBs). Both (basic) CWR, generalized CWR, extended generalized CWR, careful CWR and extended CWR (equivalent to circumscription) are considered. The focus is laid on tractable KBs belonging to target classes for exact compilation functions: Blake formulas, DNFs, disjunctions of Horn formulas, and disjunctions of renamable Horn formulas. The complexity of inference is identified for all the forms of CWR listed above. For each of them, new tractable fragments are exhibited. Interestingly, the restricted classes of formulas we consider are target classes for exact compilation functions, i.e., every KB can be turned into an equivalent formula from any of these classes. Accordingly, our results suggest knowledge compilation as a valuable, practical approach to deal with the complexity of CWR in some situations.

1 Introduction

Closed world reasoning (CWR) is a widely used inference technique in Artificial Intelligence, Database Theory and Logic Programming. It relies on the idea that negative information is often not represented in an explicit way; in this situation, every piece of positive information that cannot be deduced from a knowledge base (KB) is assumed false.

In order to define CWR in a formal way, an approach consists in characterizing the formulas that must be assumed false in the KB: closed world inference is then viewed as deduction from the closure of the KB, i.e., the KB completed with these assumptions. Several policies for characterizing such assumptions have been developed

*This paper is an extended and revised version of the paper entitled "Complexity results for propositional closed word reasoning and circumscription from tractable knowledge bases" published in the proceedings of the 16^{th} International Joint Conference on Artificial Intelligence (IJCAI'99), pages 24–29.

so far, giving rise to several forms of CWR. Let us mention the basic closed world assumption (CWA) by Reiter [45], the generalized closed world assumption (GCWA) by Minker [39], the extended generalized closed world assumption (EGCWA) by Yahya and Henschen [51], the careful closed world assumption (CCWA) by Gelfond and Przymusinska [22], and the extended closed world assumption (ECWA) by Gelfond, Przymusinska and Przymusinski [23]. The most sophisticated closed world assumption is ECWA; in the propositional case, Gelfond, Przymusinska and Przymusinski [23] shows it is equivalent to circumscription, as defined in [38][1].

The complexity of propositional closed world inference has been investigated by several researchers. Eiter and Gottlob [19] have studied in detail the complexity of inference in the general case (i.e., both the KB and the queries are unrestricted propositional formulas) and shown it hard: closed world inference typically is at the second level of the polynomial hierarchy. Cadoli and Lenzerini [11] have focused on the complexity of closed world inference for various fragments of propositional logic where clause deduction is tractable, especially the Horn CNF class, the reverse Horn CNF class and the Krom class. For such classes of formulas, the complexity of closed world inference falls one level down in the polynomial hierarchy.

The aim of this paper is to complete the complexity results pointed out in [11], by focusing on some other fragments of propositional logic where clause deduction is tractable. Especially, four classes are considered:

- The *Blake* class is the set of formulas given in prime implicates normal form,
- The *DNF* class is the set of formulas given in disjunctive normal form (DNF),
- The *Horn cover* class is the set of disjunctions of Horn CNF formulas,
- The *renamable Horn cover* class is the set of disjunctions of renamable Horn CNF formulas.

As a main contribution, the complexity of formula/clause/literal inference is identified for each form of CWR and each tractable class listed above. Several new tractable subcases are exhibited for each form of CWR; some new intractable subcases are presented as well.

From these complexity results, new conclusions can be drawn about the applicability of closed world inference, and the usefulness of knowledge compilation for closed world reasoning. Thus, our tractability results apply to classes of formulas strictly more expressive than some of the most expressive tractable classes for CWR pointed out up to now. In particular, some of the four tractable fragments we focus on include classes considered in [11] as strict subcases. Additionally, the fragments we consider are target classes for exact knowledge compilation functions. Thus, every propositional formula can be turned ("compiled") into a formula from the Blake class (the target class for Reiter and de Kleer's compilation function [46]), the DNF class (the target class for Schrag's function [47]), the Horn cover class and the renamable Horn cover class (that are specific instances of the classes of tractable covers considered by Boufkhad *et al.* [4]). Showing closed world inference from such tractable formulas computationally easier than from unconstrained formulas, our study suggests a two

[1] In a first definition [37], circumscription was given a more restricted definition where varying atoms were not allowed.

step approach to compute closed world inference: the KB is first compiled off-line, giving rise to a formula from one of the target classes, then queries are addressed on-line to the compiled KB. While such a pre-processing is known as not computationally helpful in the general case, our results indicate that it can prove valuable in the case where closed world inference from the compiled KB can be achieved in polynomial time, as long as the KB does not often change and the size of its compiled form remains "small enough".

The rest of this paper is organized as follows. Some formal preliminaries are given in Section 2. Basic definitions and the complexity of CWR in the general case are recalled in Section 3. Section 4 presents the complexity of CWR for each of the fragments we focus on. The complexity results are discussed and our study is compared with some related works. Section 5 concludes the paper. Proofs are sketched in an appendix.

2 Formal Preliminaries

We briefly present the basic notions about propositional logic and complexity theory that are considered in the following. More details about such notions can be found e.g., in [20] and [21, 42].

2.1 Propositional logic

Let PS be a denumerable set of propositional symbols. $PROP_{PS}$ denotes the propositional language built up from the symbols of PS and the connectives in the standard way. The elements of $PROP_{PS}$ are called formulas. $Var(\Sigma)$ denotes the set of propositional symbols occurring in Σ. The size of a formula Σ, noted $|\Sigma|$, is the number of signs (symbols and connectives) used to write it.

Every propositional symbol of PS is also called a positive literal and a negated one a negative literal. A literal is either a positive literal or a negative literal. For every subset V of PS, L_V (resp. L_V^+, L_V^-) is the set of literals (resp. positive literals, negative literals) built up from the propositional symbols of V. Every finite disjunction of literals is called a clause and every finite conjunction of literals is called a term.

Formulas are interpreted in the classical way. An interpretation (over $PROP_{PS}$) is a mapping I that associates every propositional symbol to one of the two truth values of $BOOL = \{true, false\}$. The semantics $[\![\Sigma]\!]$ of a formula Σ is a boolean function: a truth value can be associated to Σ as long as an interpretation I is considered. $[\![\Sigma]\!](I)$ denotes the truth value taken by Σ within I; it is defined in the usual compositional way. When I is s.t. $[\![\Sigma]\!](I) = 1$, I is said a model of Σ, noted $I \models \Sigma$; otherwise, I is a counter-model of Σ. When a formula has no model (resp. no counter-model), it is said unsatisfiable (resp. valid). The binary relation $\models$ over $PROP_{PS}$ is defined by $\Sigma \models \Phi$ iff every model of Σ is a model of Φ. $\models$ is referred to as logical entailment: whenever $\Sigma \models \Phi$ holds, Φ is said a logical consequence of Σ. Whenever $\Sigma \models \Phi$ and $\Phi \models \Sigma$ both hold, Σ and Φ are said logically equivalent, noted $\Sigma \equiv \Phi$.

Every set of formulas is interpreted conjunctively; especially, when it is finite, a set of formulas is also considered as the formula that is the conjunction of its elements.

It is well-known that every formula admits a conjunction of clauses (resp. a disjunction of terms) equivalent to it; such a conjunction of clauses (resp. disjunction of terms) is called a conjunctive normal form (CNF for short) (resp. a disjunctive normal form (DNF for short)) of the formula. A CNF formula is said positive when no negative literal occurs in it; especially, a positive clause contains only positive literals. A formula is Horn CNF (resp. reverse Horn CNF) iff it is a CNF formula s.t. every clause in it contains at most one positive (resp. negative) literal. A Krom formula is a CNF formula in which every clause contains at most two literals. A renamable Horn CNF formula is a CNF formula Σ which can be turned into a Horn CNF formula by substituting in a uniform way in Σ some literals of $L_{Var(\Sigma)}$ by their negation.

2.2 Computational complexity

A decision problem is considered efficiently solvable iff there exists a (deterministic) algorithm that can classify every instance of it in a number of computational steps polynomially bounded in the input size. The class of all decision problems solvable in polynomial time is denoted by P. The class of all languages (encoding decision problems) that can be recognized in polynomial time by a non-deterministic Turing machine is denoted by NP. Because a deterministic Turing machine can be considered as a non-deterministic one, the inclusion $\mathsf{P} \subseteq \mathsf{NP}$ is established; however, the converse is the famous open problem: $\mathsf{P} \stackrel{?}{=} \mathsf{NP}$ (conjectured false). Among all the problems in NP, the hardest ones are those from which every problem in NP can be polynomially many-one reduced: such problems are referred to as NP-complete. If any of them has a polynomial algorithm, then $\mathsf{P} = \mathsf{NP}$ holds. Accordingly, it is believed that it is impossible to solve NP-complete problems in polynomial time. SAT, the problem of determining whether a propositional CNF formula is satisfiable, is the prototypical NP-complete problem. Its complementary problem $UNSAT$ (that consists in determining whether a propositional CNF formula is unsatisfiable) is not necessarily in NP (in contrast to P, NP is not known to be closed under complementation). It is assigned to the class coNP that contains the complementary problems to problems of NP. It is conjectured that $\mathsf{NP} \neq \mathsf{coNP}$. Let X be a class of decision problems. P^X (resp. NP^X) is the class of all decision problems that can be solved in polynomial time using a deterministic (resp. non-deterministic) Turing machine that can use an oracle for deciding a problem $Q \in X$ for "free" (i.e., within a constant, unit time). The class $\mathsf{P}^{X[\mathcal{O}(log n)]}$ contains the decision problems that can be solved in polynomial time with no more than $\mathcal{O}(log\ n)$ calls to the oracle, where n represents the size of the problem instance. The classes Δ^p_k, Σ^p_k and Π^p_k can be defined by:

- $\Delta^p_0 = \Sigma^p_0 = \Pi^p_0 = \mathsf{P}$,
- $\Delta^p_{k+1} = \mathsf{P}^{\Sigma^p_k}$,
- $\Sigma^p_{k+1} = \mathsf{NP}^{\Sigma^p_k}$,
- $\Pi^p_{k+1} = \mathsf{co}\Sigma^p_{k+1}$.

Thus, $\Sigma^p_1 = \mathsf{NP}$ and $\Pi^p_1 = \mathsf{coNP}$. The *polynomial hierarchy* PH is the union of all Σ^p_k (for k integer). A decision problem is said at the k^{th} level of the polynomial hierarchy iff it belongs to Δ^p_{k+1}, and is either Σ^p_k-hard or Π^p_k-hard. While it is easy to check that both $\Delta_k^p \subseteq \Sigma_{k+1}^p$, $\Delta_k^p \subseteq \Pi_{k+1}^p$, $\Sigma_k^p \subseteq \Delta_{k+1}^p$, and $\Pi_k^p \subseteq \Delta_{k+1}^p$ hold for every k, it is unknown whether the inclusions are proper (but it is conjectured that it is the case). Thus, it is strongly believed that the polynomial hierarchy does not collapse (at any level), i.e., is a truly infinite hierarchy (for every integer k, $\mathsf{PH} \neq \Sigma_k^p$).

3 Closed world reasoning: basic definitions and complexity results

All the forms of closed world inference pointed out so far can be characterized through the notions of *closure* and *freeness for negation*. A formula Φ is then considered as a (non monotonic) consequence of a KB Σ interpreted under some closed world assumption policy *CWA (where the generic character $*$ can be replaced by G, EG, CC, E or the empty string) iff it is a logical consequence of the closure of Σ w.r.t. the policy. Both the CCWA and ECWA policies require $Var(\Sigma)$ to be partitioned into three sets, P, Q, and Z. P contains the symbols preferred false, Z contains the symbols the truth value of which can vary when trying to falsify the symbols from P, and Q contains the symbols the truth value of which cannot vary.

In order to define the notion of closure, we first need to specify what is a formula free for negation w.r.t. each *CWA policy. The formulas which are free for negation are the negations of the formulas assumed false when not deducible from Σ. We write that a formula is *CWA-ffn when it is free for negation w.r.t. the *CWA policy. More formally:

Definition 1 (*CWA-ffn formula)
Let Σ and α be two formulas from $PROP_{PS}$.

- α is CWA-ffn iff α is a positive literal s.t. $\Sigma \not\models \alpha$ holds.
- α is GCWA-ffn iff α is a positive literal and for each positive clause γ s.t. $\Sigma \not\models \gamma$ holds, $\Sigma \not\models \alpha \vee \gamma$ holds.
- α is EGCWA-ffn iff α is a conjunction of positive literals and for each positive clause γ s.t. $\Sigma \not\models \gamma$ holds, $\Sigma \not\models \alpha \vee \gamma$ holds.
- Let $\langle P, Q, Z\rangle$ be a partition of $Var(\Sigma)$. α is CCWA-ffn iff α is a literal from L_P^+ and for each clause γ containing only literals from $L_P^+ \cup L_Q$ and s.t. $\Sigma \not\models \gamma$ holds, $\Sigma \not\models \alpha \vee \gamma$ holds.
- Let $\langle P, Q, Z\rangle$ be a partition of $Var(\Sigma)$. α is ECWA-ffn iff $Var(\alpha) \cap Z = \emptyset$ and for each clause γ containing only literals from $L_P^+ \cup L_Q$ and s.t. $\Sigma \not\models \gamma$ holds, $\Sigma \not\models \alpha \vee \gamma$ holds.

From this definition, every symbol that does not belong to $Var(\Sigma)$ is CWA, GCWA and EGCWA free for negation. As to the other closed world assumption policies, every symbol from $PS \setminus Var(\Sigma)$ is assumed to belong to P (so its negation can also be inferred under the two policies). We are now ready to define the notion of closure of a KB.

Definition 2 (closure of a KB w.r.t. a CWA policy)
Let *CWA be any closed world assumption policy among (basic) CWA, GCWA, EGCWA, CCWA and ECWA. Let Σ be a formula from $PROP_{PS}$ and $\langle P, Q, Z\rangle$ a partition of $Var(\Sigma)$. The closure $*CWA(\Sigma, \langle P, Q, Z\rangle)$[2] of Σ given $\langle P, Q, Z\rangle$ w.r.t. *CWA is the formula $\Sigma \cup \{\neg\alpha$; α is a *CWA-ffn formula w.r.t. Σ and $\langle P, Q, Z\rangle\}$.

It is well-known that *CWA-ffn formulas can also be characterized in a model-theoretic way, through a notion of minimal model [34].

Definition 3 (($\langle P, Q, Z\rangle$)-minimal model)
Let Σ be a formula from $PROP_{PS}$. Let I and J be two models of Σ.

- We write $I \leq J$ iff every symbol interpreted as *true* in I also is interpreted as *true* in J. I is a *minimal model* of Σ iff there is no model J of Σ s.t. $J \leq I$ and $I \not\leq J$.
- Let $\langle P, Q, Z\rangle$ be a partition of $Var(\Sigma)$. We write $I \leq_{\langle P,Q,Z\rangle} J$ iff every symbol from Q is interpreted in the same way in I and J and every symbol from P interpreted as *true* in I also is interpreted as *true* in J. I is a $\langle P, Q, Z\rangle$-*minimal model* of Σ iff there is no model J of Σ s.t. $J \leq_{\langle P,Q,Z\rangle} I$ and $I \not\leq_{\langle P,Q,Z\rangle} J$.

Let us note that when $Q = Z = \emptyset$, the $\langle P, Q, Z\rangle$-minimal models of Σ are its minimal models.

The following semantic characterizations of *CWA-ffn formulas can be derived (see [11]):

- α is CWA-ffn iff α is a positive literal and there exists a model M of Σ s.t. $M \not\models \alpha$ holds.
- α is GCWA-ffn iff α is a positive literal and for each minimal model M of Σ, $M \not\models \alpha$ holds.
- α is EGCWA-ffn iff α is a conjunction of positive literals and for each minimal model M of Σ, $M \not\models \alpha$ holds.
- Let $\langle P, Q, Z\rangle$ be a partition of $Var(\Sigma)$. α is CCWA-ffn iff α is a literal from L_P^+ and for each $\langle P, Q, Z\rangle$-minimal model M of Σ, $M \not\models \alpha$ holds.
- Let $\langle P, Q, Z\rangle$ be a partition of $Var(\Sigma)$. α is ECWA-ffn iff $Var(\alpha) \cap Z = \emptyset$ an for each $\langle P, Q, Z\rangle$-minimal model M of Σ, $M \not\models \alpha$ holds.

It is well-known that the basic CWA policy makes sense only on restricted KBs. Indeed, unlike the other policies, the consistency of the closure of a KB w.r.t. basic CWA is not guaranteed.

In the rest of this paper, the following decision problems are considered:

[2]Since the partition of $Var(\Sigma)$ is not significant for the CWA, GCWA and EGCWA policies, the closure of a KB Σ w.r.t any of them will be written in a more simple way as $*CWA(\Sigma)$ (where * stands for G, EG or the empty string).

Definition 4 (*CWA formula inference)
Let *CWA be any closed world assumption policy among (basic) CWA, GCWA, EGCWA, CCWA and ECWA. *CWA FORMULA INFERENCE is the following decision problem:

- **Input:** Two formulas Σ and Φ from $PROP_{PS}$, a partition $\langle P, Q, Z\rangle$ of $Var(\Sigma)$ and a CWA policy *CWA.
- **Query:** Does $*CWA(\Sigma, \langle P, Q, Z\rangle) \models \Phi$ hold ?

*CWA CLAUSE INFERENCE (resp. *CWA LITERAL INFERENCE) is the restriction of the corresponding *CWA FORMULA INFERENCE problem where Φ is restricted to be a clause (resp. a literal).

As a baseline w.r.t. some complexity considerations, the following decision problem will also be considered:

Definition 5 (formula deduction)
FORMULA DEDUCTION is the following decision problem:

- **Input:** Two formulas Σ and Φ from $PROP_{PS}$,
- **Query:** Does $\Sigma \models \Phi$ hold?

CLAUSE DEDUCTION is the restriction of FORMULA DEDUCTION to the case where Φ is required to be a clause.

The complexity of propositional closed world inference has been studied by several researchers, especially Eiter and Gottlob in [19] and Cadoli and Lenzerini in [11]. The results reported in Table 1 have been established in the general case in [19]:

***CWA**	FORMULA INFERENCE CLAUSE INFERENCE LITERAL INFERENCE
CWA	NP-hard and in $\mathsf{P}^{\mathsf{NP}[\mathcal{O}(\log n)]}$
GCWA	Π_2^p-hard and in $\mathsf{P}^{\Sigma_2^p[\mathcal{O}(\log n)]}$
EGCWA	Π_2^p-complete
CCWA	Π_2^p-hard and in $\mathsf{P}^{\Sigma_2^p[\mathcal{O}(\log n)]}$
ECWA	Π_2^p-complete

Table 1: Complexity of closed world inference (general case).

Thus, all the forms of CWR (except the one based on basic CWA) are at the second level of the polynomial hierarchy. These results show closed world inference computationally harder than classical entailment in the general case (FORMULA DEDUCTION is "only" coNP-complete) unless the polynomial hierarchy collapses.

In order to circumvent this complexity, several approaches can be considered. One of them is centered around *exact knowledge compilation* (see [6] for a survey). Knowledge compilation can prove helpful in the objective of improving many forms of inference, in particular clause deduction [30] but also diagnosis, planning, belief revision, etc. [33, 40]. It can be viewed as a form of pre-processing: the original KB is turned into a compiled one during an off-line compilation phase and this compiled KB is used to answer queries on-line. Assuming that the KB does not often change and that answering queries from the compiled KB is computationally easier than answering them from the original KB, the compilation time can be balanced over a sufficient number of queries.

Existing research about knowledge compilation can be split into two categories. The first category gathers theoretical works about *compilability*, which indicates whether or not the objective can be expected to be reached in the worst case by focusing on the size of the compiled form (see [8, 9, 10, 7, 33]). Indeed, if the size of the compiled form is exponentially larger than the size of the original KB, significant computational improvements are hard to be expected. Formally, compilability concerns decision problems P with a fixed part F (for which pre-processing is allowed) and a varying part V. The notion of compilable problem is defined as follows [9][3]:

Definition 6 (compilable problem)
A problem $[P, F, V]$ is *compilable* if there exist two polynomials p_1 and p_2, and an algorithm $ASKS?$ s.t. for each instance f of F, there is a data structure D_f s.t.:

1. $|D_f| \leq p_1(|f|)$, and
2. for each instance v of V, $ASKS?(D_f, v) = \mathit{true}$ iff $\langle f, v \rangle$ is a "yes"-instance of P, and
3. $ASKS?(D_f, v)$ requires time $\leq p_2(|v| + |D_f|)$.

Some decision problems are compilable, while others are (probably) not compilable. For instance, CLAUSE DEDUCTION from a fixed KB is probably not compilable. Indeed, the existence of an equivalence-preserving compilation function $COMP$ s.t. it is guaranteed that for every propositional CNF formula Σ, CLAUSE DEDUCTION from $COMP(\Sigma)$ is in P and $|COMP(\Sigma)|$ is polynomially bounded in $|\Sigma|$ would make the polynomial hierarchy to collapse at the second level (see [30, 7] for more details).

The second category contains works that are much more oriented towards the design of compilation algorithms, and their empirical evaluations (see e.g., [46, 30, 15, 16, 17, 35, 47, 48, 4, 3]).

Now, what's about knowledge compilation and closed world inference? From the theoretical side, the compilability of closed world inference has been analyzed in depth in [8, 9, 10]. The results are typically negative. In a nutshell, inference under ECWA from a fixed KB is shown (probably) not compilable, even when the KB is Krom, the query is a literal and the partition $\langle P, Q, Z \rangle$ is allowed to vary, or when the KB is Krom, the query is a clause and the partition is fixed. Contrastingly, it is shown that CWA CLAUSE INFERENCE and CCWA CLAUSE INFERENCE from Horn CNF, reverse Horn CNF or Krom formulas are compilable (the fixed part of the problem is the KB plus the partition in the CCWA case).

From the practical side, Nerode, Ng and Subrahmanian [41] presents an algorithm for compiling a KB interpreted under circumscription into the set of its minimal

[3] A more general framework for compilability, with compilability classes and reductions, is given in [7]; Definition 6 is sufficient for our purpose.

models. The algorithm is based on a mixed integer linear programming. The circumscription policy (i.e., the partition of the symbols) is fixed. Several other approaches show how closed world inference can be reduced to classical entailment, through the computation of the closure of the KB (or the computation of any equivalent formula), see e.g., [44, 12]. All these approaches confirm the (probable) non compilability of the most sophisticated forms of CWR in the sense that, in the worst case, the compilation phase outputs a formula not polynomially bounded in the size of the original KB (see [10] for more details).

4 Complexity results for CWR from tractable KBs

In this section, the complexity of closed world inference from tractable KBs is investigated. First, complexity bounds that hold for every tractable KB are exhibited. Considering four classes of tractable formulas (the Blake class, the DNF class, the Horn cover class and finally the renamable Horn cover class), further complexity results are given. For each class, the FORMULA INFERENCE problem and two restrictions of it, the CLAUSE INFERENCE problem and the LITERAL INFERENCE problem, are successively considered.

Several new tractable subcases are exhibited. The effects of some restrictions on the partition $\langle P, Q, Z\rangle$ of the variables of the KB are also analyzed from the point of view of computational complexity. The complexity results are finally discussed and some related work is presented.

4.1 Closed world inference from a tractable KB

In the following, the complexity of CWR is investigated for tractable KBs. Let us first formally state what a tractable KB is:

Definition 7 (tractable KB)
Let Σ be a formula from $PROP_{PS}$. Σ is said *tractable for clause deduction* (tractable for short) iff there exists an algorithm $ASKS?$ s.t., for every clause γ of $PROP_{PS}$, $ASKS?(\Sigma, \gamma) = true$ iff $\Sigma \models \gamma$ holds and $ASKS?$ runs in time polynomial in $|\Sigma| + |\gamma|$.

Many fragments of propositional logic gather tractable formulas. Especially, both the Horn CNF formulas, the reverse Horn CNF formulas and the Krom formulas considered in [11] are tractable.

Considering tractable KBs is helpful for the CLAUSE DEDUCTION problem, since determining if any clause is entailed by a tractable KB Σ can be achieved in polynomial time, while the problem is coNP-complete when Σ is unconstrained.

Interestingly, considering tractable KBs is also valuable for CWR since it makes the CWA FORMULA INFERENCE problem tractable.

Proposition 8 (complexity of CWA formula inference)
Let Σ be a tractable formula from $PROP_{PS}$. Let Φ be a formula from $PROP_{PS}$. Determining whether $CWA(\Sigma) \models \Phi$ holds is in P.

Furthermore, as an easy consequence of Theorem 1 from [11], it also makes the MINIMAL MODEL CHECKING problem computationally easier (if Σ is unconstrained, the problem is coNP-complete [5]):

Proposition 9 (complexity of minimal model checking)
Let Σ be a tractable formula from $PROP_{PS}$. Let I be an interpretation over $PROP_{Var(\Sigma)}$ and $\langle P, Q, Z\rangle$ a partition of $Var(\Sigma)$. Determining whether I is a $\langle P, Q, Z\rangle$-minimal model of Σ is in P.

Intuitively, the fact that closed world inference typically is at the second level of the polynomial hierarchy can be explained by the presence of two independent sources of complexity. One of them relies on classical entailment and the other one in model minimization. Focusing on tractable KBs enables ruling out one source of difficulty in the general case, and both sources in some specific cases. Thus, from the two propositions above, the following complexity bounds (Table 2) for closed world inference from tractable KBs can be derived; compared with the general case (Table 1), we can observe that the tractability restriction makes the complexity of CWR one level below in the polynomial hierarchy.

***CWA**	FORMULA INFERENCE CLAUSE INFERENCE LITERAL INFERENCE
CWA	in P
GCWA	in $\mathsf{P}^{\mathsf{NP}[\mathcal{O}(\log n)]}$
EGCWA	in coNP
CCWA	in $\mathsf{P}^{\mathsf{NP}[\mathcal{O}(\log n)]}$
ECWA	in coNP

Table 2: Complexity of closed world inference from a tractable KB (upper bound).

Proposition 8, Proposition 9 and Table 2 are direct and easy generalizations of Theorem 2 and Theorem 1 of [11].

As reported in the table above, CWA FORMULA INFERENCE from tractable KBs is polynomially solvable. Accordingly, the (basic) CWA will not be considered in the remaining sections: CWA FORMULA INFERENCE is polynomially solvable for the four classes we focus on. Obviously, this is also the case for its restrictions CWA CLAUSE INFERENCE and CWA LITERAL INFERENCE. Interestingly, this tractability result shows that non monotonic reasoning can prove easier than classical entailment in some situations [8, 9]; indeed, FORMULA DEDUCTION is coNP-complete as long as Φ is unconstrained (even if Σ is known as tractable). We can also observe from Table 2 that EGCWA FORMULA INFERENCE and ECWA FORMULA INFERENCE from tractable KBs are at most as hard as FORMULA DEDUCTION since they belong to coNP.

Finally, imposing some restrictions on the partition $\langle P, Q, Z\rangle$ enables deriving some additional complexity results for CCWA inference and ECWA inference. Indeed, whenever $P = \emptyset$, every model of Σ is $\langle P, Q, Z\rangle$-minimal, and both forms of inference reduces to classical entailment. Accordingly:

Proposition 10

- CCWA FORMULA INFERENCE and ECWA FORMULA INFERENCE from a tractable formula under the restriction $P = Q = \emptyset$ or $P = Z = \emptyset$ are coNP-hard.

- CCWA CLAUSE INFERENCE and ECWA CLAUSE INFERENCE from a tractable formula under the restriction $P = \emptyset$ are in P.

4.2 Closed world inference from a Blake formula

Let us first analyze the complexity of CWR from Blake formulas:

Definition 11 (Blake formula)
Let Σ be a formula from $PROP_{PS}$. Σ is a *Blake formula* iff Σ is a CNF formula and for every implicate γ of Σ, there exists a clause π in Σ s.t. $\pi \models \gamma$ holds.

In other words, a Blake formula [2] is a formula given by its prime implicates. Clearly enough, such formulas are tractable (for clause deduction) since for every clause γ, $\Sigma \models \gamma$ holds iff there exists π in Σ s.t. $\pi \models \gamma$ holds and this test can be easily achieved in time polynomial in $|\pi| + |\gamma|$.

Many algorithms for computing prime implicates have been proposed so far, for instance well-known Quine's iterated consensus algorithm, and its refinements (e.g., [50, 14]) (see [36] for a survey).

Proposition 12 The complexity of *CWA FORMULA INFERENCE from a Blake formula is reported in Table 3.

***CWA**	FORMULA INFERENCE
GCWA	coNP-hard and in $\mathsf{P}^{\mathsf{NP}[\mathcal{O}(\log n)]}$
EGCWA	coNP-complete
CCWA	coNP-hard and in $\mathsf{P}^{\mathsf{NP}[\mathcal{O}(\log n)]}$
ECWA	coNP-complete

Table 3: Complexity of closed world inference from a Blake formula (formula inference).

Considering the restrictions where the query is a clause or a literal enables reducing the complexity of inference in several cases:

Proposition 13 The complexity of *CWA CLAUSE INFERENCE and *CWA LITERAL INFERENCE from a Blake formula is reported in Table 4.

The following complexity results can also be derived:

Proposition 14

- CCWA FORMULA INFERENCE from a Blake formula under the restriction $Q = Z = \emptyset$ is coNP-hard.
- ECWA CLAUSE INFERENCE from a Blake formula under the restriction $Q = Z = \emptyset$ is coNP-hard.
- ECWA LITERAL INFERENCE from a Blake formula under the restriction $Q = \emptyset$ is coNP-hard.

*CWA	CLAUSE INFERENCE	LITERAL INFERENCE
GCWA	in P	in P
EGCWA	coNP-complete	in P
CCWA	in P	in P
ECWA	coNP-complete	coNP-complete

Table 4: Complexity of closed world inference from a Blake formula (clause and literal inference).

Finally, we have derived:

Proposition 15 ECWA LITERAL INFERENCE from a Blake formula under the restriction $Z = \emptyset$ is in P.

4.3 Closed world inference from a DNF formula

Let us now investigate the complexity of CWR from DNF formulas.

Definition 16 (DNF formula)
Let Σ be a formula from $PROP_{PS}$. Σ is a *DNF formula* iff Σ is a (finite) disjunction of terms.

DNF formulas are tractable (for clause deduction) since for every clause γ, $\Sigma \models \gamma$ holds iff for every term δ in Σ, $\delta \models \gamma$ holds and this test can be easily achieved in time polynomial in $|\delta| + |\gamma|$.

Several algorithms for turning a formula into DNF can be found in the literature, e.g., [47].

Proposition 17 The complexity of *CWA FORMULA INFERENCE from a DNF formula is reported in Table 5.

*CWA	FORMULA INFERENCE
GCWA	coNP-hard and in $\mathsf{P}^{\mathsf{NP}[\mathcal{O}(\log n)]}$
EGCWA	in P
CCWA	coNP-hard and in $\mathsf{P}^{\mathsf{NP}[\mathcal{O}(\log n)]}$
ECWA	coNP-complete

Table 5: Complexity of closed world inference from a DNF formula (formula inference).

Once again, considering the restrictions where the query is a clause or a literal enables reducing the complexity of inference in several cases:

Proposition 18 The complexity of *CWA CLAUSE INFERENCE and *CWA LITERAL INFERENCE from a DNF formula is reported in Table 6.

*CWA	CLAUSE INFERENCE LITERAL INFERENCE
GCWA	in P
EGCWA	in P
CCWA	coNP-hard and in $\mathsf{P}^{\mathsf{NP}[\mathcal{O}(\log n)]}$
ECWA	coNP-complete

Table 6: Complexity of closed world inference from a DNF formula (clause and literal inference).

We have also derived the following complexity results:

Proposition 19

- CCWA LITERAL INFERENCE and ECWA LITERAL INFERENCE from a DNF formula under the restriction $Z = \emptyset$ are coNP-hard.
- CCWA FORMULA INFERENCE from a DNF formula under the restriction $Q = Z = \emptyset$ is coNP-hard.
- ECWA FORMULA INFERENCE from a DNF formula under the restriction $Q = \emptyset$ is coNP-hard.
- CCWA CLAUSE INFERENCE and ECWA CLAUSE INFERENCE from a DNF formula under the restriction $Q = \emptyset$ are in P.
- ECWA FORMULA INFERENCE from a DNF formula under the restriction $Q = Z = \emptyset$ is in P.

The tractability results above are easy consequences of the corresponding tractability results for inference from a Horn cover formula since every DNF formula is a Horn cover formula (see later on).

4.4 Closed world inference from a Horn cover formula

Let us now consider a more expressive fragment of propositional logic, namely the Horn cover class.

Definition 20 (Horn cover formula)
Let Σ be a formula from $PROP_{PS}$. Σ is a *Horn cover formula* iff Σ is a (finite) disjunction of Horn CNF formulas.

Clearly enough, every DNF formula is a Horn cover formula since every term is a Horn CNF formula. Additionally, every Horn CNF formula also is a Horn cover formula.

Horn cover formulas are tractable (for clause deduction) since for every clause γ, $\Sigma \models \gamma$ holds iff for every Horn formula Φ in Σ, $\Phi \models \gamma$ holds and this test can be easily achieved in time polynomial in $|\Phi| + |\gamma|$ [18]. Note that Horn CNF formulas can be both recognized (and checked for satisfiability) in polynomial time.

Algorithms for computing Horn cover formulas can be found in [4, 3].

Proposition 21 The complexity of *CWA FORMULA INFERENCE from a Horn cover formula is reported in Table 7.

***CWA**	FORMULA INFERENCE
GCWA	coNP-hard and in $\mathsf{P}^{\mathsf{NP}[\mathcal{O}(\log n)]}$
EGCWA	in P
CCWA	coNP-hard and in $\mathsf{P}^{\mathsf{NP}[\mathcal{O}(\log n)]}$
ECWA	coNP-complete

Table 7: Complexity of closed world inference from a Horn cover formula (formula inference).

Like for DNF KBs, restricting the queries to clauses or literals enables deriving some additional tractable subcases.

Proposition 22 The complexity of *CWA CLAUSE INFERENCE and *CWA LITERAL INFERENCE from a Horn cover formula is reported in Table 8.

***CWA**	CLAUSE INFERENCE LITERAL INFERENCE
GCWA	in P
EGCWA	in P
CCWA	coNP-hard and in $\mathsf{P}^{\mathsf{NP}[\mathcal{O}(\log n)]}$
ECWA	coNP-complete

Table 8: Complexity of closed world inference from a Horn cover formula (clause and literal inference).

We have derived the following complexity results:

Proposition 23

- CCWA LITERAL INFERENCE and ECWA LITERAL INFERENCE from a Horn cover formula under the restriction $Z = \emptyset$ are coNP-hard.
- CCWA FORMULA INFERENCE from a Horn cover formula under the restriction $Q = Z = \emptyset$ is coNP-hard.
- ECWA FORMULA INFERENCE from a Horn cover formula under the restriction $Q = \emptyset$ is coNP-hard.
- CCWA CLAUSE INFERENCE and ECWA CLAUSE INFERENCE from a Horn cover formula under the restriction $Q = \emptyset$ are in P.
- ECWA FORMULA INFERENCE from a Horn cover formula under the restriction $Q = Z = \emptyset$ is in P.

The intractability results above are direct consequences of the corresponding results for DNF formulas.

4.5 Closed world inference from a renamable Horn cover formula

Let us finally consider a still larger fragment of propositional logic, the set of all renamable Horn cover formulas.

Definition 24 (renamable Horn cover formula)
Let Σ be a formula from $PROP_{PS}$. Σ is a *renamable Horn formula* iff Σ is a (finite) disjunction of renamable Horn CNF formulas.

Renamable Horn cover formulas are tractable (for clause deduction) since for every clause γ, $\Sigma \models \gamma$ holds iff for every formula Φ in Σ, $\Phi \models \gamma$ holds, and such a test can be achieved in time polynomial in $|\Phi| + |\gamma|$ [32].

Algorithms for computing renamable Horn cover formulas can be found in [4]. Note that renamable Horn CNF formulas can be recognized (and checked for satisfiability) in polynomial time. Note also that every satisfiable Krom formula is a renamable Horn CNF one, and that every Horn cover formula is a renamable Horn cover one.

Proposition 25 The complexity of *CWA FORMULA INFERENCE, *CWA CLAUSE INFERENCE and *CWA LITERAL INFERENCE from a renamable Horn cover formula are reported in Table 9.

***CWA**	FORMULA INFERENCE CLAUSE INFERENCE LITERAL INFERENCE
GCWA	coNP-hard and in $\mathsf{P}^{\mathsf{NP}[\mathcal{O}(\log n)]}$
EGCWA	coNP-complete
CCWA	coNP-hard and in $\mathsf{P}^{\mathsf{NP}[\mathcal{O}(\log n)]}$
ECWA	coNP-complete

Table 9: Complexity of closed world inference from a renamable Horn cover formula.

The following intractability results can be easily recovered from Theorem 5, Corollary 12 and Theorem 16 from [11], because every satisfiable Krom formula is a renamable Horn cover one.

Proposition 26

- EGCWA CLAUSE INFERENCE and ECWA CLAUSE INFERENCE from a renamable Horn cover formula Σ under the restriction Σ contains only one positive Krom formula and $Q = Z = \emptyset$ are coNP-hard.
- ECWA LITERAL INFERENCE from a renamable Horn cover formula under the restriction Σ contains only one positive Krom formula and $Q = \emptyset$ is coNP-hard.

We have also derived the following results:

Proposition 27

- GCWA LITERAL INFERENCE and EGCWA LITERAL INFERENCE from a renamable Horn cover formula under the restriction Σ contains only satisfiable Krom formulas are coNP-hard.

- CCWA LITERAL INFERENCE and ECWA LITERAL INFERENCE from a renamable Horn cover formula under the restriction Σ contains only satisfiable Krom formulas and $Q = Z = \emptyset$ are coNP-hard.

4.6 Discussion and related works

The main complexity results presented in the previous sections are summarized in Tables 10, 11 and 12. They show some extent to which focusing on fragments of propositional logic where CLAUSE DEDUCTION is polynomially solvable can prove helpful for enhancing closed world inference. Accordingly, they allow new conclusions about the applicability of closed world inference to be drawn.

At the light of these results, we can observe that, given a tractable class, the most sophisticated form of closed world inference, namely inference under ECWA (or circumscription), is typically the hardest w.r.t. computational complexity. This is not very surprising and coheres with what can be observed when focusing on the classes studied in [11]. Among the tractability results above, the polynomiality of EGCWA FORMULA INFERENCE from Horn cover formulas is worth to be noted, since a quite large fragment of propositional logic is offered from which closed world inference is easier than classical entailment (FORMULA DEDUCTION from this class is coNP-complete). Following [8, 9], this gives some further evidence that closed world inference can be more efficient than classical entailment in some situations.

This work can be related to several studies where the complexity of non monotonic inference from tractable fragments of propositional logic is analyzed, for instance [29, 11, 1]. Especially, it is close to [11], where the complexity of CWR from Horn CNF formulas and Krom formulas is investigated. Nevertheless, our study extends [11] in at least two directions.

On the one hand, the tractability results reported in the tables above are about classes of formulas that are strictly more expressive than some of the most expressive tractable classes for CWR pointed out up to now. Especially, they include some of the tractability results given in [11] as subcases. Thus, every Krom formula can be turned into its prime implicates normal form in polynomial time (just because a set of binary clauses only has a polynomially bounded number of resolvents). Additionally, every Horn CNF formula belongs to the Horn cover class. However, the converse does not hold: the fragments studied in this paper are strictly more expressive that those given in [11]. For instance, every monotonic CNF formula (i.e., a CNF formula where every variable occurs only positively or negatively) can be put into prime implicates normal form in polynomial time but it is not equivalent to a Horn CNF or a Krom formula in the general case.

On the other hand, our study is directly related to exact knowledge compilation in the sense that the fragments of propositional logic we consider are target classes for exact compilation functions. Thus, every propositional formula can be turned into a Blake formula, a DNF formula, a Horn cover formula and a renamable Horn cover

Tractable KB	***CWA**	FORMULA INFERENCE	CLAUSE INFERENCE	LITERAL INFERENCE
Blake	CWA	in P	in P	in P
Blake	GCWA	coNP-hard and in $\mathsf{P}^{\mathsf{NP}[\mathcal{O}(\log n)]}$	in P	in P
Blake	EGCWA	coNP-complete	coNP-complete	in P
Blake	CCWA	coNP-hard and in $\mathsf{P}^{\mathsf{NP}[\mathcal{O}(\log n)]}$	in P	in P
Blake	ECWA	coNP-complete	coNP-complete	coNP-complete
DNF	CWA	in P	in P	in P
DNF	GCWA	coNP-hard and in $\mathsf{P}^{\mathsf{NP}[\mathcal{O}(\log n)]}$	in P	in P
DNF	EGCWA	in P	in P	in P
DNF	CCWA	coNP-hard and in $\mathsf{P}^{\mathsf{NP}[\mathcal{O}(\log n)]}$	coNP-hard and in $\mathsf{P}^{\mathsf{NP}[\mathcal{O}(\log n)]}$	coNP-hard and in $\mathsf{P}^{\mathsf{NP}[\mathcal{O}(\log n)]}$
DNF	ECWA	coNP-complete	coNP-complete	coNP-complete
Horn cover	CWA	in P	in P	in P
Horn cover	GCWA	coNP-hard and in $\mathsf{P}^{\mathsf{NP}[\mathcal{O}(\log n)]}$	in P	in P
Horn cover	EGCWA	in P	in P	in P
Horn cover	CCWA	coNP-hard and in $\mathsf{P}^{\mathsf{NP}[\mathcal{O}(\log n)]}$	coNP-hard and in $\mathsf{P}^{\mathsf{NP}[\mathcal{O}(\log n)]}$	coNP-hard and in $\mathsf{P}^{\mathsf{NP}[\mathcal{O}(\log n)]}$
Horn cover	ECWA	coNP-complete	coNP-complete	coNP-complete
renamable Horn cover	CWA	in P	in P	in P
renamable Horn cover	GCWA	coNP-hard and in $\mathsf{P}^{\mathsf{NP}[\mathcal{O}(\log n)]}$	coNP-hard and in $\mathsf{P}^{\mathsf{NP}[\mathcal{O}(\log n)]}$	coNP-hard and in $\mathsf{P}^{\mathsf{NP}[\mathcal{O}(\log n)]}$
renamable Horn cover	EGCWA	coNP-complete	coNP-complete	coNP-complete
renamable Horn cover	CCWA	coNP-hard and in $\mathsf{P}^{\mathsf{NP}[\mathcal{O}(\log n)]}$	coNP-hard and in $\mathsf{P}^{\mathsf{NP}[\mathcal{O}(\log n)]}$	coNP-hard and in $\mathsf{P}^{\mathsf{NP}[\mathcal{O}(\log n)]}$
renamable Horn cover	ECWA	coNP-complete	coNP-complete	coNP-complete

Table 10: The complexity of CWR from tractable KBs.

Tractable KB	***CWA**	FORMULA INFERENCE	CLAUSE INFERENCE	LITERAL INFERENCE
Blake	CCWA	-	*	*
Blake	ECWA	-	$P = \emptyset$	$P = \emptyset$ or $Z = \emptyset$
DNF	CCWA	-	$P = \emptyset$ or $Q = \emptyset$	$P = \emptyset$ or $Q = \emptyset$
DNF	ECWA	$Q = Z = \emptyset$	$P = \emptyset$ or $Q = \emptyset$	$P = \emptyset$ or $Q = \emptyset$
Horn cover	CCWA	-	$P = \emptyset$ or $Q = \emptyset$	$P = \emptyset$ or $Q = \emptyset$
Horn cover	ECWA	$Q = Z = \emptyset$	$P = \emptyset$ or $Q = \emptyset$	$P = \emptyset$ or $Q = \emptyset$
renamable Horn cover	CCWA	-	$P = \emptyset$	$P = \emptyset$
renamable Horn cover	ECWA	-	$P = \emptyset$	$P = \emptyset$

Table 11: Tractable restrictions for CCWA and ECWA inference ("-" means that no tractable restriction has been identified; "*" means that inference is always tractable.).

Tractable KB	***CWA**	FORMULA INFERENCE	CLAUSE INFERENCE	LITERAL INFERENCE
Blake	CCWA	$P = Q = \emptyset$ or $P = Z = \emptyset$ or $Q = Z = \emptyset$	-	-
Blake	ECWA	$P = Q = \emptyset$ or $P = Z = \emptyset$ or $Q = Z = \emptyset$	$Q = Z = \emptyset$	$Q = \emptyset$
DNF	CCWA	$P = Q = \emptyset$ or $P = Z = \emptyset$ or $Q = Z = \emptyset$	$Z = \emptyset$	$Z = \emptyset$
DNF	ECWA	$P = Q = \emptyset$ or $P = Z = \emptyset$	$Z = \emptyset$	$Z = \emptyset$
Horn cover	CCWA	$P = Q = \emptyset$ or $P = Z = \emptyset$ or $Q = Z = \emptyset$	$Z = \emptyset$	$Z = \emptyset$
Horn cover	ECWA	$P = Q = \emptyset$ or $P = Z = \emptyset$	$Z = \emptyset$	$Z = \emptyset$
renamable Horn cover	CCWA	$P = Q = \emptyset$ or $P = Z = \emptyset$ or $Q = Z = \emptyset$	$Q = Z = \emptyset$	$Q = Z = \emptyset$
renamable Horn cover	ECWA	$P = Q = \emptyset$ or $P = Z = \emptyset$ or $Q = Z = \emptyset$	$Q = Z = \emptyset$	$Q = Z = \emptyset$

Table 12: Restrictions under which CCWA and ECWA inference are coNP-hard. ("-" means that no intractable restriction exists.)

formula without modifying the set of its models. This contrasts with the fragments considered in [11] since some formulas cannot be represented either as a Horn CNF, a reverse Horn CNF or a Krom formula while preserving logical equivalence.

Accordingly, through the complexity results given in the previous sections, new conclusions about the usefulness of knowledge compilation for CWR can be drawn, both from the theoretical side and from the practical side.

From the theoretical side, the compilability results for CWR given in [8, 9] can be completed. Thus, CWA FORMULA INFERENCE from any tractable KB, and more generally, all the problems that have been shown in P are compilable. Indeed, *every polynomially solvable problem is compilable*, even if its fixed part is empty since the compilation phase can be achieved in polynomial time. More interestingly:

Proposition 28 (compilability results for CWR from tractable KBs)

- [GCWA CLAUSE INFERENCE, Σ, γ] is compilable.
- [CCWA CLAUSE INFERENCE, $\langle \Sigma, \langle P, Q, Z \rangle \rangle, \gamma$] is compilable.

Indeed, in contrast to what happens for EGCWA and ECWA in the general case, the number of GCWA-ffn and CCWA-ffn formulas to be considered are polynomially bounded in the size of the KB.

Contrastingly, FORMULA INFERENCE from a tractable KB under *CWA is (probably) not compilable when it is not in P. Indeed, FORMULA DEDUCTION from a tractable KB is known as (probably) not compilable[4], and in some situations, *CWA FORMULA INFERENCE reduces to FORMULA DEDUCTION (e.g., when the CWA policy is ECWA and the partition is $\langle \emptyset, Q, Z \rangle$).

From the practical side, the connection with exact knowledge compilation suggests a two step approach to compute closed world inference: the KB is first made tractable for clause deduction through knowledge compilation, then CWR is achieved from the compiled KB. The first step is performed off-line and done once at all (unless the KB is modified). Such a two step process contrasts with several existing algorithms for CWR (see e.g., [24, 43, 28, 26, 27] for circumscription). Within our two-step approach, the fixed part of the closed world inference problem for compilation is the KB. Thus, the closed world assumption policy adopted can vary with the queries considered. For instance, the set of variables to be minimized can change with time without requiring the KB to be re-compiled each time such a modification occurs. This is particularly valuable when the KB, representing "hard" constraints, is shared by several agents, and different agents may have different preferences encoded as CWA policies. Each time an agent queries the KB interpreted under some closed world assumption, the corresponding preferences can be taken into account without requiring any re-compilation of the KB. Such a flexibility is typically not offered by the approaches [41, 44, 12] where closed world inference is reduced to classical entailment through the computation of the closure of the KB.

Obviously, since CWR is (probably) not compilable [8, 9, 10], such a two step knowledge compilation-based approach cannot be expected as valuable for improving CWR in the worst case. Nevertheless, the (probable) non-compilability of closed world

[4]Theorem 4 from [7] still holds when the considered KB is assumed tractable, since *true* (or any formula known to be valid) is tractable for clause deduction.

inference should not be misinterpreted: this theoretical result concerns the worst case, only, and does not prevent from exhibiting instances where knowledge compilation is practically helpful. Accordingly, many experiments with various compilation functions show the practical utility of exact knowledge compilation for improving CLAUSE DEDUCTION (see e.g. [47, 4]) though the problem is known as (probably) not compilable. Thus, at the light of the complexity results given in the previous sections, our opinion is that in the many situations where some compilation techniques are computationally profitable for CLAUSE DEDUCTION, they can be profitable as well for some forms of closed world inference. For instance, compiling a KB into a Horn cover formula so as to interpret it under EGCWA reduces the complexity of inference from Π_2^p-complete to P. Thus, if the compilation phase does not result in an exponential blow up in the size of the KB, the time needed for inference from the compiled KB can be much lower than the corresponding time from the original KB; subsequently, the compilation time can be balanced.

At that point, a *knowledge compilation policy*, i.e., a way to map a KB and a closed world inference problem to a class of tractable formulas, would be helpful. Such a policy would enable determining the class to which the KB has to be compiled, so as to achieve on-line CWR as efficiently as possible. Moreover, if no improvement is possible compared with a direct, uncompiled approach, the compilation policy would have to state it. Unfortunately, such a policy does not exist. Especially, there is no way to predict in polynomial time the size of the compiled form of a KB, whatever the compilation function: the best we can do is to select one target class for compilation, and interrupt the compilation phase as soon as the size of the current compiled KB exceeds a preset bound. At that time, only heuristic, a priori choices of target classes can be made. They are guided by two, possibly conflicting, criteria[5]:

- The computational complexity of on-line closed world inference from the compiled KB: classes for which CWR is polynomially solvable must be preferred to classes for which CWR is at the first level of the polynomial hierarchy.
- The expressiveness of the target classes for compilation: most expressive classes must be preferred.

In the objective of improving closed world inference through exact knowledge compilation, our results show that focusing on the Horn cover class can prove much better than focusing on the DNF class since:

- Closed world inference from a DNF formula and closed world inference from a Horn cover formula have the same complexity.
- Every DNF formula is a Horn cover one, but the converse is false. Moreover, the size of a Horn cover formula encoding of a KB can be exponentially smaller than the size of its most compact DNF encoding. Besides, from a practicalside , experiments show that generating a Horn cover encoding of a CNF KB often prove less time-consuming than generating a DNF encoding of it [4].

It is not easy to draw some other conclusions. Thus, compiling a KB into a renamable Horn cover formula for improving EGCWA inference can be better in practice

[5]Tractability of inference and compactness of representation are two important features of any knowledge representation formalism, see e.g. [9].

than compiling it into a Horn cover formula (despite the fact that the complexity of inference in the first case is coNP-complete and in P in the second one). Indeed, if the size of the Horn cover formula that is derived is exponentially greater than the size of the renamable Horn cover formula, inference from the latter can prove practically more efficient than inference from the former.

5 Conclusion

In this paper, the complexity of closed world inference from some tractable fragments of propositional logic has been investigated. Several new tractable cases have been identified, and some new intractability results have been provided as well. Interestingly, the tractable fragments we have focused on are target classes for some exact compilation functions; in our opinion, such a connection with knowledge compilation can prove valuable from a computational point of view in the situations where the size of the compiled KB remains "small enough".

This work can be extended in several directions. First of all, due to close connections between CWR and abduction [28, 27], and between CWR and some simple forms of default reasoning [29, 25, 49], complexity results for such forms of non monotonic inference from tractable KBs can be derived from the results presented in this paper. This is a first issue for further research. Another issue concerns the empirical evaluation of the two step approach to compute some forms of CWR that is suggested in the previous section. For instance, it would be nice to estimate the computational savings that could be achieved by performing EGCWA formula inference via a preliminary off-line compilation of the KB into a Horn cover formula, compared with a direct, "uncompiled" approach.

References

[1] R. Ben-Eliyahu and R. Dechter. Default reasoning using classical logic. *Artificial Intelligence*, 84:113–150, 1996.

[2] A. Blake. *Canonical expressions in Boolean algebra.* PhD thesis, University of Chicago, Chicago (IL), 1937.

[3] Y. Boufkhad. Algorithms for propositional KB approximation. In *Proc. of the* 15^{th} *National Conference on Artificial Intelligence (AAAI'98)*, pages 280–285, Madison (WI), 1998.

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

[5] M. Cadoli. Semantical and computational aspects of Horn approximation. In *Proc. of the* 13^{th} *International Joint Conference on Artificial Intelligence (IJCAI'93)*, pages 39–44, Chambéry, 1993.

[6] M. Cadoli and F.M. Donini. A survey on knowledge compilation. *AI Communications*, 10:137–150, 1997. printed in 1998.

[7] M. Cadoli, F.M. Donini, P. Liberatore, and M. Schaerf. Preprocessing of intractable problems. Technical Report DIS 24-97, Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza", 1997.

[8] M. Cadoli, F.M. Donini, and M. Schaerf. Is intractability of non monotonic reasoning a real drawback? In *Proc. of the* 12^{th} *National Conference on Artificial Intelligence (AAAI'94)*, pages 946–951, Austin (TX), 1994.

[9] M. Cadoli, F.M. Donini, and M. Schaerf. Is intractability of non-monotonic reasoning a real drawback? *Artificial Intelligence*, 88:215–251, 1996.

[10] M. Cadoli, F.M. Donini, M. Schaerf, and R. Silvestri. On compact representations of propositional circumscription. *Theoretical Computer Science*, 182:183–202, 1997.

[11] M. Cadoli and M. Lenzerini. The complexity of propositional closed world reasoning and circumscription. *Journal of Computer and System Sciences*, 48:255–310, 1994.

[12] Th. Castell, C. Cayrol, M. Cayrol, and D. Le Berre. Using the Davis and Putnam procedure for an efficient computation of preferred models. In *Proc. of the* 12^{th} *European Conference on Artificial Intelligence (ECAI'96)*, pages 350–354, Budapest, 1996.

[13] S. Coste-Marquis and P. Marquis. Complexity results for propositional closed world reasoning and circumscription from tractable knowledge bases. In *Proc. of the* 14^{th} *International Joint Conference on Artificial Intelligence (IJCAI'99)*, pages 24–29, Stockholm (Sweden), 1999.

[14] J. de Kleer. An improved incremental algorithm for generating prime implicates. In *Proc. of the* 10^{th} *National Conference on Artificial Intelligence (AAAI'92)*, pages 780–785, San Jose (CA), 1992.

[15] R. Dechter and I. Rish. Directional resolution: the Davis-Putnam procedure, revisited. In *Proc. of the* 4^{th} *International Conference on Principles of Knowledge Representation and Reasoning (KR'94)*, pages 134–145, Bonn, 1994.

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

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

[18] W. Dowling and J.H. Gallier. Linear time algorithms for testing the satisfiability of propositional Horn formulae. *Journal of Logic Programming*, 1(3):267–284, 1984.

[19] Th. Eiter and G. Gottlob. Propositional circumscription and extended closed-world reasoning are Π_2^p-complete. *Theoretical Computer Science*, 114:231–245, 1993.

[20] H.B. Enderton. *A mathematical introduction to logic.* Academic Press, New York, 1972.

[21] M.R. Garey and D.S. Johnson. *Computers and intractability: a guide to the theory of NP-completeness.* Freeman, 1979.

[22] M. Gelfond and H. Przymusinska. Negation as failure: careful closure procedure. *Artificial Intelligence*, 30:273–287, 1986.

[23] M. Gelfond, H. Przymusinska, and T. Przymusinski. On the relationship between circumscription and negation as failure. *Artificial Intelligence*, 38:49–73, 1989.

[24] M. Ginsberg. A circumscriptive theorem prover. *Artificial Intelligence*, 39:109–230, 1989.

[25] G. Gottlob. Complexity results for nonmonotonic logics. *Journal of Logic and Computation*, 2:397–425, 1992.

[26] K. Inoue. Consequence-finding based on ordered linear resolution. In *Proc. of the 12^{th} International Joint Conference on Artificial Intelligence (IJCAI'91)*, pages 158–164, Sydney, 1991.

[27] K. Inoue. Linear resolution in consequence–finding. *Artificial Intelligence*, 56(2–3):301–353, 1992.

[28] K. Inoue and N. Helft. On theorem provers for circumscription. In *Proc. of the 8^{th} Biennal Conference of the Canadian Society for Computational Studies of Intelligence (CSCSI'90)*, pages 212–219, Ottawa, 1990.

[29] H.A. Kautz and B. Selman. Hard problems for simple default logic. *Artificial Intelligence*, 49:243–279, 1991.

[30] H.A. Kautz and B. Selman. Forming concepts for fast inference. In *Proc. of the 10^{th} National Conference on Artificial Intelligence (AAAI'92)*, pages 786–793, San Jose (CA), 1992.

[31] R.C.T. Lee. *A completeness theorem and a computer program for finding theorems derivable from given axioms.* PhD thesis, University of California, Berkeley (CA), 1967.

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

[33] P. Liberatore. On the compilability of diagnosis, planning, reasoning about actions, belief revision, etc. In *Proc. of the 6^{th} International Conference on Principles of Knowledge Representation and Reasoning (KR'98)*, pages 144–155, Trento, 1998.

[34] V. Lifschitz. Computing circumscription. In *Proc. of the* 9^{th} *International Joint Conference on Artificial Intelligence (IJCAI'85)*, pages 121–127, Los Angeles (CA), 1985.

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

[36] P. Marquis. *Consequence finding algorithms*, volume 5 of *The Handbook for Uncertain and Defeasible Reasoning*, chapter Algorithms for Uncertain and Defeasible Reasoning. Kluwer Academic Publishers, 2000. (to appear).

[37] J. McCarthy. Circumscription: a form of non–monotonic reasoning. *Artificial Intelligence*, 13:27–39, 1980.

[38] J. McCarthy. Applications of circumscription to formalizing common-sense knowledge. *Artificial Intelligence*, 28:89–116, 1986.

[39] J. Minker. On indefinite databases and the closed world assumption. In *Proc. of the* 6^{th} *Conference on Automated Deduction (CADE'82)*, pages 292–308, 1982.

[40] B. Nebel. *Belief Revision*, volume 3 of *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, chapter How hard is it to revise a belief base?, pages 77–145. Kluwer Academic, 1998.

[41] A. Nerode, R.T. Ng, and V.S. Subrahmanian. Computing circumscriptive databases. I: Theory and algorithms. *Information and Computation*, 116:58–80, 1995.

[42] Ch. H. Papadimitriou. *Computational complexity.* Addison–Wesley, 1994.

[43] T.C. Przymusinski. An algorithm to compute circumscription. *Artificial Intelligence*, 38:49–73, 1989.

[44] O. Raiman and J. de Kleer. A minimality maintenance system. In *Proc. of the* 3^{rd} *International Conference on Principles of Knowledge Representation and Reasoning (KR'92)*, pages 532–538, Cambridge (MA), 1992.

[45] R. Reiter. *Logic and Data Bases*, chapter On closed world data bases, pages 119–140. Plenum Press, 1978.

[46] R. Reiter and J. de Kleer. Foundations of assumption-based truth maintenance systems: preliminary report. In *Proc. of the* 6^{th} *National Conference on Artificial Intelligence (AAAI'87)*, pages 183–188, Seattle (WA), 1987.

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

[48] B. Selman and H.A. Kautz. Knowledge compilation and theory approximation. *Journal of the Association for Computing Machinery*, 43:193–224, 1996.

[49] J. Stillman. The complexity of propositional default logics. In *Proc. of the* 10^{th} *National Conference on Artificial Intelligence (AAAI'92)*, pages 794–799, San Jose (CA), 1992.

[50] P. Tison. Generalization of consensus theory and application to the minimization of boolean functions. *IEEE Transactions on Electronic Computers*, EC–16:446–456, 1967.

[51] A. Yahya and L.J. Henschen. Deduction in non-Horn databases. *Journal of Automated Reasoning*, 1:141–160, 1985.

Appendix

Proof (Proposition 8) It is well-known that the closure of Σ w.r.t. CWA is logically equivalent to the unique minimal model MM of Σ when $CWA(\Sigma)$ is satisfiable (see Lemma 3.5 in [19]). Thus, $CWA(\Sigma) \models \Phi$ holds iff MM exists and $MM \models \Phi$ holds, or $CWA(\Sigma)$ is unsatisfiable (then everything follows). The point is that the closure of Σ w.r.t. CWA can be computed in time polynomial in $|\Sigma|$ whenever Σ is tractable. Indeed, it is sufficient to check for every variable v of $Var(\Sigma)$ whether or not $\Sigma \not\models v$, and such a test can be achieved in polynomial time whenever Σ is tractable. Starting with an empty set S of literals, v is added to S when $\Sigma \models v$ holds and $\neg v$ is added to S otherwise. Once all the variables of $Var(\Sigma)$ have been considered, we check whether $S \models \Sigma$ holds (clearly, this can be achieved in polynomial time). If $S \models \Sigma$ holds, then S represents the unique minimal model MM of Σ (given that all the variables from $PS \setminus Var(\Sigma)$ are considered false in MM); checking whether $MM \models \Phi$ holds can then be done in polynomial time in $|S| + |\Phi|$ by replacing in a first step every variable from $Var(\Phi) \setminus Var(\Sigma)$ by $false$ in Φ. In the remaining case, $CWA(\Sigma)$ is not satisfiable, and $CWA(\Sigma) \models \Phi$ trivially holds.

Proof (Proposition 9) Theorem 1 from [11] shows that I is a $\langle P, Q, Z\rangle$-minimal model of Σ iff for every variable p of P s.t. $I \models \Sigma$ holds, $\Sigma \wedge \neg p \wedge \{\neg p'$; $p' \in P$ and $I \not\models p'\} \wedge \{q$; $q \in Q$ and $I \models q\} \wedge \{\neg q'$; $q' \in Q$ and $I \not\models q'\}$ is unsatisfiable. This is equivalent to state that I is a $\langle P, Q, Z\rangle$-minimal model of Σ iff for every variable p of P s.t. $I \models \Sigma$ holds, $\Sigma \models (p \vee \{p'$; $p' \in P$ and $I \not\models p'\} \vee \{\neg q$; $q \in Q$ and $I \models q\} \vee \{q'$; $q' \in Q$ and $I \not\models q'\})$ holds. And this is testable in time polynomial in $|\Sigma| + |I|$ whenever Σ is tractable.

Proof (Proposition 10) In the situation where $P = \emptyset$, every model of Σ is a $\langle P, Q, Z\rangle$-minimal model of it. Since $ECWA(\Sigma, \langle\emptyset, Q, Z\rangle) \models \Phi$ holds iff $[\![\Phi]\!](MM) = true$ for every $\langle P, Q, Z\rangle$-minimal model MM of Σ (Corollary 4.7 from [23]), we have $ECWA(\Sigma, \langle\emptyset, Q, Z\rangle) \models \Phi$ iff $\Sigma \models \Phi$ holds. Now, in the situation where $P = \emptyset$, the closure of Σ w.r.t. CCWA is equivalent to Σ. Hence, we have $CCWA(\Sigma, \langle\emptyset, Q, Z\rangle) \models \Phi$ iff $\Sigma \models \Phi$ holds. Finally:

- CCWA FORMULA INFERENCE and ECWA FORMULA INFERENCE from a tractable formula under the restriction $P = Q = \emptyset$ or $P = Z = \emptyset$. Let Φ be any formula from $PROP_{PS}$. Let F be the mapping s.t. $F(\Phi) = \langle true, \langle\emptyset, \emptyset, Var(\Phi)\rangle, \Phi\rangle$. Clearly enough, $F(\Phi)$ can be computed in time polynomial in $|\Phi|$. Moreover, $true$ is a tractable formula (for clause deduction). Now, Φ is valid iff

$CCWA(\Sigma, \langle\emptyset, \emptyset, Var(\Phi)\rangle) \models \Phi$ iff $ECWA(\Sigma, \langle\emptyset, \emptyset, Var(\Phi)\rangle) \models \Phi$. Because TAUT (the validity problem of a propositional formula) is a well-known coNP-complete problem, the polynomial reduction F shows the coNP-hardness of both CCWA FORMULA INFERENCE and ECWA FORMULA INFERENCE from a tractable formula under the restriction $P = Q = \emptyset$. As to the restriction $P = Z = \emptyset$, it is sufficient to replace F by F' s.t. $F'(\Phi) = \langle true, \langle\emptyset, Var(\Phi), \emptyset\rangle, \Phi\rangle$ to obtain the desired conclusion.

- CCWA CLAUSE INFERENCE and ECWA CLAUSE INFERENCE from a tractable formula under the restriction $P = \emptyset$. The membership to P is trivial because $CCWA(\Sigma, \langle\emptyset, Q, Z\rangle) \equiv ECWA(\Sigma, \langle\emptyset, Q, Z\rangle) \equiv \Sigma$ hold.

Proof (Proposition 12) The upper bounds for GCWA FORMULA INFERENCE, EGCWA FORMULA INFERENCE, CCWA FORMULA INFERENCE and ECWA FORMULA INFERENCE are consequences of Theorem 1 from [11]. The coNP-hardness of EGCWA FORMULA INFERENCE (resp. ECWA FORMULA INFERENCE) is a direct consequence of the coNP-hardness of EGCWA CLAUSE INFERENCE (resp. ECWA CLAUSE INFERENCE). The coNP-hardness of GCWA FORMULA INFERENCE and C-CWA FORMULA INFERENCE is a direct consequence of the coNP-hardness of CCWA FORMULA INFERENCE under the restriction $Q = Z = \emptyset$ (cf. Proposition 14) since for every formula Φ, $GCWA(\Sigma) \models \Phi$ holds iff $CCWA(\Sigma, \langle Var(\Sigma), \emptyset, \emptyset\rangle) \models \Phi$ holds [22].

Proof (Proposition 13) The upper bounds for EGCWA CLAUSE INFERENCE and ECWA CLAUSE INFERENCE follow from the upper bounds given for the corresponding formula inference problems (cf. Proposition 12). The coNP-hardness of ECWA CLAUSE INFERENCE is a direct consequence of the coNP-hardness of ECWA LITERAL INFERENCE, and the coNP-hardness of ECWA LITERAL INFERENCE is a direct consequence of the coNP-hardness of ECWA LITERAL INFERENCE under the restriction $Q = \emptyset$ (cf. Proposition 14). Additionally, the coNP-hardness of EGCWA CLAUSE INFERENCE is a direct consequence of the coNP-hardness of ECWA CLAUSE INFERENCE under the restriction $Q = Z = \emptyset$ (cf. Proposition 14) since for every formula Φ, $EGCWA(\Sigma) \models \Phi$ iff $ECWA(\Sigma, \langle Var(\Sigma), \emptyset, \emptyset\rangle) \models \Phi$ holds [23]. The membership to P of GCWA LITERAL INFERENCE (resp. CCWA LITERAL INFERENCE) is a direct consequence of the membership to P of GCWA CLAUSE INFERENCE (resp. CCWA CLAUSE INFERENCE). Hence, only the membership to P of GCWA CLAUSE INFERENCE and of CCWA CLAUSE INFERENCE remain to be considered.

- CCWA CLAUSE INFERENCE. The point is that the CCWA-ffn formulas w.r.t. $\langle P, Q, Z\rangle$ can be computed in time polynomial in the size of the input whenever Σ is a Blake formula. Indeed, let l be a positive literal (i.e., a variable) occurring in Σ. l is free for negation w.r.t. CCWA and $\langle P, Q, Z\rangle$ iff every implicate of $\Sigma \wedge \neg l$ containing only literals from $L_P^+ \cup L_Q$ also is an implicate of Σ iff every prime implicate of $\Sigma \wedge \neg l$ containing only literals from $L_P^+ \cup L_Q$ also is a prime implicate of Σ [28]. Interestingly, the prime implicates of $\Sigma \wedge \neg l$ that are not prime implicates of Σ can be computed in time polynomial in $|\Sigma|$ when Σ is given by its prime implicates and l is a literal (see Corollary 4 from [46]). Accordingly, it is possible to check efficiently (i.e., in polynomial time) for every candidate l whether or not it is free for negation w.r.t. CCWA and $\langle P, Q, Z\rangle$. Since the number of candidates also is polynomial in $|\Sigma|$, the set $\{l_1, ..., l_k\}$ of CCWA-ffn formulas w.r.t. $\langle P, Q, Z\rangle$ can be computed in polynomial time. The closure of Σ w.r.t. CCWA and $\langle P, Q, Z\rangle$ is $\Sigma \cup \{\neg l_1, ..., \neg l_k\}$. Accordingly, for every clause γ, we have $CCWA(\Sigma, \langle P, Q, Z\rangle) \models \gamma$ iff $\Sigma \cup \{\neg l_1, ..., \neg l_k\} \models \gamma$ iff $\Sigma \models (l_1 \vee ... \vee l_k \vee \gamma)$. Such a deduction check can be achieved in polynomial time in $|\Sigma| + |\gamma|$ since Σ is tractable.

- GCWA CLAUSE INFERENCE. The membership to P of GCWA CLAUSE INFERENCE is a direct consequence of the membership to P of GCWA FORMULA INFERENCE since for every formula Φ, $GCWA(\Sigma) \models \Phi$ holds iff $CCWA(\Sigma, \langle Var(\Sigma), \emptyset, \emptyset\rangle) \models \Phi$ holds [22].

Proof (Proposition 14)

- CCWA FORMULA INFERENCE under the restriction $Q = Z = \emptyset$. Let F be the mapping s.t. $F(\Phi) = \langle \Sigma, \Phi\rangle$ where $\Sigma = (\neg x_1 \vee \neg y_1) \wedge (x_1 \vee y_1)\wedge, ... \wedge(\neg x_n \vee \neg y_n) \wedge (x_n \vee y_n)\}$ and $Var(\Phi) = \{x_1, ..., x_n\}$. Clearly enough, $F(\Phi)$ can be computed in time polynomial in $|\Phi|$. Now, Σ is a Blake formula (the correctness of Tison's algorithm for computing prime implicates [50] shows it since every resolvent that can be generated from Σ is a tautology and no clause of Σ is a logical consequence of another clause of Σ). Moreover, every model MM of Σ is a minimal model since for every $i \in 1..n$, $[\![x_i]\!](MM) = false$ iff $[\![y_i]\!](MM) = true$. Accordingly, no variable v of $Var(\Sigma)$ is s.t. $[\![v]\!](MM) = false$ for every minimal model of Σ. Thus, there is no CCWA-ffn formula, i.e., $CCWA(\Sigma, \langle P, \emptyset, \emptyset\rangle) \equiv \Sigma$ holds. Finally, every interpretation over $Var(\Phi)$ is in one-to-one correspondence with a model of Σ over $Var(\Sigma)$. As a consequence, if Φ has a counter-model I over $Var(\Sigma)$, then there exists a model MM of Σ that is a counter-model of Φ. In this situation, $\Sigma \not\models \Phi$ holds. Thus, Φ is valid iff $CCWA(\Sigma, \langle P, \emptyset, \emptyset\rangle) \models \Phi$ holds. F is a polynomial reduction from TAUT to CCWA FORMULA INFERENCE from a Blake formula under the restriction $Q = Z = \emptyset$. Since the former is coNP-complete, the latter is coNP-hard.

- Let us now recall that the prime implicates of a Krom formula Σ can be computed in time polynomial in $|\Sigma|$. Indeed, the resolvents of a set of binary clauses can be generated in polynomial time (in particular, there is only a polynomial number of such resolvents since every resolvent of two binary clauses is binary). The correctness of any consensus-based algorithm for computing prime implicates (e.g., [50]) completes the proof. Once this is stated:
  - The coNP-hardness of ECWA CLAUSE INFERENCE from a Blake formula under the restriction $Q = Z = \emptyset$ is a direct consequence of the coNP-hardness of ECWA CLAUSE INFERENCE from a positive Krom formula under the same restriction (see Corollary 12 from [11]).
  - The coNP-hardness of ECWA LITERAL INFERENCE from a Blake formula under the restriction $Q = \emptyset$ is a direct consequence of the coNP-hardness of ECWA LITERAL INFERENCE from a positive Krom formula under the same restriction (see Theorem 16 from [11]).

Proof (Proposition 15) The membership to P of ECWA LITERAL INFERENCE under the restriction $Z = \emptyset$ is a direct consequence of the membership to P of CCWA

LITERAL INFERENCE (Proposition 13). Indeed, when $Z = \emptyset$, every literal from $L_{Var(\Sigma)}$ is a literal of L_P or a literal of L_Q. Let l be a literal from L_Q. As a direct consequence of Theorem 5.8 from [23], we know that $ECWA(\Sigma, \langle P, Q, \emptyset\rangle) \models l$ holds iff $\Sigma \models l$ holds. This can be tested in time polynomial in $|\Sigma|$ when Σ is tractable. Let l be a literal from L_P. As a direct consequence of Theorem 3.9 from [22], $ECWA(\Sigma, \langle P, Q, \emptyset\rangle) \models l$ holds iff $CCWA(\Sigma, \langle P, Q, \emptyset\rangle) \models l$ holds. As shown in Proposition 13, this can be tested in polynomial time.

Proof (Proposition 17) The upper bounds for GCWA FORMULA INFERENCE, CCWA FORMULA INFERENCE and ECWA FORMULA INFERENCE are consequences of Theorem 1 from [11]. The coNP-hardness of CCWA FORMULA INFERENCE and ECWA FORMULA INFERENCE are direct consequences of the coNP-hardness of CCWA LITERAL INFERENCE and ECWA LITERAL INFERENCE (cf. Proposition 18). The coNP-hardness of GCWA FORMULA INFERENCE is a direct consequence of the coNP-hardness of CCWA FORMULA INFERENCE under the restriction $Q = Z = \emptyset$ (cf. Proposition 19) since for every formula Φ, we have $GCWA(\Sigma) \models \Phi$ iff $CCWA(\Sigma, \langle Var(\Sigma), \emptyset, \emptyset\rangle) \models \Phi$ (see [22]). Finally, the membership to P of EGCWA FORMULA INFERENCE is a direct consequence of the membership to P of ECWA FORMULA INFERENCE under the restriction $Q = Z = \emptyset$ (cf. Proposition 19) since for every formula Φ, we have $EGCWA(\Sigma) \models \Phi$ iff $ECWA(\Sigma, \langle Var(\Sigma), \emptyset, \emptyset\rangle) \models \Phi$ (see [23]).

Proof (Proposition 18) The upper bounds for CCWA CLAUSE INFERENCE, ECWA CLAUSE INFERENCE, CCWA LITERAL INFERENCE and ECWA LITERAL INFERENCE are direct consequences of the corresponding upper bounds for CCWA FORMULA INFERENCE and ECWA FORMULA INFERENCE. The coNP-hardness of CCWA CLAUSE INFERENCE and ECWA CLAUSE INFERENCE are direct consequences of the coNP-hardness of CCWA LITERAL INFERENCE and ECWA LITERAL INFERENCE. The coNP-hardness of CCWA LITERAL INFERENCE and ECWA LITERAL INFERENCE are direct consequences of the coNP-hardness of their restrictions in which $Z = \emptyset$ (cf. Proposition 19). The membership to P of EGCWA CLAUSE INFERENCE and EGCWA LITERAL INFERENCE are direct consequences of the membership to P of EGCWA FORMULA INFERENCE (cf. Proposition 17). Finally, the membership to P of GCWA CLAUSE INFERENCE and GCWA LITERAL INFERENCE are consequences of the membership to P of CCWA CLAUSE INFERENCE under the restriction $Q = \emptyset$ (cf. Proposition 19) since for every formula Φ, we have $GCWA(\Sigma) \models \Phi$ iff $CCWA(\Sigma, \langle Var(\Sigma), \emptyset, \emptyset\rangle) \models \Phi$ (see [22]).

Proof (Proposition 19)

- CCWA LITERAL INFERENCE and ECWA LITERAL INFERENCE under the restriction $Z = \emptyset$. Let Σ be a DNF formula. Let F be the mapping s.t. $F(\Sigma) = \langle \Sigma \vee new, \langle \{new\}, Var(\Sigma), \emptyset\rangle, \neg new\rangle$, where new is a variable from $PS \setminus Var(\Sigma)$. Clearly enough, $F(\Sigma)$ can be computed in time polynomial in $|\Sigma|$. Σ is valid iff $ECWA(\Sigma \vee new, \langle \{new\}, Var(\Sigma), \emptyset\rangle) \models \neg new$. Indeed, if Σ is valid, then the DNF formula $\Sigma \vee new$ is valid as well. As a consequence, every interpretation over $Var(\Sigma) \cup \{new\}$ where new is interpreted as $false$ is a $\langle \{new\}, Var(\Sigma), \emptyset\rangle$-minimal model of $\Sigma \vee new$. Hence,

$ECWA(\Sigma \lor new, \langle\{new\}, Var(\Sigma), \emptyset\rangle) \models \neg new$ holds. In the remaining case, there exists a counter-model I of Σ over $Var(\Sigma)$. Let MM be the interpretation over $Var(\Sigma) \cup \{new\}$ s.t. $[\![v]\!](MM) = [\![v]\!](I)$ for every $v \in Var(\Sigma)$ and $[\![new]\!](MM) = true$. MM is a $\langle\{new\}, Var(\Sigma), \emptyset\rangle$-minimal model of $\Sigma \lor new$. Since $[\![new]\!](MM) = true$, it is not the case that every $\langle\{new\}, Var(\Sigma), \emptyset\rangle$-minimal model of $\Sigma \lor new$ is a model of $\neg new$; in other words, $ECWA(\Sigma \lor new, \langle\{new\}, Var(\Sigma), \emptyset\rangle) \models \neg new$ does not hold. Thus, F is a polynomial reduction from DNF-TAUT to ECWA LITERAL INFERENCE from a DNF formula under the restriction $Z = \emptyset$. Since DNF-TAUT (the problem of checking if a DNF formula is valid) is coNP-complete, the coNP-hardness of ECWA LITERAL INFERENCE from a DNF formula under the restriction $Z = \emptyset$ results. Theorem 3.9 from [22] shows that, for every literal l from L_P, $ECWA(\Sigma, \langle P, Q, Z\rangle) \models l$ holds iff $CCWA(\Sigma, \langle P, Q, Z\rangle) \models l$ holds. Accordingly, the polynomial reduction F above can also be viewed as a reduction from DNF-TAUT to CCWA LITERAL INFERENCE from a DNF formula under the restriction $Z = \emptyset$. This shows the coNP-hardness of this last problem.

- CCWA FORMULA INFERENCE under the restriction $Q = Z = \emptyset$. Let Φ be any formula. Let F be the mapping s.t. $F(\Phi) = \langle v_1 \lor \ldots \lor v_n \lor new, \langle Var(\Phi) \cup \{new\}, \emptyset, \emptyset\rangle, \Phi \lor \neg new\rangle$, where $Var(\Phi) = \{v_1, \ldots, v_n\}$, and $new \in PS \setminus Var(\Sigma)$. Clearly enough, $F(\Phi)$ can be computed in time polynomial in $|\Phi|$. Now, Φ is valid iff $CCWA(v_1 \lor \ldots \lor v_n \lor new, \langle Var(\Phi) \cup \{new\}, \emptyset, \emptyset\rangle) \models \Phi \lor \neg new$ holds. Indeed, for every variable v_i occurring in Φ, v_i is *not* free for negation w.r.t. CCWA and $\langle Var(\Phi) \cup \{new\}, \emptyset, \emptyset\rangle$ since expanding $v_1 \lor \ldots \lor v_n \lor new$ with $\neg v_i$ gives rise to a new positive clause. For the same reason, new is not free for negation w.r.t. CCWA. Accordingly, the closure $CCWA(v_1 \lor \ldots \lor v_n \lor new, \langle Var(\Phi), \emptyset, \emptyset\rangle)$ is equivalent to the DNF formula $v_1 \lor \ldots \lor v_n \lor new$. Now, if Φ is valid, then $\Phi \lor \neg new$ is valid as well and $v_1 \lor \ldots \lor v_n \lor new \models \Phi \lor \neg new$ holds. If Φ is not valid, then $new \models \Phi$ does *not* hold and as a consequence, $new \models \Phi \lor \neg new$ does not hold, so $v_1 \lor \ldots \lor v_n \lor new \models \Phi \lor \neg new$ does not hold. Thus, F is a polynomial reduction from TAUT (the validity problem of a propositional formula) to CCWA FORMULA INFERENCE from a DNF formula under the restriction $Q = Z = \emptyset$. The coNP-hardness of this last problem is a consequence of the coNP-completeness of TAUT.

- ECWA FORMULA INFERENCE under the restriction $Q = \emptyset$. Let Φ be any formula. Let F be the mapping s.t. $F(\Phi) = \langle v_1 \lor \neg v_1, \langle \emptyset, \emptyset, Var(\Phi)\rangle, \Phi\rangle$, where v_1 is any variable from $Var(\Phi)$ (if $Var(\Phi) = \emptyset$, then we replace Φ by $\Phi \land (v_1 \lor \neg v_1)$). Clearly enough, $F(\Phi)$ can be computed in time polynomial in $|\Phi|$. Now, Φ is valid iff $ECWA(v_1 \lor \neg v_1, \langle \emptyset, \emptyset, Var(\Phi)\rangle) \models \Phi$ holds. Indeed, it is well-known that the closure $ECWA(v_1 \lor \neg v_1, \langle \emptyset, \emptyset, Var(\Phi)\rangle)$ is equivalent to the disjunction of all the $\langle \emptyset, \emptyset, Var(\Phi)\rangle$-minimal models of $v_1 \lor \neg v_1$ (see Corollary 4.7 from [23]). Since the DNF formula $v_1 \lor \neg v_1$ is valid, every $\langle \emptyset, \emptyset, Var(\Phi)\rangle$-minimal model MM of this formula is generated by extending an interpretation over $Var(\Phi)$, assuming that $[\![v]\!](MM) = false$ for every $v \in PS \setminus Var(\Phi)$. Thus, the closure $ECWA(v_1 \lor \neg v_1, \langle \emptyset, \emptyset, Var(\Phi)\rangle)$ is equivalent to the conjunction of all the literals $\neg v$ for $v \in PS \setminus Var(\Sigma)$. Accordingly, we have $ECWA(v_1 \lor \neg v_1, \langle \emptyset, \emptyset, Var(\Phi)\rangle) \models \Phi$ iff Φ is valid. This shows F as a polynomial reduction from TAUT to ECWA FORMULA INFERENCE from a DNF formula under the restriction $Q = \emptyset$. The coNP-hardness of this last problem is a consequence of the coNP-completeness of TAUT.

- CCWA CLAUSE INFERENCE (resp. ECWA CLAUSE INFERENCE) under the restriction $Q = \emptyset$. The membership to P is a direct consequence of the membership to P of CCWA CLAUSE INFERENCE (resp. ECWA CLAUSE INFERENCE) from a Horn cover formula under the restriction $Q = \emptyset$ (see Proposition 23).
- ECWA FORMULA INFERENCE under the restriction $Q = Z = \emptyset$. The membership to P is a direct consequence of the membership to P of ECWA FORMULA INFERENCE from a Horn cover formula under the restriction $Q = Z = \emptyset$ (see Proposition 23).

Proof (Proposition 21) The upper bounds for GCWA FORMULA INFERENCE, CCWA FORMULA INFERENCE and ECWA FORMULA INFERENCE are consequences of Theorem 1 from [11]. The coNP-hardness of CCWA FORMULA INFERENCE and ECWA FORMULA INFERENCE are direct consequences of the coNP-hardness of CCWA LITERAL INFERENCE and ECWA LITERAL INFERENCE (cf. Proposition 22). The coNP-hardness of GCWA FORMULA INFERENCE is a direct consequence of the coNP-hardness of CCWA FORMULA INFERENCE under the restriction $Q = Z = \emptyset$ (cf. Proposition 23) since for every formula Φ, we have $GCWA(\Sigma) \models \Phi$ iff $CCWA(\Sigma, \langle Var(\Sigma), \emptyset, \emptyset \rangle) \models \Phi$ (see [22]). Finally, the membership to P of EGCWA FORMULA INFERENCE is a direct consequence of the membership to P of ECWA FORMULA INFERENCE under the restriction $Q = Z = \emptyset$ (cf. Proposition 23) since for every formula Φ, we have $EGCWA(\Sigma) \models \Phi$ iff $ECWA(\Sigma, \langle Var(\Sigma), \emptyset, \emptyset \rangle) \models \Phi$ (see [23]).

Proof (Proposition 22) The upper bounds for CCWA CLAUSE INFERENCE, ECWA CLAUSE INFERENCE, CCWA LITERAL INFERENCE and ECWA LITERAL INFERENCE are direct consequences of the corresponding upper bounds for CCWA FORMULA INFERENCE and ECWA FORMULA INFERENCE. The coNP-hardness of CCWA CLAUSE INFERENCE and ECWA CLAUSE INFERENCE are direct consequences of the coNP-hardness of CCWA LITERAL INFERENCE and ECWA LITERAL INFERENCE. The coNP-hardness of CCWA LITERAL INFERENCE and ECWA LITERAL INFERENCE are direct consequences of the coNP-hardness of their restrictions $Z = \emptyset$ (cf. Proposition 23). The membership to P of EGCWA CLAUSE INFERENCE and EGCWA LITERAL INFERENCE are direct consequences of the membership to P of EGCWA FORMULA INFERENCE (cf. Proposition 21). Finally, the membership to P of GCWA CLAUSE INFERENCE and GCWA LITERAL INFERENCE are consequences of the membership to P of CCWA CLAUSE INFERENCE under the restriction $Q = \emptyset$ (cf. Proposition 23) since for every formula Φ, we have $GCWA(\Sigma) \models \Phi$ iff $CCWA(\Sigma, \langle Var(\Sigma), \emptyset, \emptyset \rangle) \models \Phi$ (see [22]).

Proof (Proposition 23)

- CCWA LITERAL INFERENCE and ECWA LITERAL INFERENCE under the restriction $Z = \emptyset$. The coNP-hardness is a direct consequence of the coNP-hardness of the corresponding restrictions CCWA LITERAL INFERENCE and

ECWA LITERAL INFERENCE from a DNF formula under the restriction $Z = \emptyset$ (cf. Proposition 19).

- CCWA FORMULA INFERENCE under the restriction $Q = Z = \emptyset$. The coNP-hardness is a direct consequence of the coNP-hardness of the corresponding restriction CCWA FORMULA INFERENCE from a DNF formula under the restriction $Q = Z = \emptyset$ (cf. Proposition 19).

- ECWA FORMULA INFERENCE under the restriction $Q = \emptyset$. The coNP-hardness is a direct consequence of the coNP-hardness of its restriction to DNF formulas (cf. Proposition 19).

- CCWA CLAUSE INFERENCE under the restriction $Q = \emptyset$. By definition, the CCWA-ffn formulas are among the literals p of L_P^+. Moreover, as a consequence of Theorems 3.6 and 3.9 from [22], for every $p \in L_P^+$, p is free for negation w.r.t. CCWA iff $ECWA(\Sigma, \langle P, Q, Z\rangle) \models \neg p$ holds. Since checking whether $ECWA(\Sigma, \langle P, Q, Z\rangle) \models \neg p$ holds can be achieved in polynomial time when Σ is a Horn cover formula and $Q = \emptyset$ (see just below), the closure of Σ w.r.t. CCWA and $\langle P, Q, Z\rangle$ can be computed in polynomial time as well in this situation. This closure is equivalent to $\Sigma \cup \{\neg p; p \in L_P^+$ and $ECWA(\Sigma, \langle P, Q, Z\rangle) \models \neg p\}$. When $\Sigma = \Phi_1 \vee \ldots \vee \Phi_n$ is a Horn cover formula, this is also the case for the closure $(\Phi_1 \wedge (\neg p_1 \wedge \ldots \wedge \neg p_k)) \vee \ldots \vee (\Phi_n \wedge (\neg p_1 \wedge \ldots \wedge \neg p_k))$, where $p_1, \ldots, p_k$ are the CCWA-ffn formulas. Hence, checking whether $CCWA(\Sigma, \langle P, Q, Z\rangle) \models \gamma$ holds can be achieved in time polynomial in $|\Sigma| + |\gamma|$ when Σ is a Horn cover formula and $Q = \emptyset$.

- ECWA CLAUSE INFERENCE under the restriction $Q = \emptyset$.

  1. We first consider the case where the clause γ under consideration does not contain any variable from Z. In this situation, Theorem 4.5 from [23] shows that $ECWA(\Sigma, \langle P, \emptyset, Z\rangle) \models \gamma$ iff $\neg\gamma$ is free for negation w.r.t. ECWA and $\langle P, \emptyset, Z\rangle$. Now, $\neg\gamma$ is free for negation w.r.t. ECWA and $\langle P, \emptyset, Z\rangle$ iff every implicate of $\Sigma \wedge \gamma$ containing only literals from L_P^+ is an implicate of Σ (containing only literals from L_P^+) [28]. Equivalently, $\neg\gamma$ is free for negation w.r.t. ECWA and $\langle P, \emptyset, Z\rangle$ iff the conjunction of all prime implicates of $\Sigma \wedge \gamma$ containing only literals from L_P^+ is a logical consequence of the conjunction of all prime implicates of Σ containing only literals from L_P^+. For every formula α, let us note by $PI_{L_P^+}(\alpha)$ the (conjunctively-interpreted) set of prime implicates π of α s.t. every π contains only literals from L_P^+. When $\Sigma = \Phi_1 \vee \ldots \vee \Phi_n$ is a Horn cover formula:

     - For every clause γ not containing a variable from Z, a DNF formula equivalent to $PI_{L_P^+}(\Sigma \wedge \gamma)$ can be computed in time polynomial in $|\Sigma| + |\gamma|$. Moreover, every term in this DNF formula is positive when $\Sigma \wedge \gamma$ is satisfiable. Let us explain how to compute such a positive DNF formula. Since $\Sigma \wedge \gamma$ is equivalent to $(\Phi_1 \wedge \gamma) \vee \ldots \vee (\Phi_n \wedge \gamma)$, we have $PI_{L_P^+}(\Sigma \wedge \gamma) \equiv PI_{L_P^+}((\Phi_1 \wedge \gamma) \vee \ldots \vee (\Phi_n \wedge \gamma))$. Accordingly, when $\gamma = l_1 \vee \ldots \vee l_m$, we have $PI_{L_P^+}(\Sigma \wedge \gamma) \equiv PI_{L_P^+}((\Phi_1 \wedge l_1) \vee \ldots \vee (\Phi_1 \wedge l_m) \vee (\Phi_n \wedge l_1) \vee \ldots \vee (\Phi_n \wedge l_m))$. As a direct consequence of Proposition 3.14

from [36] (relating the prime implicates of a disjunctive formula to the prime implicates of its disjunct), we have $PI_{L_P^+}(\Sigma \wedge \gamma) \equiv PI_{L_P^+}(\Phi_1 \wedge l_1) \vee \ldots \vee PI_{L_P^+}(\Phi_1 \wedge l_m) \vee PI_{L_P^+}(\Phi_n \wedge l_1) \vee \ldots \vee PI_{L_P^+}(\Phi_n \wedge l_m))$. This formula contains $n{*}m$ disjuncts (with $n \leq |\Sigma|$ and $m \leq |\gamma|$); once every unsatisfiable disjunct occurring in it has been removed, the resulting formula actually is the positive DNF formula we are looking for. To prove it, it remains to show that every $PI_{L_P^+}(\Phi_i \wedge l_j)$ ($i \in 1..n$ and $j \in 1..m$) is equivalent to the empty clause or a conjunction of literals of L_P^+ and can be computed in time polynomial in $|\Phi_i|$. The proof relies on the following lemma:

Lemma 29 Let Φ be a Horn CNF formula. If Φ is satisfiable, every positive prime implicate of it is a (positive) literal. Otherwise, the unique positive prime implicate of Φ is the empty clause.

Proof (lemma) Clearly, Φ is not satisfiable iff the empty clause is the unique prime implicate of it. In the remaining case, assume that there exists a positive prime implicate γ of Φ containing at least two different literals. As a consequence of Lee's theorem of completeness in consequence finding for resolution [31], there exists a resolution proof of γ from Φ (see [36] for more details). Since the resolvent of two Horn clauses is a Horn clause and since γ is not a Horn clause, such a proof cannot exist.

Now, all the $\Phi_i \wedge l_j$ ($i \in 1..n$ and $j \in 1..m$) that are considered are Horn CNF formulas, and the prime implicates containing only literals from L_P^+ are positive prime implicates. Every $PI_{L_P^+}(\Phi_i \wedge l_j)$ ($i \in 1..n$ and $j \in 1..m$) can be computed thanks to the following algorithm:

Input: `a Horn CNF formula` Ψ.
Output: `a term equivalent to` $PI_{L_P^+}(\Psi)$
`if` Ψ `is satisfiable, the empty clause (`$\Box$`) otherwise.`
Begin
`(1) If` Ψ `is not satisfiable then return(`$\Box$`) fi.`
`(2) Set` α `to the empty set of literals.`
`(3) For every` $p \in L_P^+$ `do`
`(4)     If` $\Psi \models p$ `then add` p `to` α `fi.`
`(5) od.`
`(6) return(`α`).`
End

With $\Psi = \Phi_i \wedge l_j$ as input, this algorithm requires at most $|P| + 1$ deduction checks from Ψ. Since every such test can be achieved in polynomial time, the algorithm runs in polynomial time as well. Clearly enough, every $PI_{L_P^+}(\Phi_i \wedge l_j)$ ($i \in 1..n$ and $j \in 1..m$) is equivalent to the empty clause or a conjunction of literals of L_P^+. Accordingly, the formula $PI_{L_P^+}(\Phi_1 \wedge l_1) \vee \ldots \vee PI_{L_P^+}(\Phi_1 \wedge l_m) \vee PI_{L_P^+}(\Phi_n \wedge l_1) \vee \ldots \vee PI_{L_P^+}(\Phi_n \wedge l_m))$ is equivalent to a positive DNF formula, once every $PI_{L_P^+}(\Phi_i \wedge l_j)$ equivalent to the empty clause has been removed from the disjunction. The unique pathological exception is the case where every $PI_{L_P^+}(\Phi_i \wedge l_j)$ should be removed. In this situation, $PI_{L_P^+}(\Sigma \wedge \gamma)$ is unsatisfiable and $\neg\gamma$ is free for negation w.r.t. ECWA and $\langle P, \emptyset, Z\rangle$ iff Σ is unsatisfiable as well (which can be tested in time polynomial in $|\Sigma|$).

– Thanks to the algorithm above, a positive DNF formula equivalent to $PI_{L_P^+}(\Sigma)$ can be computed in time polynomial in $|\Sigma|$. Now, checking logical entailment between positive DNF formulas $PI_{L_P^+}(\Sigma \wedge \gamma)$ and $PI_{L_P^+}(\Sigma)$ can be achieved in time polynomial in the input size. Indeed, the prime implicants of a positive DNF formula are among its terms (no consensus between positive terms is possible). Hence, $PI_{L_P^+}(\Sigma) \models PI_{L_P^+}(\Sigma \wedge \gamma)$ holds iff for every term δ_Σ of $PI_{L_P^+}(\Sigma)$, there exists a term $\delta_{\Sigma\wedge\gamma}$ of $PI_{L_P^+}(\Sigma \wedge \gamma)$ s.t. $\delta_\Sigma \models \delta_{\Sigma\wedge\gamma}$ holds. Clearly enough, this test can be achieved in time polynomial in the size of both terms. Thus, it is possible to determine whether $PI_{L_P^+}(\Sigma) \models PI_{L_P^+}(\Sigma \wedge \gamma)$ holds in time polynomial in $|\Sigma| + |\gamma|$.

2. The case where the clause γ under consideration may contain some literals of Z remains to be considered. In this situation, we take advantage of the following lemma:

Lemma 30 Let Σ, Φ be formulas from $PROP_{PS}$ and $\langle P, Q, Z\rangle$ a partition of $Var(\Sigma)$. We have $ECWA(\Sigma, \langle P, Q, Z\rangle) \models \Phi$ iff $\Sigma \models \Phi$ holds or $ECWA(\Sigma, \langle P, Q, Z\rangle) \models \neg PI_{L_P^+ \cup L_Q}(\Sigma \wedge \neg\Phi)$ holds.

Proof (lemma) Theorem 2.6 from [43] shows that $ECWA(\Sigma, \langle P, Q, Z\rangle) \models \Phi$ iff $\Sigma \models \Phi$ holds or there exists a formula Ψ from $PROP_{PS}$ s.t.:

(a) Ψ does not contain any literal from $L_P^- \cup L_Z$, and
(b) $\Sigma \wedge \neg\Phi \models \Psi$, and
(c) $ECWA(\Sigma, \langle P, Q, Z\rangle) \models \neg\Psi$.

Clearly enough, the formula $\Psi = PI_{L_P^+ \cup L_Q}(\Sigma \wedge \neg\Phi)$ always satisfies points (1) and (2) above. Thus, when $ECWA(\Sigma, \langle P, Q, Z\rangle) \models \neg PI_{L_P^+ \cup L_Q}(\Sigma \wedge \neg\Phi)$ holds, we have $ECWA(\Sigma, \langle P, Q, Z\rangle) \models \Phi$. Conversely, let us assume that there exists a formula Ψ satisfying (1), (2) and (3). Let $\gamma_1 \wedge \ldots \wedge \gamma_n$ a CNF of Ψ. As a consequence of (1), every γ_i ($i \in 1..n$) contains only literals from $L_P^+ \cup L_Q$, and $\Sigma \wedge \neg\Phi \models \gamma_i$ holds. Hence, for every γ_i ($i \in 1..n$), there exists a prime implicate π_i from $PI_{L_P^+ \cup L_Q}(\Sigma \wedge \neg\Phi)$ s.t. $\pi_i \models \gamma_i$ holds. Hence, every γ_i ($i \in 1..n$) is a logical consequence of $PI_{L_P^+ \cup L_Q}(\Sigma \wedge \neg\Phi)$. Equivalently, $PI_{L_P^+ \cup L_Q}(\Sigma \wedge \neg\Phi) \models \Psi$ holds, i.e., $\neg\Psi \models \neg PI_{L_P^+ \cup L_Q}(\Sigma \wedge \neg\Phi)$ holds. Subsequently, when (3) is satisfied, $ECWA(\Sigma, \langle P, Q, Z\rangle) \models \neg PI_{L_P^+ \cup L_Q}(\Sigma \wedge \neg\Phi)$ holds as well.

This lemma shows how determining whether $ECWA(\Sigma, \langle P, \emptyset, Z\rangle) \models \gamma$ holds can be achieved in polynomial time when Σ is a Horn cover formula and γ is a clause. The point is that:

(a) Since Σ is tractable, checking whether $\Sigma \models \gamma$ holds can be done in time polynomial in $|\Sigma| + |\gamma|$. If $\Sigma \models \gamma$ holds, then $ECWA(\Sigma, \langle P, \emptyset, Z\rangle) \models \gamma$ holds as well.

(b) In the remaining case, we can compute a CNF formula $\gamma_1 \wedge \ldots \wedge \gamma_n$ of $\neg PI_{L_P^+}(\Sigma \wedge \neg\gamma)$ in time polynomial in $|\Sigma| + |\gamma|$. Indeed, since $\Sigma \wedge \neg\gamma$ is a Horn CNF formula, a DNF equivalent to $PI_{L_P^+}(\Sigma \wedge \neg\gamma)$ can be computed in polynomial time (see above). When Σ is the disjunction of n Horn CNF formulas, no more than n clauses γ_i are generated in this way. Moreover, every γ_i $(i \in 1..n)$ contains only literals from L_P^- (hence, $|\gamma_i| \leq |P|$). Since $ECWA(\Sigma, \langle P, \emptyset, Z\rangle) \models \neg PI_{L_P^+}(\Sigma \wedge \neg\gamma)$ holds iff $ECWA(\Sigma, \langle P, \emptyset, Z\rangle) \models \gamma_i$ holds for every γ_i $(i \in 1..n)$, checking whether $ECWA(\Sigma, \langle P, \emptyset, Z\rangle) \models \neg PI_{L_P^+}(\Sigma \wedge \neg\gamma)$ holds once $\gamma_1 \wedge \ldots \wedge \gamma_n$ has been computed requires at most n tests of the type $ECWA(\Sigma, \langle P, \emptyset, Z\rangle) \models \gamma_i$, where γ_i is a clause s.t. $Var(\gamma_i) \cap Z = \emptyset$. Since such tests can be computed in polynomial time (cf. above), $ECWA(\Sigma, \langle P, \emptyset, Z\rangle) \models \neg PI_{L_P^+}(\Sigma \wedge \neg\gamma)$ can be tested in polynomial time as well, and this completes the proof.

- ECWA FORMULA INFERENCE under the restriction $Q = Z = \emptyset$. It is well-known that $ECWA(\Sigma, \langle P, Q, Z\rangle) \models \Phi$ holds iff every $\langle P, Q, Z\rangle$-minimal model of Σ is a model of Φ (see e.g., Corollary 4.7 from [23]). Now, when $Q = Z = \emptyset$, the notion of $\langle P, Q, Z\rangle$-minimal model is equivalent to the notion of minimal model. The membership to P of ECWA FORMULA INFERENCE under the restriction $Q = Z = \emptyset$ relies on the fact that the disjunction of all the minimal models of Σ (restricted to $Var(\Sigma) \cup Var(\Phi)$) can be computed in time polynomial in $|\Sigma| + |\Phi|$. Indeed:

  - Let $\Sigma = \Phi_1 \vee \ldots \vee \Phi_n$ be a Horn cover formula. For every Φ_i $(i \in 1..n)$, let $M(\Phi_i)$ be the set of all models of Φ_i. The set $MinModels(\Sigma)$ containing all the minimal models of Σ is given by $MinModels(\Sigma) = min(M(\Phi_1 \vee \ldots \vee \Phi_n), \leq) = min(M(\Phi_1) \cup \ldots \cup M(\Phi_n), \leq) = min(min(M(\Phi_1), \leq) \cup \ldots \cup min(M(\Phi_n), \leq), \leq) = min(MinModels(\Phi_1) \cup \ldots \cup MinModels(\Phi_n), \leq)$.
  - It is well-known that every Horn CNF formula Φ_i has a unique minimal model MM_i when it is satisfiable. Thus, every $MinModels(\Phi_i)$ above is the empty set or a singleton. Whatever the case, $MinModels(\Phi_i)$ can be computed in polynomial time. Thus, if Φ_i is unsatisfiable (which can be tested in polynomial time), then $MinModels(\Phi_i)$ is the empty set. Otherwise, for every variable v from $Var(\Sigma) \cup Var(\Phi)$, we check (in polynomial time) whether $\Phi_i \models v$ holds; if so, we have $[\![v]\!](MM_i) = true$, else $[\![v]\!](MM_i) = false$. In this way, for every satisfiable Φ_i of Σ, we can compute in polynomial time a conjunction of literals encoding its minimal model MM_i (restricted to $Var(\Sigma) \cup Var(\Phi)$). Since the number of such models is bounded by n, and since checking whether or not $MM_i \leq MM_j$ holds can be achieved in time polynomial in $|Var(\Sigma) \cup Var(\Phi)|$, the set $MinModels(\Sigma)$ can be computed in time polynomial in $|\Sigma| + |\Phi|$. Then, in order to determine whether $ECWA(\Sigma, \langle P, Q, Z\rangle) \models \Phi$ holds, it is sufficient

to test if every MM from $MinModels(\Sigma)$ is s.t. $[\![\Phi]\!](MM) = true$, which can be done in time polynomial in the input size.

Proof (Proposition 25) The upper bounds for GCWA FORMULA INFERENCE, EGCWA FORMULA INFERENCE, CCWA FORMULA INFERENCE and ECWA FORMULA INFERENCE are consequences of Theorem 1 from [11]. They trivially hold as well for the restrictions GCWA CLAUSE INFERENCE, EGCWA CLAUSE INFERENCE, CCWA CLAUSE INFERENCE, ECWA CLAUSE INFERENCE, GCWA LITERAL INFERENCE, EGCWA LITERAL INFERENCE, CCWA LITERAL INFERENCE, and ECWA LITERAL INFERENCE. The coNP-hardness of GCWA FORMULA INFERENCE, EGCWA FORMULA INFERENCE, CCWA FORMULA INFERENCE, ECWA FORMULA INFERENCE, GCWA CLAUSE INFERENCE, EGCWA CLAUSE INFERENCE, CCWA CLAUSE INFERENCE, and ECWA CLAUSE INFERENCE are direct consequences of the coNP-hardness of GCWA LITERAL INFERENCE, EGCWA LITERAL INFERENCE, CCWA LITERAL INFERENCE, and ECWA LITERAL INFERENCE. Finally, the coNP-hardness of GCWA LITERAL INFERENCE, EGCWA LITERAL INFERENCE, CCWA LITERAL INFERENCE, and ECWA LITERAL INFERENCE are direct consequences of the coNP-hardness of their restrictions given in Propositions 26 and 27.

Proof (Proposition 26) Let us first recall that every satisfiable Krom formula is a renamable Horn cover formula, and that every positive formula is satisfiable. Then:

- EGCWA CLAUSE INFERENCE and ECWA CLAUSE INFERENCE under the restriction where Σ contains only one positive Krom formula and $Q = Z = \emptyset$. The coNP-hardness is a direct consequence of the coNP-hardness of EGCWA CLAUSE INFERENCE and ECWA CLAUSE INFERENCE from a positive Krom formula and $Q = Z = \emptyset$. See Theorem 5 and Corollary 12 from [11].

- ECWA LITERAL INFERENCE under the restriction where Σ contains only one positive Krom formula and $Q = \emptyset$. The coNP-hardness is a direct consequence of the coNP-hardness of ECWA CLAUSE INFERENCE from a positive Krom formula and $Q = \emptyset$. See Theorem 16 from [11].

Proof (Proposition 27)

- GCWA LITERAL INFERENCE and EGCWA LITERAL INFERENCE under the restriction where Σ contains only satisfiable Krom formulas. The coNP-hardness is a direct consequence of the coNP-hardness of CCWA LITERAL INFERENCE and ECWA LITERAL INFERENCE under the restriction where Σ contains only satisfiable Krom formulas and $Q = Z = \emptyset$ (cf. the point just below) since for every literal l, we have $GCWA(\Sigma) \models l$ iff $CCWA(\Sigma, \langle Var(\Sigma), \emptyset, \emptyset\rangle) \models l$ (cf. [22]) and $EGCWA(\Sigma) \models l$ iff $ECWA(\Sigma, \langle Var(\Sigma), \emptyset, \emptyset\rangle) \models l$ (cf. [23]).

- CCWA LITERAL INFERENCE and ECWA LITERAL INFERENCE under the restriction where Σ contains only Krom formulas and $Q = Z = \emptyset$. Let Σ be any positive Krom formula and $\neg l_1 \vee \ldots \vee \neg l_n$ be any clause from $PROP_{PS}$. Let F be the mapping s.t. $F(\langle \Sigma, \langle Var(\Sigma), \emptyset, \emptyset\rangle \neg l_1 \vee \ldots \vee \neg l_n\rangle) = \langle (\Sigma \wedge (\neg l_1 \vee new)) \vee \ldots \vee (\Sigma \wedge (\neg l_n \vee new)), \langle Var(\Sigma), \emptyset, \emptyset\rangle, \neg new\rangle$, where $new \in PS \setminus Var(\Sigma)$. Clearly enough, $F(\langle \Sigma, \langle Var(\Sigma), \emptyset, \emptyset\rangle, \neg l_1 \vee \ldots \vee \neg l_n\rangle)$ can be computed in time polynomial in the size of the input. F is a polynomial reduction from ECWA CLAUSE

INFERENCE from a positive Krom formula under the restriction $Q = Z = \emptyset$ to ECWA LITERAL INFERENCE from a renamable Horn cover formula containing only satisfiable Krom formulas, under the restriction $Q = Z = \emptyset$. Indeed, we have $ECWA(\Sigma, \langle Var(\Sigma), \emptyset, \emptyset\rangle) \models (\neg l_1 \vee \ ... \ \vee \neg l_n)$ iff $ECWA((\Sigma \wedge (\neg l_1 \vee new)) \vee \ ... \ \vee (\Sigma \wedge (\neg l_n \vee new)), \langle Var(\Sigma), \emptyset, \emptyset\rangle) \models \neg new$. Let us explain why this holds. From Theorem 4.5 from [23], we know that $ECWA(\Sigma, \langle Var(\Sigma), \emptyset, \emptyset\rangle) \models (\neg l_1 \vee \ ... \ \vee \neg l_n)$ holds iff $(l_1 \wedge \ ... \ \wedge l_n)$ is free for negation w.r.t. ECWA and $\langle Var(\Sigma), \emptyset, \emptyset\rangle$. Using the same theorem, we can also conclude that $ECWA((\Sigma \wedge (\neg l_1 \vee new)) \vee \ ... \ \vee (\Sigma \wedge (\neg l_n \vee new)), \langle Var(\Sigma), \emptyset, \emptyset\rangle) \models \neg new$ holds iff new is free for negation w.r.t. ECWA and $\langle Var(\Sigma), \emptyset, \emptyset\rangle$. Hence, we just have to prove that $PI_{L_P^+}(\Sigma) \models PI_{L_P^+}(\Sigma \wedge (\neg l_1 \vee \ ... \ \vee \neg l_n))$ holds iff $PI_{L_P^+}((\Sigma \wedge (\neg l_1 \vee new)) \vee \ ... \ \vee (\Sigma \wedge (\neg l_n \vee new))) \models PI_{L_P^+}((\Sigma \wedge (\neg l_1 \vee new)) \vee \ ... \ \vee (\Sigma \wedge (\neg l_n \vee new)) \wedge \neg new)$ holds.

First, as a direct consequence of Proposition 3.14 from [36] (relating the prime implicates of a disjunctive formula to the prime implicates of its disjunct), we have $PI_{L_P^+}((\Sigma \wedge (\neg l_1 \vee new) \wedge \neg new) \vee \ ... \ \vee (\Sigma \wedge (\neg l_n \vee new) \wedge \neg new)) \equiv PI_{L_P^+}(\Sigma \wedge (\neg l_1 \vee new) \wedge \neg new) \vee \ ... \ \vee PI_{L_P^+}(\Sigma \wedge (\neg l_n \vee new) \wedge \neg new)$. For every l_i $(i \in 1..n)$, we have $\Sigma \wedge (\neg l_i \vee new) \wedge \neg new \equiv \Sigma \wedge \neg l_i \wedge \neg new$. Since $new \notin Var(\Sigma) \cup Var(l_i)$, we have $PI_{L_P^+}(\Sigma \wedge \neg l_i \wedge \neg new) \equiv PI_{L_P^+}(\Sigma \wedge \neg l_i)$. Hence, we have $PI_{L_P^+}((\Sigma \wedge (\neg l_1 \vee new) \wedge \neg new) \vee \ ... \ \vee (\Sigma \wedge (\neg l_n \vee new) \wedge \neg new)) \equiv PI_{L_P^+}(\Sigma \wedge \neg l_1) \vee \ ... \ \vee PI_{L_P^+}(\Sigma \wedge \neg l_n) \equiv PI_{L_P^+}((\Sigma \wedge \neg l_1) \vee \ ... \ \vee (\Sigma \wedge \neg l_n))$. Now, since $(\Sigma \wedge (\neg l_1 \vee new)) \vee \ ... \ \vee (\Sigma \wedge (\neg l_n \vee new)) \models \Sigma$ holds, we have $PI_{L_P^+}((\Sigma \wedge (\neg l_1 \vee new)) \vee \ ... \ \vee (\Sigma \wedge (\neg l_n \vee new))) \models PI_{L_P^+}(\Sigma)$. It remains to prove that for every $\pi \in PI_{L_P^+}(\Sigma \wedge (\neg l_1 \vee \ ... \ \vee \neg l_n))$, if $PI_{L_P^+}((\Sigma \wedge (\neg l_1 \vee new)) \vee \ ... \ \vee (\Sigma \wedge (\neg l_n \vee new))) \models \pi$ holds, then $PI_{L_P^+}(\Sigma) \models \pi$ holds as well. Clearly enough, if for every l_i $(i \in 1..n)$, $PI_{L_P^+}(\Sigma \wedge (\neg l_i \vee new)) \models \pi$ holds, then there exists $\pi' \in PI_{L_P^+}(\Sigma \wedge (\neg l_i \vee new))$ s.t. $\pi' \models \pi$ holds. Since $\pi \in PI_{L_P^+}(\Sigma \wedge (\neg l_1 \vee \ ... \ \vee \neg l_n))$, we have $new \notin Var(\pi)$. Subsequently, $new \notin Var(\pi')$. Since $new \notin Var(\Sigma)$, every prime implicate π'' of $PI_{L_P^+}(\Sigma \wedge (\neg l_i \vee new))$ that is not a prime implicate of $PI_{L_P^+}(\Sigma)$ is s.t. $new \in Var(\pi'')$. As a consequence, we have $\pi' \in PI_{L_P^+}(\Sigma)$. Hence, $PI_{L_P^+}(\Sigma) \models \pi$ holds.

The coNP-hardness of ECWA CLAUSE INFERENCE from a positive Krom formula under the restriction $Q = Z = \emptyset$ (Corollary 12 from [11]) is then sufficient to complete the proof. Now, as a consequence of Theorem 3.9 from [22], for every literal l from L_P, we have $ECWA(\Sigma, \langle Var(\Sigma), \emptyset, \emptyset\rangle) \models l$ holds iff $CCWA(\Sigma, \langle Var(\Sigma), \emptyset, \emptyset\rangle) \models l$. Accordingly, the polynomial reduction F above can also be viewed as a reduction from ECWA CLAUSE INFERENCE from a positive Krom formula under the restriction $Q = Z = \emptyset$ to CCWA LITERAL INFERENCE from a renamable Horn cover formula, containing satisfiable Krom formulas only, under the restriction $Q = Z = \emptyset$. This shows the coNP-hardness of this last problem.

Proof (Proposition 28) The key of the proof lies in the fact that the GCWA-ffn formulas (or CCWA-ffn formulas) are polynomially bounded in $|\Sigma|$. To be more precise, these formulas are among the positive literals l_1, ..., l_k (i.e., variables) occurring in Σ. Accordingly, the size of the closure $\Sigma \cup \{\neg l_1, ..., \neg l_k\}$ of Σ w.r.t. GCWA (or CCWA) is polynomial in the size of Σ. Once this closure has been computed (using e.g., the algorithm given in [12]), checking whether a clause γ is a logical consequence of it can be achieved in polynomial time when Σ is tractable. Indeed, $\Sigma \cup \{\neg l_1, ..., \neg l_k\} \models \gamma$ holds iff $\Sigma \models (l_1 \vee \; ... \; \vee l_k \vee \gamma)$ holds. Note that the computational effort spent for computing the closure is neglected here (in the theoretical framework of compilability, even non-recursive compilation functions are allowed).