

# On Stratified Belief Base Compilation

Sylvie Coste-Marquis<sup>a</sup> and Pierre Marquis<sup>b</sup>

<sup>a</sup> *CRIL-CNRS, IUT de Lens, Université d'Artois, rue de l'Université, SP 16, 62307 Lens cedex, FRANCE,*

E-mail: [coste@cril.univ-artois.fr](mailto:coste@cril.univ-artois.fr)

<sup>b</sup> *CRIL-CNRS, Université d'Artois, rue de l'Université, SP 16, 62307 Lens cedex, FRANCE,*

E-mail: [marquis@cril.univ-artois.fr](mailto:marquis@cril.univ-artois.fr)

In this paper, we investigate the extent to which knowledge compilation can be used to circumvent the complexity of skeptical inference from a stratified belief base (SBB). We first analyze the compilability of skeptical inference from an SBB, under various requirements concerning both the selection policy under consideration, the possibility to make the stratification vary at the on-line query answering stage and the expected complexity of inference from the compiled form. Not surprisingly, the results are mainly negative. However, since they concern the worst case situation only, they do not prevent a compilation-based approach from being practically useful for some families of instances. While many approaches to compile an SBB can be designed, we are primarily interested in those which take advantage of existing knowledge compilation techniques for classical inference. Specifically, we present a general framework for compiling SBBs into so-called  $\mathcal{C}$ -normal SBBs, where  $\mathcal{C}$  is any tractable class for clausal entailment which is the target class of a compilation function. Another major advantage of the proposed approach lies in the flexibility of the  $\mathcal{C}$ -normal belief bases obtained, which means that changing the stratification does not require to re-compile the SBB. For several families of compiled SBBs and several selection policies, the complexity of skeptical inference is identified. Some tractable restrictions are exhibited for each policy. Finally, some empirical results are presented.

**Keywords:** Reasoning under inconsistency, stratified belief base, knowledge compilation, computational complexity.

**AMS Subject classification:** 68T27, 68T30, 03B53

## 1. Introduction

Dealing with inconsistency is required in many situations in which pieces of information come from different, possibly conflicting sources, or when some

exceptions to knowledge must be handled. In order to prevent reasoning from trivialization, classical inference cannot be directly used from an inconsistent formula, or a conjunctively interpreted set of formulas.

To cope with this problem, we adhere in this paper to the *coherence-based approach to inconsistency handling*. Pieces of information are represented by *propositional stratified belief bases* (SBBs for short), i.e., finite sets of propositional formulas equipped with a total pre-order which represents the relative plausibility of the given beliefs.<sup>1</sup> Following [44,8,43], coherence-based nonmonotonic entailment can be viewed as a two-step process: first, the preferred consistent subbases of the given SBB  $B$  are characterized and then inference from  $B$  is defined as classical inference from some of the selected subbases.

Among the various formalisms pointed out so far to deal with inconsistency (see [31] for a survey), the coherence-based approach is quite popular. Its success can be explained by several factors: the fact that it is quite simple in essence, close to well-studied possibilistic logic [2,4], and that it encompasses other important frameworks, like supernormal default theories with priorities [8] and syntax-based belief revision [38,39], as specific cases.

Clearly enough, there are many ways to extend a given total pre-order over formulas into a preference relation over sets of beliefs. In this paper, four important subbases selection policies are considered, namely the *possibilistic* policy, the *linear order* policy, the *inclusion-preference* policy and the *lexicographic* policy [2,4]. Additionally, several entailment principles can be defined [43,4]; indeed, a formula can be considered as a (nonmonotonic) consequence of  $B$  whenever it is a logical consequence of (1) all preferred subbases of  $B$  (skeptical inference), or (2) at least one preferred subbase of  $B$  (credulous inference), or alternatively (3) when it can be credulously inferred from  $B$  but its negation cannot be (argumentative inference). These three entailment principles have their own motivations and features; among them, *skeptical inference* is the most rational principle [2,15], which means that it leads to inference relations that are preferential, quasi-rational or even rational [33] (depending on the selection policy under consideration). This explains why we focus on it in this paper.

Whatever the selection policy and the entailment principle among those sketched above, inference from an SBB is computationally expensive. Especially,

<sup>1</sup> Many techniques for deriving such plausibility information from a “flat” belief base exist (see [1] for a survey).

the decision problem associated with skeptical inference from SBBs typically is at the first level or even at the second level of the polynomial hierarchy [41,16]. Thus an important question is: How to circumvent the intractability of inference from an SBB in order to enlarge the set of instances which can be solved in practice?

In this paper, we propose to use *knowledge compilation* as a way to improve inference from an SBB when many queries are to be considered. The key idea of compilation is pre-processing the fixed part of the inference problem (the SBB under consideration). This SBB is turned into a compiled one during an off-line compilation phase and then the compiled SBB is used to answer on-line queries. Assuming that the SBB does not often change and that answering queries from the compiled SBB is computationally easier than answering them from the original SBB, the compilation time can be balanced over a sufficient number of queries. Several knowledge compilation techniques for improving classical inference have been proposed so far (see [10] for a survey). When compiled knowledge bases are considered and queries are CNF formulas, the complexity of classical inference falls from coNP-complete down to P. While none of the knowledge compilation techniques that have been proposed so far can ensure the objective of improving classical inference to be reached in the worst case (because the size of the compiled form can be exponentially larger than the size of the original knowledge base), experiments reported in [46,6,20] have shown such approaches valuable in many practical situations.

In the following, we investigate the extent to which knowledge compilation can be used to circumvent the complexity of skeptical inference from an SBB. We first analyze the compilability of skeptical inference from an SBB, under various requirements concerning both the selection policy under consideration, the possibility to make the stratification vary at the on-line query answering stage and the expected complexity of inference from the compiled form. Not surprisingly, the results are mainly negative. However, since they concern the worst case situation only, they do not prevent a compilation-based approach from being practically useful for some families of instances.

While many approaches to compile an SBB can be designed, we are primarily interested in those which take advantage of existing knowledge compilation techniques for classical inference. Obviously, it is not possible to compile a belief base as a knowledge base in the general case, just because a belief base can be inconsistent and equivalence-preserving knowledge compilation functions would lead to trivialization in this situation. Accordingly, for any equivalence-preserving

compilation function which maps propositional formulas into a tractable class  $\mathcal{C}$  for clausal entailment, we show how to compile any SBB  $B$  into a so-called  $\mathcal{C}$ -normal base while preserving the set of its skeptical consequences (w.r.t. the four subbases selection policies evoked above) over the vocabulary of  $B$ . The proposed transformation has two main advantages. First, it can take advantage of any equivalence-preserving knowledge compilation technique for clausal entailment; This is an important point from the practical side since the size of the compiled base may heavily vary depending on the choice of the compilation technique. Second, it does not require the SBB to be re-compiled whenever the given stratification of beliefs changes. This is particularly helpful when some further pieces of evidence lead to question the plausibility of some pieces of belief, or when the preferential information encoded by the stratification change with queries.

Clearly enough, our compilation-based approach can prove helpful to improve reasoning only if the complexity of inference from a compiled SBB is lower than the complexity of inference from the original SBB. That is why it is important to identify the complexity pattern of inference from a  $\mathcal{C}$ -normal SBB. We achieve it, focusing on four major knowledge compilation functions found in the literature. Interestingly, for each subbase selection policy under consideration, we show that a judicious choice of the compilation function has the ability to render tractable skeptical inference from a compiled SBB.

The rest of this paper is organized as follows. Some formal preliminaries are given in Section 2, and some background about SBBs and knowledge compilation in Section 3. The compilability of skeptical inference from SBBs is investigated in Section 4. Our approach to compile SBBs is presented in Section 5, and the complexity of inference from  $\mathcal{C}$ -normal SBBs is reported in Section 6. Some empirical results are given in Section 7. Connections to some related works are discussed in Section 8, and Section 9 concludes the paper. Proofs are reported in the appendix.

## 2. Formal Preliminaries

In the following,  $PROP_{PS}$  denotes the propositional language built up from a finite set  $PS$  of symbols, the boolean constants *true* and *false*, and the connectives  $\neg$ ,  $\wedge$ ,  $\vee$ ,  $\Rightarrow$ ,  $\Leftrightarrow$ , and  $\oplus$  in the standard way.  $Var(\Sigma)$  denotes the set of propositional variables occurring in  $\Sigma$ .

The size of a formula  $\Sigma$  from  $PROP_{PS}$ , noted  $|\Sigma|$ , is the number of occurrences of 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. A CNF formula is any finite conjunction of clauses, while a DNF formula is any finite disjunction of terms.

Formulas are interpreted in the classical way.  $\models$  denotes classical entailment and  $\equiv$  denotes logical equivalence. Every finite set  $\Sigma$  of formulas is interpreted conjunctively.  $card(\Sigma)$  denotes the cardinal of  $\Sigma$ . A Krom formula is a CNF formula in which every clause contains at most two literals. A formula is Horn CNF iff it is a CNF formula in which every clause contains at most one positive literal. A renamable Horn CNF formula  $\Sigma$  is a CNF formula which can be turned into a Horn CNF formula by substituting in a uniform way in  $\Sigma$  some literals of  $L_{Var(\Sigma)}$  by their negation.

The following classes of formulas (which are target classes for some existing compilation functions) are considered in the paper:

- The *Blake* class is the set of formulas given in prime implicants normal form, i.e., for every CNF formula  $\Sigma$  from the Blake class and every clause  $\gamma$ , we have  $\Sigma \models \gamma$  iff there exists a clause  $\pi$  of  $\Sigma$  s.t.  $\pi \models \gamma$  holds,
- the *DNF* class is the set of DNF formulas,
- the *Horn cover* class is the set of finite disjunctions of Horn CNF formulas,
- the *renamable Horn cover* class is the set of finite disjunctions of renamable Horn CNF formulas.

Any formula  $\Sigma$  has a unique set of prime implicants (when each prime implicate is considered up to logical equivalence), hence a unique Blake equivalent (up to the ordering of the conjuncts) which can be viewed as the Blake normal form of  $\Sigma$ ; no similar canonicity result holds for any of the three other tractable classes under consideration.

We assume that the reader is familiar with some basic notions of computational complexity, especially the complexity classes  $P$ ,  $NP$ , and  $coNP$ , and the classes  $\Delta_k^p$ ,  $\Sigma_k^p$  and  $\Pi_k^p$  of the polynomial hierarchy  $PH$  (see [42] for details).  $\Delta_2^p[\mathcal{O}(\log n)]$  (sometimes called  $\Theta_2^p$ ) is the class of problems which can be de-

cided in polynomial time using only logarithmically many calls to an NP oracle. Let us recall that a decision problem is said to be at the  $k^{th}$  level of PH iff it belongs to  $\Delta_{k+1}^p$ , and is either  $\Sigma_k^p$ -hard or  $\Pi_k^p$ -hard. It is strongly believed that PH does not collapse (at any level), i.e., it is a truly infinite hierarchy (for every integer  $k$ ,  $\text{PH} \neq \Sigma_k^p$ ).

### 3. Stratified Belief Bases and Knowledge Compilation

Let us first give some background about stratified belief bases and knowledge compilation.

#### 3.1. Inference from Stratified Belief Bases

In this paper, we are concerned with (propositional) stratified belief bases:

**Definition 1** Stratified belief bases.

A *stratified belief base* (SBB)  $B$  is an ordered pair  $B = \langle \Delta, \leq \rangle$ , where  $\Delta = \{\phi_1, \dots, \phi_n\}$  is a finite set of formulas from  $PROP_{PS}$  and  $\leq$  is a total pre-order over  $\Delta$  (i.e., a reflexive and transitive relation over  $\Delta$  s.t. for every  $\phi_i, \phi_j$  belonging to  $\Delta$ , we have  $\phi_i \leq \phi_j$  or  $\phi_j \leq \phi_i$ ). Every subset  $S$  of  $\Delta$  is a *subbase* of  $B$ .

It is equivalent to define  $B$  as a finite sequence  $(\Delta_1, \dots, \Delta_k)$  of subbases of  $\Delta$ , where each  $\Delta_i$  ( $i \in 1 \dots k$ ) is the non-empty set which contains all the minimal elements of  $\Delta \setminus (\bigcup_{j=1}^{i-1} \Delta_j)$  w.r.t.  $\leq$ .<sup>2</sup> Clearly enough,  $\{\Delta_1, \dots, \Delta_k\}$  is a partition of  $\Delta$ . Each subset  $\Delta_i$  ( $i \in 1 \dots k$ ) is called a *stratum* of  $B$ , and  $i$  is the priority level of each formula of  $\Delta_i$ . Intuitively, the lower the priority level of a formula the higher its plausibility. Given a subbase  $S$  of  $B$ , we note  $S_i$  ( $i \in 1 \dots k$ ) the subset of  $S$  defined by  $S_i = S \cap \Delta_i$ . We also note  $Var(B) = \bigcup_{i=1}^k Var(\Delta_i)$ .

In the following, we assume that  $\Delta_1$  is a singleton, consisting of the (consistent) conjunction of all certain beliefs (i.e., the pieces of *knowledge*) of  $\Delta$ . Slightly abusing notations, we will identify  $\Delta_1$  with the (unique) formula it contains. Note that our assumption can be done without loss of generality since when no certain beliefs are available, it is sufficient to add *true* to  $\Delta$  as its unique minimal element w.r.t.  $\leq$  (and this will not lead to any significant computational overhead).

<sup>2</sup> By convention,  $\bigcup_{j=1}^0 \Delta_j = \emptyset$ .

Accordingly, an SBB  $B = (\Delta_1, \dots, \Delta_k)$  is a “standard” consistent knowledge base when  $k = 1$ , a supernormal default theory without prioritization when  $k = 2$ , and a supernormal default theory with priorities in the general case [8].

There are several ways to use the information given by an SBB corresponding to several epistemic attitudes. Following the analysis of previous works [44,8,43], inference from an SBB  $B$  is considered as a two-step process, consisting first in generating some preferred consistent subbases of  $B$  and then using classical inference from some of them. Many policies (or generation mechanisms) for the selection of preferred consistent subbases can be defined. In formal terms, a policy  $\mathcal{P}$  is a mapping that associates every SBB  $B$  with a set  $B_{\mathcal{P}}$  consisting of all the preferred consistent subbases of  $B$  w.r.t.  $\mathcal{P}$ . In the following, four policies are considered: The *possibilistic* policy, the *linear order* policy, the *inclusion-preference* policy, and the *lexicographic* policy.

**Definition 2** Selection policies.

Let  $B = (\Delta_1, \dots, \Delta_k)$  be an SBB.

- The set  $B_{\mathcal{PO}}$  of all the preferred subbases of  $B$  w.r.t. the possibilistic policy is the singleton  $\{\bigcup_{i=1}^{s-1} \Delta_i\}$ , where  $s$  is the smallest index ( $1 \leq s \leq k$ ) s.t.  $\bigcup_{i=1}^s \Delta_i$  is inconsistent.
- The set  $B_{\mathcal{LO}}$  of all the preferred subbases of  $B$  w.r.t. the linear order policy is the singleton  $\{\bigcup_{i=1}^k \Delta_i\}$ , where  $\Delta_i$  ( $i \in 1 \dots k$ ) is defined by  $\Delta_i = \Delta_i$  if  $\Delta_i \cup \bigcup_{j=1}^{i-1} \Delta_j$  is consistent,  $\emptyset$  otherwise.
- The set  $B_{\subseteq}$  of all maximal (w.r.t.  $\subseteq$ ) consistent subbases of  $B$  containing  $\Delta_1$  is  $\{S \subseteq \Delta \mid S \text{ is consistent, } \Delta_1 \subseteq S, \text{ and } \forall \phi \in \Delta \setminus S, S \cup \{\phi\} \text{ is inconsistent}\}$ . Two valuable subsets of it are:
  - \* The set  $B_{\mathcal{IP}}$  of all the preferred subbases of  $B$  w.r.t. the inclusion-preference policy is  $\{S \subseteq \Delta \mid S \text{ is consistent and } \forall S' \subseteq \Delta \text{ s.t. } S' \neq S \text{ and } S' \text{ is consistent, } \forall i \in 1 \dots k ((\forall j < i (S'_j = S_j)) \Rightarrow S_i \notin S'_i)\}$ .
  - \* The set  $B_{\mathcal{LE}}$  of all the preferred subbases of  $B$  w.r.t. the lexicographic policy is  $\{S \subseteq \Delta \mid S \text{ is consistent and } \forall S' \subseteq \Delta \text{ s.t. } S' \neq S \text{ and } S' \text{ is consistent, } \forall i \in 1 \dots k ((\forall j < i (card(S'_j) = card(S_j))) \Rightarrow card(S_i) \not\prec card(S'_i))\}$ .

**Example 3.** Let us ask Tweety the penguin for an illustration (for sure, Tweety is a bird; normally, every penguin is a bird and every feathered animal is a bird; we also have some evidence that birds fly, and that Tweety is feathered and does

not fly; Finally, it is only plausible that Tweety eats fish and that animals eating fish can swim). Formally, let  $B = (\Delta_1, \Delta_2, \Delta_3)$  with:

$$\Delta_1 = \{\text{penguin}(\text{Tweety})\},$$

$$\Delta_2 = \{\text{penguin}(\text{Tweety}) \Rightarrow \text{bird}(\text{Tweety}), \text{feathered}(\text{Tweety}) \Rightarrow \text{bird}(\text{Tweety}), \\ \text{bird}(\text{Tweety}) \Rightarrow \text{fly}(\text{Tweety}), \text{feathered}(\text{Tweety}) \wedge \neg \text{fly}(\text{Tweety})\},$$

$$\Delta_3 = \{\text{eat\_fish}(\text{Tweety}), \text{eat\_fish}(\text{Tweety}) \Rightarrow \text{swim}(\text{Tweety})\}$$

Using the policies previously defined, we get:

- $B_{\mathcal{PO}} = \{\{\text{penguin}(\text{Tweety})\}\}$
- $B_{\mathcal{LO}} = \{\{\text{penguin}(\text{Tweety}), \text{eat\_fish}(\text{Tweety}), \text{eat\_fish}(\text{Tweety}) \Rightarrow \text{swim}(\text{Tweety})\}\}$
- $B_{\mathcal{IP}} = \{\{\text{penguin}(\text{Tweety}), \text{penguin}(\text{Tweety}) \Rightarrow \text{bird}(\text{Tweety}), \\ \text{feathered}(\text{Tweety}) \Rightarrow \text{bird}(\text{Tweety}), \text{bird}(\text{Tweety}) \Rightarrow \text{fly}(\text{Tweety}), \\ \text{eat\_fish}(\text{Tweety}), \text{eat\_fish}(\text{Tweety}) \Rightarrow \text{swim}(\text{Tweety})\}, \\ \{\text{penguin}(\text{Tweety}), \text{penguin}(\text{Tweety}) \Rightarrow \text{bird}(\text{Tweety}), \\ \text{feathered}(\text{Tweety}) \Rightarrow \text{bird}(\text{Tweety}), \\ \text{feathered}(\text{Tweety}) \wedge \neg \text{fly}(\text{Tweety}), \\ \text{eat\_fish}(\text{Tweety}), \text{eat\_fish}(\text{Tweety}) \Rightarrow \text{swim}(\text{Tweety})\}, \\ \{\text{penguin}(\text{Tweety}), \text{bird}(\text{Tweety}) \Rightarrow \text{fly}(\text{Tweety}), \\ \text{feathered}(\text{Tweety}) \wedge \neg \text{fly}(\text{Tweety}), \\ \text{eat\_fish}(\text{Tweety}), \text{eat\_fish}(\text{Tweety}) \Rightarrow \text{swim}(\text{Tweety})\}\}$
- $B_{\mathcal{LE}} = \{\{\text{penguin}(\text{Tweety}), \text{penguin}(\text{Tweety}) \Rightarrow \text{bird}(\text{Tweety}), \\ \text{feathered}(\text{Tweety}) \Rightarrow \text{bird}(\text{Tweety}), \text{bird}(\text{Tweety}) \Rightarrow \text{fly}(\text{Tweety}), \\ \text{eat\_fish}(\text{Tweety}), \text{eat\_fish}(\text{Tweety}) \Rightarrow \text{swim}(\text{Tweety})\}, \\ \{\text{penguin}(\text{Tweety}), \text{penguin}(\text{Tweety}) \Rightarrow \text{bird}(\text{Tweety}), \\ \text{feathered}(\text{Tweety}) \Rightarrow \text{bird}(\text{Tweety}), \\ \text{feathered}(\text{Tweety}) \wedge \neg \text{fly}(\text{Tweety}), \\ \text{eat\_fish}(\text{Tweety}), \text{eat\_fish}(\text{Tweety}) \Rightarrow \text{swim}(\text{Tweety})\}\}$

All preferred subbases  $S$  of  $B$  (w.r.t. any of the above policies) are (by construction) consistent sets. Moreover, since  $\Delta_1$  is assumed consistent, we always have  $\Delta_1 \subseteq S$ . Unlike  $B_{\mathcal{PO}}$  and  $B_{\mathcal{LO}}$ , every element  $S$  of  $B_{\mathcal{IP}}$  (or  $B_{\mathcal{LE}}$ ) always is a maximal (w.r.t.  $\subseteq$ ) consistent subbase of  $B$ . To be more precise, we have

$$B_{\mathcal{LE}} \subseteq B_{\mathcal{IP}} \subseteq B_{\subseteq}.$$

While every stratum  $\Delta_i$  with  $i > 1$  plays the same role w.r.t.  $B_{\subseteq}$ , this is not the case when  $B_{\mathcal{IP}}$  (resp.  $B_{\mathcal{LE}}$ ) is considered; indeed, among the elements of  $B_{\subseteq}$ ,



those containing as many priority formulas as possible, w.r.t.  $\subseteq$  (resp. cardinality) are selected when the inclusion-preference (resp. the lexicographic) policy is considered. Given  $B_{\subseteq}$ , both  $B_{\mathcal{IP}}$  and  $B_{\mathcal{LE}}$  can be computed in polynomial time (just filter out the preferred elements w.r.t. the chosen selection policy). Notwithstanding  $\Delta_1$ , the elements of  $B_{\mathcal{IP}}$  correspond to the so-called preferred subtheories of [8].

All these selection policies have their own motivations and lead to inference relations that are more or less satisfying from a logical point of view and from the computational complexity point of view (see [2,4,15]).

At that stage, it must be noted that there is a close correspondance between inference from an SBB and inference from a revised belief base in the “syntax-based” belief revision setting [41]. Indeed, we can define the set of consequences of any SBB  $(\Delta_1, \dots, \Delta_k)$  in the latter setting as the set of consequences of the SBB  $(\Delta_2, \dots, \Delta_k)$  revised by  $\Delta_1$  in the former setting, and the converse also holds. We heavily take advantage of such a correspondance in our complexity proofs. Thus, the cut base revision scheme corresponds exactly to the possibilistic policy, the linear-base revision scheme to the linear order policy, the prioritized base revision scheme to the inclusion-preference policy, and the lexicographic base revision scheme to the lexicographic policy [41].

Now, given a selection policy, several entailment principles can be considered, especially credulous inference, argumentative inference, skeptical inference. Among them, skeptical inference leads to inference relations which are at least preferential ones, i.e., relations satisfying all the rationality postulates given in [33]: Reflexivity, Left Logical Equivalence, Right Weakening, Cut, Cautious Monotonicity and Or (see [2,15]); this is why we focus on skeptical inference in the following.

**Definition 4** Skeptical inference.

Let  $B = (\Delta_1, \dots, \Delta_k)$  be an SBB,  $\mathcal{P}$  a policy for the generation of preferred subbases, and  $\Psi$  a formula from  $PROP_{PS}$ .  $\Psi$  is a (*skeptical*) *consequence* of  $B$  w.r.t.  $\mathcal{P}$ , noted  $B \sim_{\mathcal{P}}^{\Psi} \Psi$ , iff  $\forall S \in B_{\mathcal{P}}, S \models \Psi$ .

Unfortunately, whatever the selection policy among  $\mathcal{PO}, \mathcal{LO}, \mathcal{IP}, \mathcal{LE}$ , skeptical inference is not tractable (under the standard assumptions of complexity theory).

**Problem 5** FORMULA  $\vdash_{\mathcal{V}}^{\mathcal{P}}$ .

Let  $\vdash_{\mathcal{V}}^{\mathcal{P}}$  be any inference relation from  $\{\vdash_{\mathcal{V}}^{\mathcal{PO}}, \vdash_{\mathcal{V}}^{\mathcal{LO}}, \vdash_{\mathcal{V}}^{\mathcal{IP}}, \vdash_{\mathcal{V}}^{\mathcal{LE}}\}$ . FORMULA  $\vdash_{\mathcal{V}}^{\mathcal{P}}$  is the following decision problem:

- **Input:** An SBB  $B = (\Delta_1, \dots, \Delta_k)$  and a formula  $\Psi$  from  $PROP_{PS}$ .
- **Query:** Does  $B \vdash_{\mathcal{V}}^{\mathcal{P}} \Psi$  hold?

CLAUSE  $\vdash_{\mathcal{V}}^{\mathcal{P}}$  (resp. LITERAL  $\vdash_{\mathcal{V}}^{\mathcal{P}}$ ) is the restriction of FORMULA  $\vdash_{\mathcal{V}}^{\mathcal{P}}$  to the case  $\Psi$  is required to be a CNF formula (resp. a term).<sup>3</sup>

The following complexity results can be found in the literature. Note that these previous complexity results typically concern the FORMULA  $\vdash_{\mathcal{V}}^{\mathcal{P}}$  problem. Nevertheless, it is easy to modify the corresponding hardness proofs to show that the complexity lower bounds are also valid for both the corresponding CLAUSE  $\vdash_{\mathcal{V}}^{\mathcal{P}}$  and LITERAL  $\vdash_{\mathcal{V}}^{\mathcal{P}}$  problems. Proofs are given in the appendix.

**Proposition 6.** [Skeptical inference from SBBs]

The complexity of FORMULA  $\vdash_{\mathcal{V}}^{\mathcal{P}}$  from an SBB and of its restrictions to clause and literal inference for  $\mathcal{P} \in \{\mathcal{PO}, \mathcal{LO}, \mathcal{IP}, \mathcal{LE}\}$  is as reported in Table 1.

Table 1  
Complexity of skeptical inference from SBBs (general case).

$\mathcal{P}$	FORMULA / CLAUSE / LITERAL $\vdash_{\mathcal{V}}^{\mathcal{P}}$
$\mathcal{PO}$	$\Delta_2^p[\mathcal{O}(\log n)]$ -complete [41]
$\mathcal{LO}$	$\Delta_2^p$ -complete [40]
$\mathcal{IP}$	$\Pi_2^p$ -complete [38]
$\mathcal{LE}$	$\Delta_2^p$ -complete [14]

### 3.2. Knowledge Compilation

*Knowledge compilation* (see [10] for a survey) gathers several techniques which prove helpful in the objective of improving inference, in particular clause entailment [47], but also diagnosis, planning, belief revision, etc [35]. In this

<sup>3</sup> Whenever  $\vdash_{\mathcal{V}}^{\mathcal{P}}$  satisfies the rationality postulate And, the complexity of determining whether a CNF (resp. a term)  $\psi$  is a consequence of an SBB  $B$  amounts to the complexity of the same problem with  $\psi$  a clause (resp. a literal).

section, we specifically focus on knowledge compilation techniques for improving classical inference, i.e., for making the following decision problem computationally easier:

**Problem 7**  $\text{FORMULA} \models$ .

$\text{FORMULA} \models$  is the following decision problem:

- **Input:** Two formulas  $\Sigma$  and  $\Psi$  from  $PROP_{PS}$ .
- **Query:** Does  $\Sigma \models \Psi$  hold?

$\text{CLAUSE} \models$  (resp.  $\text{LITERAL} \models$ ) is the restriction of  $\text{FORMULA} \models$  to the case when  $\Psi$  is required to be a CNF formula (resp. a term).

Existing researches about knowledge compilation can be split into two categories. The first category gathers theoretical works about *compilability*, which indicates whether the objective can be expected to be reached in the worst case by focusing on the size of the compiled form (see e.g., [12,35]) which should remain polynomial in the input size. Indeed, if the size of the compiled form is exponentially larger than the size of the original KB  $\Sigma$ , significant computational improvements are hard to be expected. Accordingly, some decision problems are compilable, while others are probably not compilable (i.e., not compilable under the standard assumptions of the complexity theory). Thus,  $\text{LITERAL} \models$  is compilable while both  $\text{FORMULA} \models$  and  $\text{CLAUSE} \models$  are (probably) not compilable.<sup>4</sup>

The second category contains works that are much more oriented towards the design of compilation algorithms for clause entailment and their empirical evaluations. Their purpose is to point out some compilation functions, i.e., functions which map any propositional formula (or any conjunctively-interpreted finite set of propositional formulas) into a “more tractable”, yet equivalent formula. Thus, among others, [45,25,36,46,6,19] present equivalence-preserving knowledge compilation methods for clause entailment. In this context, “equivalent” means either “logically equivalent” (the usual notion) or more generally “query-equivalent”, which means that logical equivalence is only guaranteed over the propositional language built up from the set of variables occurring in the original knowledge

<sup>4</sup>The existence of an equivalence-preserving compilation function  $COMP$  s.t. it is guaranteed that there exists a polynomial  $p$  s. t. for every propositional CNF formula  $\Sigma$ ,  $\text{FORMULA} \models$  (resp.  $\text{CLAUSE} \models$ ) from  $COMP(\Sigma)$  is in  $\mathbf{P}$  and  $|COMP(\Sigma)|$  is  $\leq p(|\Sigma|)$  would make  $\mathbf{P} = \mathbf{NP}$  (just because determining whether a formula is valid is  $\text{coNP}$ -complete) (resp. the polynomial hierarchy to collapse at the second level (see [47,10] for more details)).

base (indeed, new variables can be incorporated into the compiled form if this proves useful). All these methods aim at computing a formula  $COMP(\Sigma)$  “equivalent” to  $\Sigma$  s.t.  $CLAUSE \models$  belongs to  $\mathcal{P}$  when restricted to inputs of the form  $\langle COMP(\Sigma), \psi \rangle$ . Stated otherwise, compiling  $\Sigma$  consists in turning it into an “equivalent” formula belonging to a  $\models$ -tractable class  $\mathcal{C}$  for clause entailment (i.e., a class of formulas for which there exists a polytime algorithm for deciding the clausal entailment problem). Formally:

**Definition 8** Compilation function.

An *equivalence-preserving knowledge compilation function*  $COMP$  is a mapping from  $PROP_{PS}$  to  $PROP_{PS}$  which associates any  $\Sigma \in PROP_{PS}$  with a formula  $COMP(\Sigma)$  s.t.

- $COMP(\Sigma)$  is query-equivalent to  $\Sigma$ , and
- the range of  $COMP$  is a  $\models$ -tractable class  $\mathcal{C}$ .

Considering KBs  $\Sigma$  from  $\models$ -tractable class is helpful for the  $CLAUSE \models$  problem, since determining whether a clause is entailed by such a KB  $\Sigma$  can be achieved in polynomial time, while the problem is  $\text{coNP}$ -complete when  $\Sigma$  is unconstrained. In the rest of this paper, the following tractable classes of formulas which are target classes for some existing compilation functions are considered:

- The *Blake* class,
- the *DNF* class,
- the *Horn cover* class,
- the *renamable Horn cover* class.

**Example 9.** Let  $\Sigma = (a \vee b \vee c) \wedge (\neg c \vee d) \wedge (\neg a \vee \neg b \vee \neg c)$ .  $COMP$ -compiled forms of  $\Sigma$  are:

- $Blake(\Sigma) = (a \vee b \vee c) \wedge (\neg c \vee d) \wedge (\neg a \vee \neg b \vee \neg c) \wedge (a \vee b \vee d)$ .
- $DNF(\Sigma) = (a \wedge \neg b \wedge c \wedge d) \vee (a \wedge \neg c) \vee (\neg a \wedge c \wedge d) \vee (\neg a \wedge b \wedge \neg c)$ .
- $Horn\_cover(\Sigma) = (a \wedge ((\neg b \vee \neg c) \wedge (\neg c \vee d))) \vee (\neg a \wedge c \wedge d) \vee (\neg a \wedge b \wedge \neg c)$ .
- $r\_Horn\_cover(\Sigma) = (a \wedge ((\neg b \vee \neg c) \wedge (\neg c \vee d))) \vee (\neg a \wedge ((b \vee c) \wedge (\neg c \vee d)))$ .

Each of these  $COMP$ -compiled forms of  $\Sigma$  is logically equivalent to it.

The Blake class (resp. the DNF class) is the target class of the compilation function described in [45] (resp. in [46]). The Horn cover class and the renamable Horn cover class are target classes for the tractable covers compilation functions given in [6].

Among these four target classes, the DNF class, the Horn cover class and the renamable Horn cover class can be compared w.r.t. succinctness [29]: Every DNF formula is a Horn cover formula, and every Horn cover formula is a renamable Horn cover formula. Accordingly, the latter may offer much more compact representations than the former, hence it is often a better performer when clausal entailment is concerned (see the experimental results reported in [6]). However, as we will see in the following, this is not necessarily the case when more sophisticated forms of inference are considered (and this is why all three classes are considered).

Of course, the four compilation functions  $COMP$  discussed above are subject to the theoretical limitation evoked previously: In the worst case, the size of the compiled form  $COMP(\Sigma)$  is exponential in the size of  $\Sigma$ . Nevertheless, there is some empirical evidence that some of these approaches can prove computationally valuable for many instances of the  $CLAUSE \models$  problem (see the experimental results given in [46,6,20]).

#### 4. On the Compilability of Skeptical Inference from SBBs

Intuitively, compiling an SBB interpreted under some selection policy consists in turning it off-line into a formula (or more generally into a data structure), called a  $\mathcal{C}$ -normal SBB, from which on-line query answering can be done *more efficiently* than from the original SBB. As evoked in the introductory section, the compilation time can be balanced over sufficiently many queries, provided that the size of the compiled SBB remains *small enough*. In this case, we roughly say that inference from the SBB is compilable (which just means that compiling the SBB is computationally helpful).

Defining a formal notion of compilability that matches this intuitive definition but at the problem level instead of the instance level requires to make precise what (i) “more efficiently” and (ii) “small enough” mean [12]. In the following, the (skeptical) inference problem from an SBB interpreted under a given selection policy is said to be *compilable* whenever (i) the complexity of the inference problem from such a data structure is *below* the complexity of the corresponding

problem in the polynomial hierarchy (the counterpart of “more efficiently”), and (ii) any SBB can be associated with a query-equivalent *polyspace* data structure (the counterpart of “small enough”).

Since the complexity of the inference problem from an SBB lies between  $\Delta_2^p[\mathcal{O}(\log n)]$  and  $\Pi_2^p$  (depending on the selection policy, cf. Proposition 6), it is natural to wonder whether any SBB interpreted under a given selection policy can be compiled into a (1) query-equivalent polyspace formula, or even into a (2) query-equivalent polyspace formula belonging to a  $\models$ -tractable class. These two cases correspond respectively to two complexity classes located below  $\Delta_2^p[\mathcal{O}(\log n)]$  in the polynomial hierarchy. Indeed, in case (1), on-line query answering would be “only” in  $\text{coNP}$ . In the more demanding case (2), on-line clausal query answering would even be tractable, i.e., in  $\text{P}$ . Accordingly, these two cases can be interpreted as degrees of compilability for the inference problem: if (1) is feasible, the inference problem will be said to be *weakly compilable* while if (2) is satisfied in addition, it will be said to be *strongly compilable*.

Now, an important feature of any inference problem in a compilability perspective is the separation between its fixed part (subject to preprocessing) and its variable part (which is known at the on-line stage only). The basic case (a) consists in considering the SBB  $(\Delta, \leq)$  as fixed and the queries as varying. A more demanding case (b) consists in considering the set  $\Delta$  of formulas forming the SBB as fixed, while both the partitioning of formulas into strata induced by  $\leq$  and the queries are varying. This is particularly helpful in the situation when the preferential information encoded by the stratification change with queries. These two cases can be viewed as degrees of flexibility for the inference problem: in case (a), the inference problem is said to be *unflexible* and it is said to be *flexible* if it satisfies the requirement (b).

We have obtained the following compilability results:

**Proposition 10.** [Compilability results]

The results reported in Table 2 hold. **U** (resp. **F**) means that the inference problem is unflexible (resp. flexible). **W** (resp. **S**) means that the expected compilability strength is weak (resp. strong). “no $\downarrow$  2” means “no” unless the polynomial hierarchy collapses at the second level, while “no $\downarrow$  3” means “no” unless the polynomial hierarchy collapses at the third level.

Many of these results are easy consequences of results reported in [12,13,11]

Table 2  
 Compilability of skeptical inference from SBBs.

$\mathcal{P}$	<b>U - W</b>	<b>F - W</b>	<b>U - S</b>	<b>F - S</b>
$\mathcal{PO}$	yes	no↓ 3	no↓ 2	no↓ 2
$\mathcal{LO}$	yes	no↓ 3	no↓ 2	no↓ 2
$\mathcal{IP}$	no↓ 3	no↓ 3	no↓ 2	no↓ 2
$\mathcal{LE}$	yes	no↓ 3	no↓ 2	no↓ 2

(see Appendix). As it could be expected, both the flexibility dimension, the compilability strength and the selection policy under consideration may have a great influence on the compilability issue. Thus, the  $\mathcal{IP}$  selection policy leads to inference problems which are at least as hard as the other policies w.r.t. compilability (a similar behaviour can be observed when considering the direct, “uncompiled” inference problem (cf. Table 1)). The only compilable problems (under the usual assumptions of the complexity theory) are obtained for the  $\mathcal{PO}$ ,  $\mathcal{LO}$  and  $\mathcal{LE}$  policies when the preferential information are available at the compilation time and tractability for answering on-line clausal queries is not required.

## 5. Compiling Stratified Belief Bases

From the previous compilability results, we cannot conclude that knowledge compilation is out of value in the objective of improving inference from the practical side. Indeed, such results concern the worst case situation only, so they do not prevent some compilation-based approaches from being valuable at least for some instances. This will become more salient at Section 7, where some empirical results are reported.

There exist many ways to compile an SBB (we will present the main ones in Section 8). In this paper, we are interested in techniques taking advantage of existing compilation functions for classical inference. Because a piece of belief is not always a piece of knowledge, the conjunction of all pieces of belief of an SBB is not always consistent. This explains why we cannot take advantage of equivalence-preserving knowledge compilation techniques to compile belief bases in a direct way when they are inconsistent. Indeed, such techniques are dedicated to “flat” bases, i.e., no preferential information is part of the input. Accordingly, compiling a “flat”, inconsistent knowledge base results in a “flat”, inconsistent

knowledge base from which classical entailment is trivial. For instance, an inconsistent knowledge base has only one prime implicate, the empty clause, from which no significant inference can be done.

In order to avoid such a trivialization, our approach consists in compiling only the conjunction of pieces of knowledge of  $\Delta$  (i.e., the stratum  $\Delta_1$ ), provided that each other piece of belief  $\phi_{i,j}$  has been first given a name under the form of a new literal  $holds_{i,j}$ , i.e.,  $\phi_{i,j}$  is replaced by  $holds_{i,j}$  and  $(holds_{i,j} \Rightarrow \phi_{i,j})$  is added in  $\Delta_1$ .<sup>5</sup> Formally, we define a normal SBB as an SBB s.t. the first stratum belongs to a  $\models$ -tractable class and the remaining beliefs are represented by literals:

**Definition 11**  $\mathcal{C}$ -normal SBBs.

An SBB  $B = (\Delta_1, \dots, \Delta_k)$  is  $\mathcal{C}$ -normal iff  $\Delta_1$  belongs to the  $\models$ -tractable class  $\mathcal{C}$  and  $\bigcup_{i=2}^k \Delta_i$  is a consistent subset of  $L_{PS}$ . We also say that a  $\mathcal{C}$ -normal SBB  $B = (\Delta_1, \dots, \Delta_k)$  is *COMP-normal* when  $\Delta_1$  belongs to the target class of the equivalence-preserving compilation function *COMP*.

Let us now show how any equivalence-preserving knowledge compilation function can be used to compile an SBB.

**Definition 12** *COMP*-compilations.

Let  $B = (\Delta_1, \dots, \Delta_k)$  be an SBB (with  $\Delta = \bigcup_{i=1}^k \Delta_i$ ) and let *COMP* be any equivalence-preserving compilation function (for clause entailment). Without loss of generality, let us assume that every stratum  $\Delta_i$  ( $i \in 1 \dots k$ ) of  $B$  is totally ordered (w.r.t. any fixed order) and let us note  $\phi_{i,j}$  the  $j^{th}$  formula of  $\Delta_i$  w.r.t. this order. The SBB

$$(\chi_1, \dots, \chi_k)$$

where  $\chi_i = \{holds_{i,1}, \dots, holds_{i,card(\Delta_i)}\}$  for  $i \in 2 \dots k$ , each  $holds_{i,j} \in L_{PS} \setminus L_{Var(\Delta)}$ , and

$$\chi_1 = \{COMP(\Delta_1 \wedge (\bigcup_{i=2}^k \{ \bigwedge_{j=1}^{card(\Delta_i)} (holds_{i,j} \Rightarrow \phi_{i,j}) \}))\}$$

<sup>5</sup> Adding  $(holds_{i,j} \Leftrightarrow \phi_{i,j})$  to  $\Delta_1$  would work as well but it does not allow a linear time translation of the resulting stratum into a logically equivalent CNF formula, and many existing compilation functions require the input to be CNF; nevertheless a linear time translation into a query-equivalent CNF exists.



is the *COMP*-*compilation* of  $B$ . Slightly abusing notations, we note it  $COMP(B)$ .<sup>6</sup>

This transformation basically consists in giving a name (under the form of a new literal  $holds_{i,j}$ ) to each assumption of  $\Delta$  and in storing the correspondance assumption/name with the certain beliefs *before* compiling them for clause entailment. A similar transformation is given in [16]; it has also been used in the context of Dynamic Constraint Satisfaction Problems in order to enable a pre-determined subset of constraints to be added or removed in an efficient way (see e.g., [7]).

Obviously, every *COMP*-*compilation* of an SBB is a *COMP*-normal SBB.

**Example 13.** Let us now ask Omer the emu for an illustration of the transformation (for sure, Omer is an emu and every emu is a bird; normally, emus do not fly; with more exceptions, birds fly). Formally, let  $B = (\Delta_1, \Delta_2, \Delta_3)$  with:

$$\Delta_1 = \{emu(Omer) \wedge (emu(Omer) \Rightarrow bird(Omer))\},$$

$$\Delta_2 = \{emu(Omer) \Rightarrow \neg fly(Omer)\}, \text{ and}$$

$$\Delta_3 = \{bird(Omer) \Rightarrow fly(Omer)\}.$$

The stratification used here reflects the fact that most specific beliefs are preferred (exceptional emus are rarer than exceptional birds). On this simple example, all four selection policies give the same results (w.r.t. skeptical inference).

The *COMP*-*compilation* of  $B$  is  $COMP(B) = (\chi_1, \chi_2, \chi_3)$  where:

$$\begin{aligned} \chi_1 = \{ & COMP(emu(Omer) \wedge (emu(Omer) \Rightarrow bird(Omer))) \wedge \\ & (Emusdon'tfly(Omer) \Rightarrow (emu(Omer) \Rightarrow \neg fly(Omer))) \wedge \\ & (Birdsfly(Omer) \Rightarrow (bird(Omer) \Rightarrow fly(Omer))) \} \end{aligned}$$

$$\chi_2 = \{Emusdon'tfly(Omer)\},$$

$$\chi_3 = \{Birdsfly(Omer)\}.$$

Here,  $Emusdon'tfly(Omer)$  and  $Birdsfly(Omer)$  are the new literals used to name (uncertain) beliefs before compilation.

When *COMP* is a Blake compilation function (i.e., any function which computes the set of all prime implicates of the given propositional formula), we obtain:

$$Blake(B) = (\chi_1, \chi_2, \chi_3)$$

where:

$$\begin{aligned} \chi_1 = \{ & emu(Omer) \wedge bird(Omer) \wedge \\ & (\neg Emusdon'tfly(Omer) \vee \neg fly(Omer)) \wedge \end{aligned}$$

<sup>6</sup> The notation is compatible with the notation  $COMP(\Sigma)$  when  $B = (\Sigma)$  is a one-stratum SBB.

$$\begin{aligned}
& (\neg \text{Birdsfly}(\text{Omer}) \vee \text{fly}(\text{Omer})) \wedge \\
& (\neg \text{Emusdon'tfly}(\text{Omer}) \vee \neg \text{Birdfly}(\text{Omer})), \\
\chi_2 &= \{\text{Emusdon'tfly}(\text{Omer})\}, \\
\chi_3 &= \{\text{Birdsfly}(\text{Omer})\}.
\end{aligned}$$

Interestingly, for every SBB, there exists an “equivalent”  $\mathcal{C}$ -normal SBB, where equivalence is defined by:

**Definition 14** Equivalence of SBBs.

Let  $B = (\Delta_1, \dots, \Delta_k)$  and  $B' = (\Delta'_1, \dots, \Delta'_l)$  be two SBBs. Let  $V$  be a subset of  $PS$  and  $\mathcal{P}$  a selection policy.  $B$  and  $B'$  are *equivalent* on  $V$  w.r.t.  $\mathcal{P}$  iff there exists a bijection  $\beta$  from  $B_{\mathcal{P}}$  to  $B'_{\mathcal{P}}$  s.t. for every  $S \in B_{\mathcal{P}}$  and every formula  $\Psi$  from  $PROP_V$ ,  $S \models \Psi$  iff  $\beta(S) \models \Psi$ .

Indeed, our compilation approach turns the input SBB  $B$  into a query-equivalent one, i.e., into an SBB equivalent to  $B$  on  $Var(B)$ :

**Proposition 15** Equivalence preservation.

Let  $B = (\Delta_1, \dots, \Delta_k)$  be an SBB and let  $COMP$  be any equivalence-preserving compilation function (for clause entailment).  $COMP(B)$  is a  $COMP$ -normal SBB equivalent to  $B$  on  $Var(B)$  w.r.t.  $\mathcal{P} \in \{\mathcal{PO}, \mathcal{LO}, \mathcal{IP}, \mathcal{LE}\}$ .

**Example 16.** Whatever the selection policy may be (among the four ones considered in this paper),  $B$  has only one preferred subbase:  $\Delta_1 \cup \Delta_2$ .  $COMP(B)$  also has one preferred subbase:  $\chi_1 \cup \chi_2$ . Both subbases are equivalent on the set of variables occurring in  $B$ .

The motivation for our definition of  $\mathcal{C}$ -normal SBBs  $B$  relies on the fact that it is not sufficient that every formula of  $\Delta$  belongs to a  $\models$ -tractable class in order to improve  $\text{CLAUSE} \sim_{\mathcal{V}}^{\mathcal{P}}$  in the general case. Indeed, forming preferred subbases of  $B$  requires to check the consistency of conjunctions of such formulas and, usually, formulas from  $\models$ -tractable class do not mix well w.r.t. conjunction as far as computational complexity is concerned. For instance, determining whether a finite set of clauses containing only Horn CNF clauses and Krom clauses is consistent is NP-complete (see Problem 9.5.6 in [42]).<sup>7</sup> More specifically, tractable classes of formulas are typically not closed under conjunction (especially, that is

<sup>7</sup> To every instance  $\Sigma = \{\gamma_1, \dots, \gamma_n\}$  of the 3-SAT problem, let us associate in polynomial time

not the case for any of the four tractable classes considered in this paper), and the existence of a polytime algorithm that would turn the conjunction of two input formulas of a given tractable class into one equivalent formula from that class is hard to be expected.

Contrastingly, whatever the compilation function used to compile  $\Delta_1$  may be, the consistency of any subbase of a compiled SBB  $B$  which contains  $\Delta_1$  can be checked in polynomial time because every assumption from  $\bigcup_{i=2}^k \Delta_i$  is a literal. Thus, any equivalence-preserving compilation function can be used for compiling an SBB. Since many of the existing compilation functions have no comparable computational behaviours (each of them performs better than the others on some instances), this is valuable from the practical side.

Another important feature offered by our compilation framework for SBBs lies in the fact that no re-compilation is required whenever the stratification of the given pieces of belief changes (even in the case when the number of strata changes). Indeed, would the plausibility ordering  $\leq$  over  $\bigcup_{i=2}^k \Delta_i$  change, the only impact on any *COMP*-normal  $B$  from any update would be limited to the corresponding re-partitionning of  $\bigcup_{i=2}^k \Delta_i$ .

## 6. Complexity of Inference from $\mathcal{C}$ -Normal SBBs

The purpose of compiling an SBB is to enhance inference from it. This objective can be achieved only if (1) the size of the compiled SBB remains small enough, compared with the size of the original SBB, and (2) inference from the compilation is computationally easier than inference from the original SBB. Because classical entailment can be recovered as a specific case of any inference relation considered in this paper (just consider SBBs for which  $\Delta = \Delta_1$ ), the compilability limitations for both *FORMULA*  $\models$  and *CLAUSE*  $\models$  also apply for these more sophisticated forms of inference: it is not granted that the size of the compiled form of an SBB remains polynomial in the size of the original SBB, regardless of the compilation function. Let us stress that these limitations not only concern the compilation technique proposed in this paper, but any conceivable preprocessing of SBBs. Because some of these functions have empirically proved

the CNF formula  $\Sigma' = \{\gamma'_1, \dots, \gamma'_n\} \cup \{x_i \vee new\_x_i, \neg x_i \vee \neg new\_x_i \mid x_i \text{ is a variable occurring as a positive literal in } \Sigma\}$ , where each  $\gamma'_i$  is obtained by replacing every positive literal  $x_j$  in it by the negative literal  $\neg new\_x_j$ .  $\Sigma'$  contains only Horn and Krom clauses and is satisfiable iff  $\Sigma$  is satisfiable.

their computational value, we can nevertheless expect computational benefits for many instances. In this section, we show the extent to which (2) can be expected, depending on the inference relation under consideration, the nature of the query (formula, clause, literal) and the compilation function  $COMP$  used.

We have first identified the following complexity upper bounds, corresponding to the situation when no assumption is made about the nature of the tractable fragment to which  $\Delta_1$  belongs:

**Proposition 17** Skeptical inference from  $\mathcal{C}$ -normal SBBs.

The complexity of  $\text{FORMULA} \vdash_{\forall}^{\mathcal{P}}$  and of its restrictions to clause and literal inference for  $\mathcal{P} \in \{\mathcal{PO}, \mathcal{LO}, \mathcal{IP}, \mathcal{LE}\}$  from a normal SBB is as reported in Table 3.

Table 3  
Complexity of skeptical inference from  $\mathcal{C}$ -normal SBBs (upper bounds).

$\mathcal{P}$	FORMULA $\vdash_{\forall}^{\mathcal{P}}$	CLAUSE / LITERAL $\vdash_{\forall}^{\mathcal{P}}$
$\mathcal{PO}$	in coNP	in P
$\mathcal{LO}$	in coNP	in P
$\mathcal{IP}$	in coNP	in coNP
$\mathcal{LE}$	in $\Delta_2^P$	in $\Delta_2^P$

Proposition 17 shows that considering  $\mathcal{C}$ -normal SBBs can actually make inference computationally easier, even if this is not the case in every situation (just consider  $\vdash_{\forall}^{\mathcal{LE}}$ ).

Within Proposition 17, no assumption on the nature of the  $\mathcal{C}$ -normal SBB has been done. In order to obtain complexity lower bounds (hardness results) or tractability results for clausal inference w.r.t. the  $\mathcal{IP}$  policy and the  $\mathcal{LE}$  policy, restricted  $\mathcal{C}$ -normal SBBs must be considered. In the following, we focus on  $COMP$ -normal SBBs where  $COMP$  is a compilation function which maps any propositional formula into a Blake, DNF, Horn cover or renaming Horn cover formula.

**Proposition 18** Skeptical inference w.r.t.  $\mathcal{IP}$  from  $COMP$ -normal SBBs.

The complexity of  $\text{FORMULA} \vdash_{\forall}^{\mathcal{IP}}$  and of its restrictions to clause and literal inference from  $COMP$ -normal SBBs is as reported in Table 4.

Table 4

Complexity of skeptical inference w.r.t.  $\mathcal{IP}$  from  $COMP$ -normal SBBs.

$COMP$	FORMULA $\vdash_{\forall}^{\mathcal{IP}}$	CLAUSE / LITERAL $\vdash_{\forall}^{\mathcal{IP}}$
<b>Blake</b>	coNP-complete	coNP-complete
<b>DNF</b>	coNP-complete	in P
<b>Horn cover</b>	coNP-complete	coNP-complete
<b>renamable Horn cover</b>	coNP-complete	coNP-complete

**Proposition 19** Skeptical inference w.r.t.  $\mathcal{LE}$  from  $COMP$ -normal SBBs.

The complexity of FORMULA  $\vdash_{\forall}^{\mathcal{LE}}$  and of its restrictions to clause and literal inference from  $COMP$ -normal SBBs is as reported in Table 5.

Table 5

Complexity of skeptical inference w.r.t.  $\mathcal{LE}$  from  $COMP$ -normal SBBs.

$COMP$	FORMULA $\vdash_{\forall}^{\mathcal{LE}}$	CLAUSE / LITERAL $\vdash_{\forall}^{\mathcal{LE}}$
<b>Blake</b>	$\Delta_2^p$ -complete	$\Delta_2^p$ -complete
<b>DNF</b>	coNP-complete	in P
<b>Horn cover</b>	$\Delta_2^p$ -complete	$\Delta_2^p$ -complete
<b>renamable Horn cover</b>	$\Delta_2^p$ -complete	$\Delta_2^p$ -complete

Some of the hardness results are easy consequences of results reported in [16,41,18] (see Appendix for more details).

Tractability is only achieved for  $COMP$ -normal SBBs for which  $\Delta_1$  is a DNF formula and queries are restricted to CNF formulas. Intractability results w.r.t. both  $\vdash_{\forall}^{\mathcal{IP}}$  and  $\vdash_{\forall}^{\mathcal{LE}}$  still hold when  $\Delta_1$  is a consistent Krom formula (such formulas are renamable Horn CNF ones, and can be turned in polynomial time into Blake normal form, see e.g., [37]), or when  $\Delta_1$  is a Horn CNF formula. All the hardness results presented in Table 4 still hold in the specific case when the number  $k$  of strata under consideration satisfies  $k \geq 2$ . LITERAL  $\vdash_{\forall}^{\mathcal{LE}}$  remains coNP-hard in the restricted case when  $k \geq 2$ .

Interestingly, imposing some restrictions on the literals used to name assumptions enables us to derive tractable restrictions for both CLAUSE  $\vdash_{\forall}^{\mathcal{IP}}$  and

CLAUSE  $\vdash_{\forall}^{\mathcal{L}\mathcal{E}}$  from the *COMP*-compilation of an SBB where  $\Delta_1$  is a Horn cover formula. Indeed, we have:

**Proposition 20** Tractable restrictions.

CLAUSE  $\vdash_{\forall}^{\mathcal{I}\mathcal{P}}$  and CLAUSE  $\vdash_{\forall}^{\mathcal{L}\mathcal{E}}$  from a  $\mathcal{C}$ -normal SBB  $B = (\Delta_1, \dots, \Delta_k)$  where  $\Delta_1$  is a Horn cover formula and  $\bigcup_{i=2}^k \Delta_i$  contains only negative literals are in  $\mathcal{P}$ .

Actually, all the tractable cases listed above rely on the following lemma:

**Lemma 21.** Let  $B = (\Delta_1, \dots, \Delta_k)$  be an SBB with  $\Delta = \bigcup_{i=1}^k \Delta_i$ . We recall that  $B_{\subseteq} = (\Delta_1, \dots, \Delta_k)_{\subseteq}$  is the set of all maximal (w.r.t.  $\subseteq$ ) consistent subbase of  $B = \bigcup_{i=1}^k \Delta_i$  containing  $\Delta_1$ . We have:

1. If  $\Delta_1 = \{\alpha_1 \vee \dots \vee \alpha_n\}$  where each  $\alpha_i$  ( $i \in 1 \dots n$ ) is a formula from *PROPPS*,  $B_{\subseteq} = \max_{\subseteq}(\{\Delta_1 \cup (S \cap \Delta) \mid S \in \bigcup_{i=1}^n (\{\alpha_i\}, \bigcup_{j=2}^k \Delta_j)_{\subseteq}\})$ .
2. If  $\alpha$  is a term and  $\bigcup_{j=2}^k \Delta_j$  is a consistent set of literals, then  $(\{\alpha\}, \bigcup_{j=2}^k \Delta_j)_{\subseteq}$  is the singleton  $\{\{\alpha\} \cup \{\phi \in \bigcup_{j=2}^k \Delta_j \mid \alpha \not\models \neg\phi\}\}$ .
3. If  $\alpha$  is a Horn CNF formula and  $\bigcup_{j=2}^k \Delta_j$  contains only negative literals, then  $(\{\alpha\}, \bigcup_{j=2}^k \Delta_j)_{\subseteq}$  is the singleton  $\{\{\alpha\} \cup \{\phi \in \bigcup_{j=2}^k \Delta_j \mid \alpha \not\models \neg\phi\}\}$ .

Since the transformation reported in Definition 12 does not require any constraint on the literals used to name beliefs, negative literals can be used. Accordingly, it is possible to compile any SBB so as to make both CLAUSE  $\vdash_{\forall}^{\mathcal{I}\mathcal{P}}$  and CLAUSE  $\vdash_{\forall}^{\mathcal{L}\mathcal{E}}$  tractable from the compiled form. Of course, this is already achieved by only requiring  $\Delta_1$  to be a DNF formula. However, while every DNF formula is a Horn cover formula, the converse typically does not hold and the Horn cover fragment is strictly more compact than the DNF one as a representation formalism (some DNF formulas can be represented by Horn cover formulas the sizes of which are logarithmically smaller but the converse does not hold).<sup>8</sup>

**Example 22.**  $B$  can be turned into the following Horn cover compilation  $Horn\_cover(B) = (\chi_1, \chi_2, \chi_3)$ :

$$\chi_1 = \{(fly(Omer) \wedge emu(Omer) \wedge (emu(Omer) \Rightarrow bird(Omer))) \wedge$$

<sup>8</sup> For instance, the size of the smallest DNF formula equivalent to the Horn cover formula  $\bigwedge_{i=1}^m (\neg x_{2i} \vee \neg x_{2i+1})$  is  $\Omega(2^m)$ . Indeed, this formula has  $\Omega(2^m)$  essential prime implicants.

$$\begin{aligned}
& (emu(Omer) \Rightarrow Emusfly(Omer)) \vee \\
& (\neg fly(Omer) \wedge emu(Omer) \wedge \\
& (emu(Omer) \Rightarrow bird(Omer)) \wedge \\
& (bird(Omer) \Rightarrow Birdsdon'tfly(Omer)))
\end{aligned}$$

$$\chi_2 = \{\neg Emusfly(Omer)\},$$

$$\chi_3 = \{\neg Birdsdon'tfly(Omer)\}.$$

From this  $\mathcal{C}$ -normal SBB,  $Horn\_cover(B)_{\subseteq}$  can be derived in polynomial time as:

$$\{\chi_1 \cup \{\neg Birdsdon'tfly(Omer)\}, \chi_1 \cup \{\neg Emusfly(Omer)\}\}.$$

By construction, each of the two elements of  $Horn\_cover(B)_{\subseteq}$  is a Horn cover formula. Only the latter one is preferred w.r.t.  $\mathcal{IP}$  (or  $\mathcal{LE}$ ), enabling us to conclude the desired result (Omer doesn't fly).

## 7. Some Empirical Results

In order to assess the feasibility of the compilation approach from the practical side, we have computed DNF-compilations of SBBs. We have implemented Schrag's algorithm for computing prime implicants covers (i.e., DNFs in which each term is a prime implicant) of propositional CNF formulas [46]; this algorithm is based on the well-known Davis/Logemann/Loveland procedure for satisfiability testing [24].

### 7.1. The Empirical Framework

All the experimentations have been done on PC under Linux. The code has been written in C and compiled using `gcc` with the option `-O2`. It is available from the authors.

For the sake of comparison, we have performed tests using the same families of benchmarks as those reported in [16]. Thus, we have made experiments on a complete version of the famous penguin problem and two formalizations of a small real common-sense problem involving respectively 31 and 77 formulas, distributed in respectively 7 and 9 strata, as given in [16]. We have also performed tests on random 3-CNF contradictory instances, generated following the standard model [34]: Each clause is obtained by randomly picking up 3 variables among the available ones and negating each variable with probability 0.5. This model has been studied in depth by the SAT community; specifically, it has been observed that it

exhibits a phase transition phenomenon at a ratio of  $\frac{\#clauses}{\#variables} = 4.25$ . Instances generated using a lower (resp. higher) ratio are supposed to be satisfiable (resp. unsatisfiable) with high probability. Furthermore, instances generated using the above ratio of 4.25 are typically difficult for satisfiability testing. We have considered instances generated from 10, 20 and 30 variables with a ratio  $\frac{\#clauses}{\#variables}$  varying from 5 to 10 by increment of 1. Clauses are gathered conjunctively into formulas to form the original belief bases  $B$ . The number of resulting formulas varies from 2 to 20 percent of the total number of clauses by increment of 1. The number of strata varies from 10 to 100 percent<sup>9</sup> of the total number of formulas by increment of 10 percent.

For each SBB under consideration, we have applied the transformation given at Definition 12: Each formula is given a new name  $holds_{i,j}$ , which leads to increase the number of variables of the base, and then a DNF  $\chi_1$  of the conjunction of  $\Delta_1$  and all mappings formula/name is computed using Schrag’s algorithm.

Due to the number of parameters used, we cannot present here exhaustive results from our experimentations (additional results can be obtained from the authors on request). Instead, we just report in the following table and figures the number of terms and the number of literals in  $\chi_1$ . It should be obvious that for each SBB  $B$  under consideration, the number of literals in the DNF-compilation of  $B$  (as given in Definition 12) is bounded by the number of literals in  $\chi_1$  plus the number of new names  $holds_{i,j}$  (and this last number can typically be neglected). Hence, the number of literals in  $\chi_1$  represents in a faithful way the size of the corresponding DNF-compilation.

## 7.2. Experimental Results

Table 6 gives both the number of terms and the number of literals in  $\chi_1$ , for the three toy problems. The compilation time can be neglected for each of them.

Table 6  
Three toy problems.

Problem	#implicants	#literals
Penguin	4	22
31 formulas	6	117
77 formulas	6	129

<sup>9</sup> In this case, each stratum contains exactly one formula.





Figure 2. Results for 20 variables.

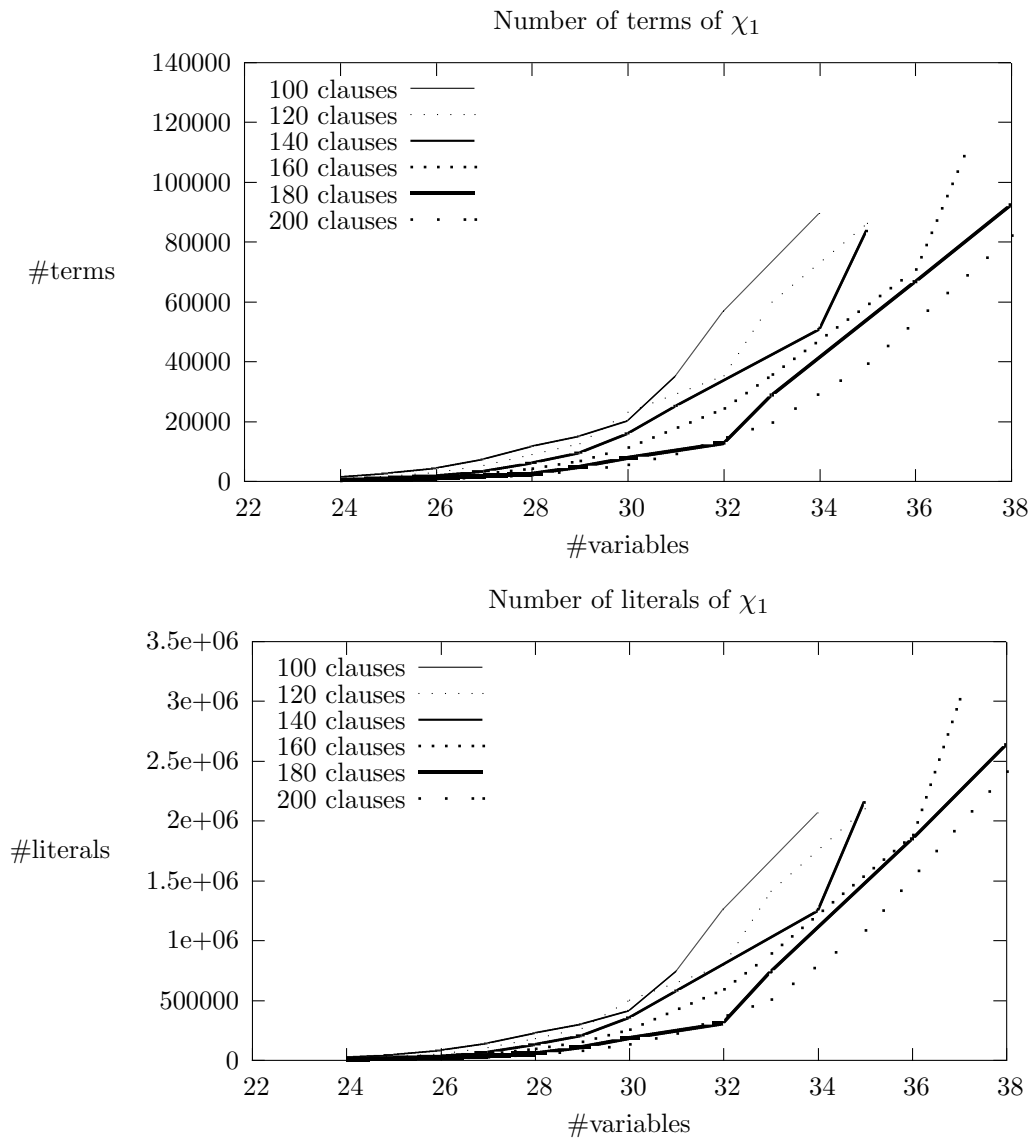
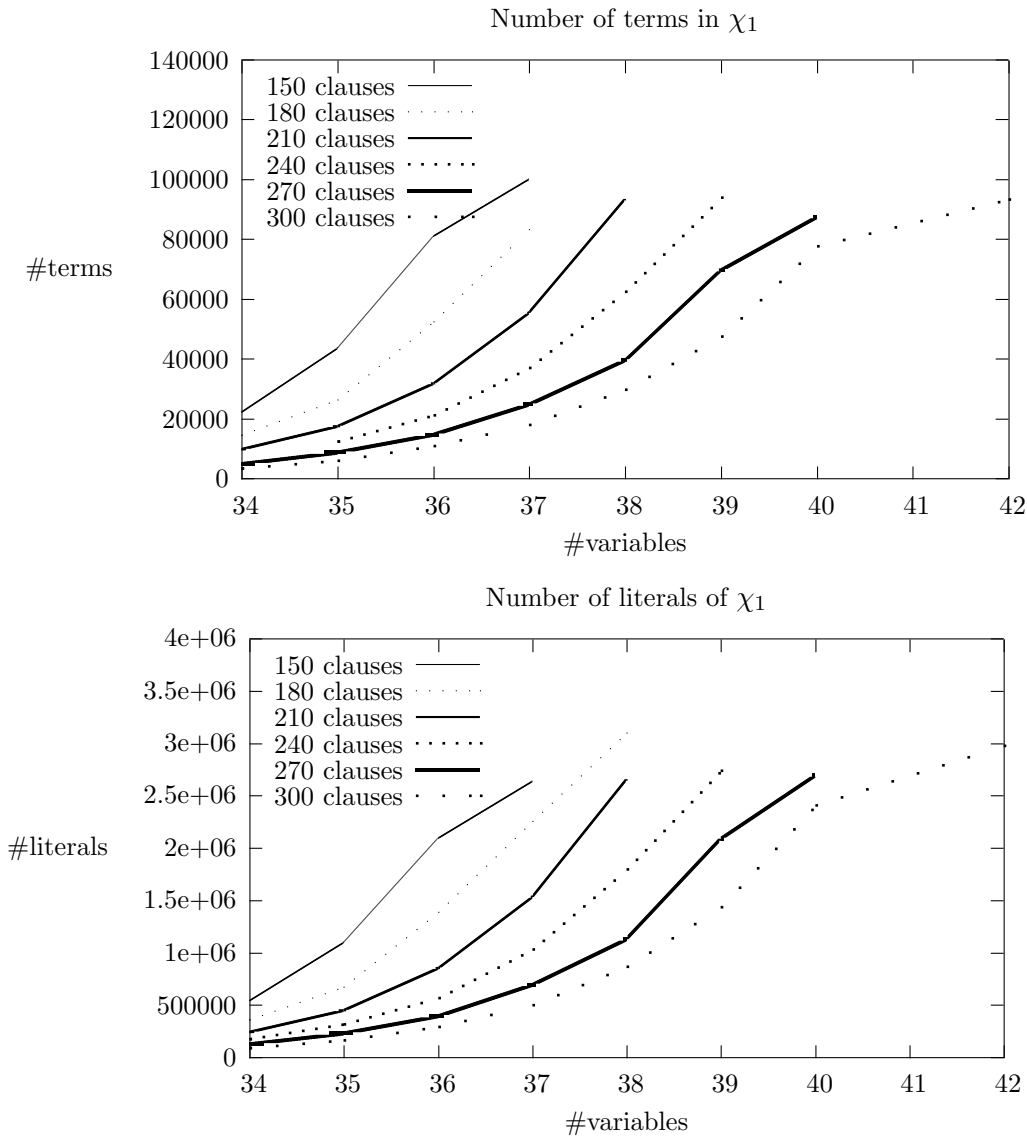


Figure 1 gives the the number of terms and the number of literals in  $\chi_1$  for the 3-CNF random instances generated using 10 variables. Each curve corresponds to a specific ratio, hence to a specific number of clauses. 50 instances per point on the curves have been generated with the same parameters. Each point is the mean value<sup>10</sup> obtained over the 50 instances. For each instance tested, the

<sup>10</sup> For the sake of readability, we do not present other statistical information on the curves (like

computation time required to compute  $\chi_1$  was lower than 0.1 s.

Figure 3. Results for 30 variables.



Similarly, Figure 2 (resp. 3) gives the number of terms and number of literal in  $\chi_1$  for 3-CNF random instances generated using 20 (resp. 30) variables. Each the min, max, standard deviation and median); actually, average seems to be a significant measurement here (especially, the standard deviation is typically small for each point).

point here is the mean value obtained over at most 50 instances, as we have imposed a upper limit of one hour CPU time per instance to compute  $\chi_1$ .

On Figure 3, the displayed points are the averages of 24 to 50 values. The points for which less than 24 values over 50 have been obtained (given the CPU time limit of one hour per instance) have been discarded.

### 7.3. Discussion

In the light of these experiments, a few conclusions can be drawn:

- The number of terms in  $\chi_1$  is relevant to the whole compilation process. On the one hand, the number of literals in  $\chi_1$  is obviously bounded by the number of variables in the instance after transformation times the number of terms in  $\chi_1$ . Empirically, we observe that the number of literals in  $\chi_1$  is proportional to the number of terms in it. On the other hand, we also observe that the time needed to compute  $\chi_1$  is proportional to the number of implicants in it (this has already been noticed in [46]).
- The more constrained the input instance, the less implicants in  $\chi_1$ . The number of formulas in the base also has a major impact on the size of the compilation since the transformation leads to increase the number of variables in the instance in a significant way, leading to a less constrained instance. Indeed, within the limit of one hour CPU time per base, it has not been possible to compute  $\chi_1$  for more than 50 percent of the bases generated from 20 variables and containing 200 clauses, provided that such clauses are gathered into at least 40 formulas. This empirical evidence coheres with the results reported in [6], showing that DNF-compilations are only suited to hardly constrained instances.

Similar experiments are reported in [16], using Reduced Ordered Binary Decision Diagrams (ROBDD) compilations of SBBs. ROBDD [9] can be viewed as a  $\models$ -tractable class consisting of some DAG-based propositional formulas (i.e., identical subformulas are shared). Looking at the experimental results given in [16] and related to ROBDD-compilations of SBBs (cf. Section 8), we can observe that DNF-compilations seem to perform better than ROBDD-ones on the toy problems (comparing the sizes of the compilations). However, in contrast to DNF-compilations, it seems that ROBDD-compilations can be computed in

a few minutes on random 3-CNF instances generated using 20 variables<sup>11</sup> with ratio 8, and one clause per formula. We can also observe that the size of ROBDD-compilations grows with the number of clauses in the instances, but it is unclear whether this is due to the fact that the original instances are more constrained or, on the contrary, to the fact that the transformed ones (with names added) are less constrained due to the new variables stemming from the transformation.

Contrariwise to [16], we do not report on the time needed to achieve clausal inference from DNF-compilations. Actually, the time complexity of this problem is linear in the number of terms of the DNF formula  $B_{DNF}$  which can be computed from  $DNF(B)$  thanks to the following algorithm.<sup>12</sup> Let  $\mathcal{P} \in \{\mathcal{IP}, \mathcal{LE}\}$ :

1. Compute  $B_{\mathcal{P}}$  in time polynomial in  $|DNF(B)|$  taking advantage of points 1. and 2. of Lemma 21.
2. Remove from  $\chi_1$  every term  $t$  for which there does not exist  $S \in B_{\mathcal{P}}$  s.t.  $t \wedge (S \setminus \{\chi_1\})$  is consistent.
3. Forget every name  $holds_{i,j}$  in the resulting DNF (i.e., remove every literal built up from such a variable from the terms where it occurs).
4. Minimize the resulting DNF (i.e., remove every term that entails another term).

It is obvious that this algorithm takes only time polynomial in the size of  $DNF(B)$ . Furthermore, for any clause  $\gamma$  built up from  $Var(B)$ , we have  $B \sim_{\mathcal{P}}^{\forall} \gamma$  iff  $B_{DNF} \models \gamma$ . Indeed, by definition of skeptical inference, we have  $B \sim_{\mathcal{P}}^{\forall} \gamma$  iff for each  $S \in B_{\mathcal{P}}$ ,  $S \models \gamma$ . Now, all preferred subbases  $S \in B_{\mathcal{P}}$  share by construction the same first stratum  $\chi_1$ , which can be factored. Hence,  $B \sim_{\mathcal{P}}^{\forall} \gamma$  iff  $\chi_1 \wedge \bigvee_{S \in B_{\mathcal{P}}} (S \setminus \{\chi_1\}) \models \gamma$ . Now, since each  $S \setminus \{\chi_1\}$  is a (conjunctively-interpreted) set consisting of literals built up from variables  $holds_{i,j}$ , it can be viewed as a term; subsequently,  $\bigvee_{S \in B_{\mathcal{P}}} (S \setminus \{\chi_1\})$  can be viewed as a DNF generated from the variables  $holds_{i,j}$ . Because we are interested in queries which do not contain any variable  $holds_{i,j}$ , we do not need to compute a DNF equivalent to  $\chi_1 \wedge \bigvee_{S \in B_{\mathcal{P}}} (S \setminus \{\chi_1\})$  by distributing the  $\wedge$  connective over the two DNFs  $\chi_1$  and  $\bigvee_{S \in B_{\mathcal{P}}} (S \setminus \{\chi_1\})$ : Computing a query-equivalent DNF (on  $Var(B)$ ) is suffi-

<sup>11</sup> No empirical results are reported concerning instances generated using 30 variables.

<sup>12</sup> Note that a similar algorithm (restricted to steps 1. and 2.) can be designed for the case  $\chi_1$  is a Horn cover formula (each term  $t$  in the algorithm above being replaced by a Horn CNF formula).

cient; to this purpose, it is sufficient to keep each term  $t$  from  $\chi_1$  that would lead to a consistent term if such an explicit distribution were made. This is exactly what step 2. in the algorithm above is intended to. Steps 3. and 4. just lead to minimize the size of the resulting DNF, by forgetting the variables  $holds_{i,j}$  which are irrelevant when queries built up from  $Var(B)$  are considered, only. Step 3. leads to compute in linear time a DNF formula equivalent to the most general consequence of the input DNF that is independent from any variable  $holds_{i,j}$ . Step 4. is a minimization stage that preserves equivalence and can be achieved in time at most quadratic in the size of the input. Clearly, steps 3. and 4. in the algorithm above can be omitted without questioning its correctness.

Computing  $B_{DNF}$  from  $DNF(B)$  can be viewed as a further compilation step, but which can be achieved in time polynomial in the size of  $DNF(B)$ . At an abstract level, it is reminiscent to the transformation of an ROBDD-compilation of an SBB into a “fat-free” one, dedicated to skeptical inference w.r.t. the lexicographic policy, as described in [16]; however, unlike the latter, our transformation is not dependent on the  $\mathcal{LE}$  policy but applies as well to the  $\mathcal{IP}$  one. An important fact is that this further compilation may lead to save some space since the size of the resulting DNF  $B_{DNF}$  is always at least as small as  $|\chi_1|$ . Those savings have a direct impact on the time needed to perform clausal query answering since a non-tautological clause  $\gamma$  is a logical consequence of  $B_{DNF}$  iff each term from  $B_{DNF}$  contains a literal from  $\gamma$ . Accordingly, the number of implicants in  $\chi_1$  gives an upper bound (up to a constant factor) of the time needed to perform clausal query answering from  $B_{DNF}$ . The price to be paid for the space savings that may result from the computation of  $B_{DNF}$  lies in a loss of flexibility. Indeed, from  $DNF(B)$ , it is possible to perform tractable clausal query answering even if the stratification or the selection policy (in  $\{\mathcal{IP}, \mathcal{LE}\}$ ) change with time. No re-compilation is required at all. Obviously enough, it is not the case any longer if the formula stored as a compilation is  $B_{DNF}$  instead of  $DNF(B)$ , since both the stratification and the selection policy determines  $B_{\mathcal{P}}$ , and this is a parameter on which  $B_{DNF}$  depends heavily. Of course, once both the stratification and the selection policy are fixed, it is a good idea to replace  $DNF(B)$  by  $B_{DNF}$ ; however, since  $B_{DNF}$  is derived from  $DNF(B)$ , the size of the latter is significant for the compilation process. This observation and the fact that flexibility is an important feature of our compilation method explain why we have focused on the size of  $\chi_1$  in our experiments.

## 8. Related Work

Our approach is closely related to previous works centered on the complexity of inference from belief bases and their compilation.

From the compilability side, we have exploited some compilability results reported in [12,13,11] in order to achieve our analysis. In contrast to ours, these works are not directly centered on skeptical inference from SBBs but on other related forms of nonmonotonic inference, like circumscription and belief revision. [23] presents compilability results for inference and model checking in the penalty logic setting (the inference relation considered in penalty logic encompasses  $\sim_{\forall}^{\mathcal{L}^{\mathcal{E}}}$  as a specific case). Our non-compilability results are consistent with those reported in [23], showing especially that the flexibility requirement is computationally demanding.

From the computational complexity side, the closest works are [41,16,26]. Our work aims at completing the complexity results pointed out in these papers, by focusing on other  $\models$ -tractable fragments of propositional logic. Interestingly, we have considered fragments that are *complete* for propositional logic, i.e., every propositional formula can be turned into an equivalent formula from any of them, while [41,16] mainly consider the Horn CNF class, which is not complete (see nevertheless later on the discussion about the ROBDD class considered in [16]). Accordingly, any SBB can be compiled in our framework, even if it is not composed of Horn CNF formulas. [18] shows how knowledge compilation can prove helpful to improve various forms of closed-world reasoning. We consider the same target classes for compilation, but different inference relations (which are nevertheless connected since closed-world reasoning can be reduced to assumption-based reasoning, i.e., inference from a supernormal default theory, which is a specific case of SBB - see [27,32,30,48]).

The idea of compiling an SBB is not new. A very natural way to implement inference from an SBB  $B$  consists in computing  $B_{\mathcal{P}}$ . We refer to it as the basic approach to compile  $B$ . Once this is done, inference is reduced to classical entailment. Since classical entailment is “only”  $\text{coNP}$ -complete, this basic approach amounts to knowledge compilation: the generation of the set of preferred subbases is the compilation step and query answering from the compiled form is easier than from the original SBB from the computational complexity point of view [10]. Like our approach, the basic one cannot be achieved in polynomial time in the general case ( $B_{\mathcal{P}}$  can easily contain exponentially many elements

when  $\mathcal{P} \in \{\mathcal{IP}, \mathcal{LE}\}$ ). However, our transformation is much more flexible. First of all, many knowledge compilation functions can be used within it (and some of them may achieve the objective of keeping the size “small enough” for some instances). Furthermore, re-partitioning the belief base can be achieved without requiring any expensive re-compilation step, when compiled using our approach, while this is not the case when the basic one is considered. Finally, it is obvious that, in the general case, there is no guarantee that every element of  $B_{\mathcal{P}}$  belongs to a  $\models$ -tractable class, while on-line tractability can be ensured by our approach.

More sophisticated compilation-based approaches to inference from SBBs are given in [3] and [16].

The approach presented in [3] aims at reducing many forms of skeptical inference from an SBB (including those based on the  $\mathcal{IP}$  and  $\mathcal{LE}$  policies) to the possibilistic one. Given a selection policy, this is achieved by compiling the given SBB into an SBB equivalent to it when interpreted under  $\sim_{\forall}^{\mathcal{PO}}$ . From the compilability point of view, the aim is to reduce the complexity of inference to  $\Delta_2^p[\mathcal{O}(\log n)]$ , through a pre-processing. The framework in which this approach takes place (labelled knowledge bases) is more general than the SBB one (especially, the preferential information which are considered take the form of a partial pre-ordering over beliefs, not a total one). Loosely speaking, this approach leads to take account for any disjunction built up from explicit beliefs from the base. While several simplification techniques may enable to remove much of them in practice, the size of the resulting compiled SBB can easily be exponential in the size of the original SBB (not surprisingly, this seems the price to be paid to reduce the  $\sim_{\forall}^{\mathcal{IP}}$  inference - the decision problem of which is  $\Pi_2^p$ -complete - to the  $\sim_{\forall}^{\mathcal{PO}}$  inference). Compared with the basic approach, the approach presented in [3] has the major advantage that no information is lost during the compilation step, which can be achieved in an incremental way using a syntactic combination operator. Compared with our approach, it ensures that the combination of a compiled base containing  $n$  formulas with a new incoming base containing  $m$  formulas leads to a new compiled base containing at most  $n \times m$  formulas; in our approach, it is much more difficult to give such a tight bound on the size of a compiled base, issued from the re-compilation of a given compiled base augmented with a new incoming formula. Contrastingly, the combination approach does not enable a re-partitioning of beliefs if no re-compilation is done (unless the resulting base is not filtered – but filtering is a way to possibly avoid a combinatorial blow-up). Finally, unlike our approach, the combination approach does



not necessarily lead to a base from a  $\models$ -tractable class.

Close to this work is [5] which aims at computing a formula equivalent to an SBB skeptically interpreted under the  $\mathcal{LE}$  policy. In this work, the objective is to reduce a  $\Delta_2^p$ -complete problem to a  $\text{coNP}$  one through a pre-processing. Here again, the size of the resulting compiled SBB can easily be exponential in the size of the original SBB (note that Proposition 10 shows that this can be avoided for sure provided that query-equivalence is considered). Moreover, the flexibility requirement is not satisfied in the general case and there is no guarantee that the resulting base belongs to a  $\models$ -tractable class.

From the compilation side, the closest approaches to ours are [16,22,23]. In [16], an ROBDD-based algorithm for  $\sim_{\forall}^{\mathcal{LE}}$  inference is presented (see [9] for a presentation of ROBDDs). From an ROBDD-compilation of an SBB, skeptical inference w.r.t.  $\mathcal{LE}$  is shown tractable as long as CNF queries are considered. [22,23] extends this work in two directions. On the one hand, a more general form of inference is considered. On the other hand, a more succinct tractable fragment of propositional logic (the DNNF one) is considered as a target language for compilation. Our work completes those approaches by (1) considering other selection policies (especially  $\mathcal{IP}$ ) and (2) focusing on other tractable fragments of propositional logic as target classes, showing that  $\sim_{\forall}^{\mathcal{LE}}$  inference remains intractable or becomes tractable according to the class into which the SBB is compiled. Especially, new tractable restrictions are provided; furthermore, the corresponding target classes cannot be compared with the ROBDD one w.r.t. succinctness (see [21]). Thus, some propositional formulas have ROBDD representations which are exponentially larger than some of their DNF representations, and the converse also holds. From the practical side, it seems that both classes have their own set of favourable instances (i.e., those leading to compiled forms of reasonable size). Thus, DNF-compilations (or more generally Horn cover compilations) are apparently more suited to critically constrained bases, while ROBDD compilations are more adapted to underconstrained bases (see [49,46]). For instance, [46] reports on successful compilations of hard random 3-CNF formulas with 200 variables while experiments with Bryant's ROBDD package shows difficult to derive within a reasonable amount of time an ROBDD representation of such formulas with more than 50 variables.

## 9. Conclusion

The coherence-based approach to inconsistency handling offers an important and quite popular framework for reasoning under inconsistency. While it has been investigated in depth from a logical point of view, less efforts have been spent from the practical side and few implementations exist. To which extent is the approach convenient when large-size applications are considered is a question that mainly remains unanswered. An explanation relies on the fact that inference is known as intractable in the coherence-based framework. To fill this gap, it is important to point out tractable restrictions and to design techniques for circumventing such an intractability, i.e., to increase the set of instances feasible from the practical side.

Along this line of research, we have shown in this paper how knowledge compilation techniques can be used to compile SBBs in order to make skeptical inference more efficient. Investigating the compilability of skeptical inference from SBBs, we have first shown that computational gains are hard to be expected from compilation in the worst case. Nevertheless, we have demonstrated that improvements can be expected (as long as the size of the compiled form remains “small enough”) for all the selection policies under consideration, except  $\mathcal{LE}$ . Focusing on four compilation functions found in the literature, tractable fragments have also been exhibited for both  $\mathcal{IP}$  and  $\mathcal{LE}$ .

Our results give both good news and bad news. Among the good news is the fact that knowledge compilation appears as an interesting approach to enhance clausal inference since skeptical inference from a  $\mathcal{C}$ -normal base is typically computationally easier than from an unconstrained base (just compare Table 3 with Table 1) and that tractable cases have been pointed out for every selection policy. Our preliminary empirical results also give some evidence about the feasibility of the approach, despite the fact that DNF-compilations are seemingly only suited to very constrained SBBs. The bad news here are that the computational improvement expected from a compilation approach is not systematic; thus, the complexity of inference w.r.t.  $\mathcal{LE}$  policy – one of the most interesting ones when logical properties and cautiousness are considered [4,15] – does not necessarily decrease when normal SBBs are considered. Note that the difficulty to lower the complexity of  $\Delta_2^p$ -complete problems by focusing on formulas belonging to a  $\models$ -tractable class is consistent with many results obtained so far in other contexts (e.g., in belief revision [41]).

This work calls for several perspectives. The major one concerns the empirical side. A more intensive experimental evaluation must be conducted to assess the computational benefits which can be expected from the approach. Considering the Horn cover class - strictly more succinct than the DNF one - as the target class would certainly push back the computational limitations encountered in practice. It would also be interesting to evaluate in depth the computational space savings that can be achieved in practice through the generation of  $B_{DNF}$  from  $DNF(B)$ , as explained in Section 7. On another side, it would be interesting to consider other target classes for compilation functions  $COMP$  and to investigate the corresponding  $COMP$ -normal forms, both from the complexity point of view and from the practical side. Finally, an additional perspective would be to compare our approach with the sequential combination of the basic compilation approach, i.e., the one consisting in computing  $B_{\mathcal{P}}$  from  $B$  (or the combination one reported in [3]) with standard knowledge compilation techniques (enabling to turn each element of  $B_{\mathcal{P}}$  into a formula from a  $\models$ -tractable class); this way, tractability w.r.t. the resulting base would be ensured. While it is clear that such an approach (with two successive compilation steps) is less flexible than ours (especially w.r.t. the possibility to re-partition beliefs), it would be interesting to empirically compare the sizes of the resulting bases.

## Acknowledgements

The authors would like to thank the anonymous referees for many interesting suggestions and comments. Many thanks to Klaus Wagner for his help with  $\Delta_2^p$ -complete problems. Many thanks as well to Salem Benferhat for discussions about the compilation of SBBs, to Yacine Boufkhad for his implementation of Davis/Logemann/Loveland procedure, to Claudette Cayrol, Marie-Christine Lagasquie-Schiex and Thomas Schiex for providing us with details of the experimentations reported in [16].

The authors have been partly supported by the the Région Nord/Pas-de-Calais through the IRCICA Consortium and by the European Community FEDER Program.

## References

- [1] S. Benferhat. Computing specificity in default reasoning. In D. Gabbay and P. Smets, editors, *Algorithms for Uncertainty and Defeasible Reasoning*, volume 5 of *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, pages 147–177. Kluwer Academic, 2000.
- [2] S. Benferhat, C. Cayrol, D. Dubois, J. Lang, and H. Prade. Inconsistency management and prioritized syntax-based entailment. In *Proceedings of the 13<sup>th</sup> International Joint Conference on Artificial Intelligence*, pages 640–645, Chambéry (France), 1993.
- [3] S. Benferhat, D. Dubois, J. Lang, H. Prade, A. Saffiotti, and P. Smets. A general approach for inconsistency handling and merging information in prioritized knowledge bases. In *Proceedings of the 6<sup>th</sup> International Conference on Principles of Knowledge Representation and Reasoning*, pages 466–477, Trento (Italy), 1998.
- [4] S. Benferhat, D. Dubois, and H. Prade. How to infer from inconsistent beliefs without revising. In *Proceedings of the 14<sup>th</sup> International Joint Conference on Artificial Intelligence*, pages 1449–1455, Montreal (Canada), 1995.
- [5] S. Benferhat, S. Kaci, D. Le Berre, and M.-A. Williams. Weakening conflicting information for iterated revision and knowledge integration. In *Proceedings of the 17<sup>th</sup> International Joint Conference on Artificial Intelligence*, pages 109–115, Seattle (WA), 2001.
- [6] Y. Boufkhad, E. Grégoire, P. Marquis, B. Mazure, and L. Saïs. Tractable cover compilations. In *Proceedings of the 15<sup>th</sup> International Joint Conference on Artificial Intelligence*, pages 122–127, Nagoya (Japan), 1997.
- [7] F. Bouquet and Ph. Jégou. Using OBDDs to handle dynamic constraints. *Information Processing Letters*, 62:111–120, 1997.
- [8] G. Brewka. Preferred subtheories: an extended logical framework for default reasoning. In *Proceedings of the 11<sup>th</sup> International Joint Conference on Artificial Intelligence*, pages 1043–1048, Detroit (MI), 1989.
- [9] R. E. Bryant. Graph-based algorithms for boolean function manipulation. *IEEE Transactions on Computers*, C-35:677–691, 1986.
- [10] M. Cadoli and F.M. Donini. A survey on knowledge compilation. *AI Communications*, 10:137–150, 1998.
- [11] M. Cadoli, F.M. Donini, P. Liberatore, and M. Schaerf. The size of a revised knowledge base. *Artificial Intelligence*, 115(1):25–64, 1999.
- [12] M. Cadoli, F.M. Donini, and M. Schaerf. Is intractability of non-monotonic reasoning a real drawback? *Artificial Intelligence*, 88:215–251, 1996.
- [13] M. Cadoli, F.M. Donini, M. Schaerf, and R. Silvestri. On compact representations of propositional circumscription. *Theoretical Computer Science*, 182:183–202, 1997.
- [14] C. Cayrol and M.-C. Lagasque-Schiex. On the complexity of non-monotonic entailment in syntax-based approaches. In *Proc. of the ECAI'94 Workshop on Algorithm, Complexity and Commonsense Reasoning*, Amsterdam (Netherlands), 1994.
- [15] C. Cayrol and M.-C. Lagasque-Schiex. Non-monotonic syntax-based entailment: a classification of consequence relations. In *Proceedings of the European Conference on Symbolic and*

- Quantitative Approaches to Reasoning and Uncertainty*, pages 107–114, Fribourg (Switzerland), 1995.
- [16] C. Cayrol, M.-C. Lagasquie-Schiex, and T. Schiex. Nonmonotonic reasoning: from complexity to algorithms. *Annals of Mathematics and Artificial Intelligence*, 22(3–4):207–236, 1998.
  - [17] S. Coste-Marquis and P. Marquis. Compiling stratified belief bases. In *Proceedings of the 14<sup>th</sup> European Conference on Artificial Intelligence*, pages 23–27, Berlin (Germany), 2000.
  - [18] S. Coste-Marquis and P. Marquis. Knowledge compilation for circumscription and closed-world reasoning. *Journal of Logic and Computation*, 11(4):579–607, August 2001.
  - [19] A. Darwiche. Compiling devices into decomposable negation normal form. In *Proceedings of the 16<sup>th</sup> International Joint Conference on Artificial Intelligence*, pages 284–289, Stockholm (Sweden), 1999.
  - [20] A. Darwiche. A compiler for deterministic decomposable negation normal form. In *Proceedings of the 18<sup>th</sup> National Conference on Artificial Intelligence*, pages 627–634, 2002.
  - [21] A. Darwiche and P. Marquis. A perspective on knowledge compilation. In *Proceedings of the 17<sup>th</sup> International Joint Conference on Artificial Intelligence*, pages 175–182, 2001.
  - [22] A. Darwiche and P. Marquis. Compilation of propositional weighted bases. In *Proc. of the 9<sup>th</sup> International Workshop on Non-Monotonic Reasoning (NMR'02)*, pages 6–14, Toulouse (France), 2002.
  - [23] A. Darwiche and P. Marquis. Compiling propositional weighted bases. *Artificial Intelligence*, 2004. to appear.
  - [24] M. Davis, G. Logemann, and D. Loveland. A machine program for theorem proving. *Communications of the ACM*, 5:394–397, 1962.
  - [25] A. del Val. Tractable databases: how to make propositional unit resolution complete through compilation. In *Proceedings of the 4<sup>th</sup> International Conference on Principles of Knowledge Representation and Reasoning*, pages 551–561, Bonn, 1994.
  - [26] T. Eiter and T. Lukasiewicz. Default reasoning from conditional knowledge bases: complexity and tractable cases. *Artificial Intelligence*, 124(2):169–241, 2000.
  - [27] D.V. Etherington. Relating default logic and circumscription. In *Proceedings of the 10<sup>th</sup> International Joint Conference on Artificial Intelligence*, pages 489–494, Milano (Italy), 1987.
  - [28] M.R. Garey and D.S. Johnson. *Computers and intractability: a guide to the theory of NP-completeness*. Freeman, 1979.
  - [29] G. Gogic, H. Kautz, C. Papadimitriou, and B. Selman. The comparative linguistics of knowledge representation. In *Proceedings of the 14<sup>th</sup> International Joint Conference on Artificial Intelligence*, pages 862–869, Montreal (Canada), 1995.
  - [30] G. Gottlob. Complexity results for nonmonotonic logics. *Journal of Logic and Computation*, 2:397–425, 1992.
  - [31] A. Hunter. Paraconsistent logics. In D. Gabbay and P. Smets, editors, *Reasoning with Actual and Potential Contradictions*, volume 2 of *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, pages 12–36. Kluwer Academic, 1998.
  - [32] K. Inoue and N. Helft. On theorem provers for circumscription. In *Proceedings of the*

- 8<sup>th</sup> Biennial Conference of the Canadian Society for Computational Studies of Intelligence, pages 212–219, Ottawa, 1990.
- [33] S. Kraus, D. Lehmann, and M. Magidor. Nonmonotonic reasoning, preferential models and cumulative logics. *Artificial Intelligence*, 44(1-2):167–207, 1990.
- [34] H. Levesque, D. Mitchell, and B. Selman. Hard and easy distribution of SAT problems. In *Proceedings of the 10<sup>th</sup> National Conference on Artificial Intelligence*, pages 440–446, San Jose (CA), 1992.
- [35] P. Liberatore. On the compilability of diagnosis, planning, reasoning about actions, belief revision, etc. In *Proceedings of the 6<sup>th</sup> International Conference on Principles of Knowledge Representation and Reasoning*, pages 144–155, Trento (Italy), 1998.
- [36] P. Marquis. Knowledge compilation using theory prime implicates. In *Proceedings of the 14<sup>th</sup> International Joint Conference on Artificial Intelligence*, pages 837–843, Montreal (Canada), 1995.
- [37] P. Marquis. Consequence finding algorithms. In D. Gabbay and P. Smets, editors, *Algorithms for Uncertain and Defeasible Reasoning*, volume 5 of *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, pages 41–145. Kluwer Academic Publishers, 2000.
- [38] B. Nebel. Belief revision and default reasoning: syntax-based approaches. In *Proceedings of the 2<sup>nd</sup> International Conference on Principles of Knowledge Representation and Reasoning*, pages 417–428, Cambridge (MA), 1991.
- [39] B. Nebel. Belief revision. In P. Gärdenfors, editor, *Cambridge Tracts in Theoretical Computer Science*, volume 29, chapter Syntax-based approaches to belief revision, pages 52–88. Cambridge University Press, Cambridge (UK), 1992.
- [40] B. Nebel. Base revision operations and schemes: semantics, representation and complexity. In *Proceedings of the 11<sup>th</sup> European Conference on Artificial Intelligence*, pages 341–345, Amsterdam (Netherlands), 1994.
- [41] B. Nebel. How hard is it to revise a belief base? In D. Gabbay and P. Smets, editors, *Belief Change*, volume 3 of *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, pages 77–145. Kluwer Academic, 1998.
- [42] C. Papadimitriou. *Computational complexity*. Addison–Wesley, 1994.
- [43] G. Pinkas and R. Loui. Reasoning from inconsistency: a taxonomy of principles for resolving conflict. In *Proceedings of the 3<sup>rd</sup> International Conference on Principles of Knowledge Representation and Reasoning*, pages 709–719, Cambridge (MA), 1992.
- [44] D. Poole. A logical framework for default reasoning. *Artificial Intelligence*, 36:27–47, 1988.
- [45] R. Reiter and J. de Kleer. Foundations of assumption-based truth maintenance systems: preliminary report. In *Proceedings of the 6<sup>th</sup> National Conference on Artificial Intelligence*, pages 183–188, Seattle (WA), 1987.
- [46] R. Schrag. Compilation for critically constrained knowledge bases. In *Proceedings of the 15<sup>th</sup> National Conference on Artificial Intelligence*, pages 510–515, Portland (OR), 1996.
- [47] B. Selman and H. Kautz. Knowledge compilation and theory approximation. *Journal of the ACM*, 43:193–224, 1996.

- [48] J. Stillman. The complexity of propositional default logics. In *Proceedings of the 10<sup>th</sup> National Conference on Artificial Intelligence*, pages 794–799, San Jose (CA), 1992.
- [49] T.E. Uribe and M.E. Stickel. Ordered binary decision diagrams and the Davis-Putnam procedure. In *Proceedings of the 1<sup>st</sup> International Conference on Constraints in Computational Logic*, 1994.
- [50] K.W. Wagner. Bounded query classes. *SIAM Journal of Computing*, 19(5):833–846, 1990.
- [51] K.W. Wagner. Personal communication. unpublished, 2000.

## Proofs

### Proposition 6

The complexity of FORMULA  $\vdash_{\forall}^{\mathcal{P}}$  from an SBB and of its restrictions to clause and literal inference for  $\mathcal{P} \in \{\mathcal{PO}, \mathcal{LO}, \mathcal{IP}, \mathcal{LE}\}$  is as reported in the following table.

Table 7  
Complexity of skeptical inference from SBBs (general case).

$\mathcal{P}$	FORMULA / CLAUSE / LITERAL $\vdash_{\forall}^{\mathcal{P}}$
$\mathcal{PO}$	$\Delta_2^p[\mathcal{O}(\log n)]$ -complete [41]
$\mathcal{LO}$	$\Delta_2^p$ -complete [40]
$\mathcal{IP}$	$\Pi_2^p$ -complete [38]
$\mathcal{LE}$	$\Delta_2^p$ -complete [14]

*Proof of Proposition 6.* We give a proof here for the sake of completeness, despite the fact that many of the results reported in this proposition are not new ones. As a notable exception, we show how some hardness results for FORMULA inference can be easily modified to the LITERAL inference case (which results in more general hardness results).

- $\mathcal{PO}$ : In order to prove the complexity results, it is sufficient to show that FORMULA  $\vdash_{\forall}^{\mathcal{PO}}$  belongs to  $\Delta_2^p[\mathcal{O}(\log n)]$  and that its restriction LITERAL  $\vdash_{\forall}^{\mathcal{PO}}$  is  $\Delta_2^p[\mathcal{O}(\log n)]$ -hard.
- \* The membership of FORMULA  $\vdash_{\forall}^{\mathcal{PO}}$  to  $\Delta_2^p[\mathcal{O}(\log n)]$  is a direct consequence of Theorem 6.5 from [41] (since inference from a SBB  $(\Delta_2, \dots, \Delta_k)$  revised by  $\Delta_1$  w.r.t. the cut base revision scheme corresponds exactly to FORMULA  $\vdash_{\forall}^{\mathcal{PO}}$ ).

\* The  $\Delta_2^p[\mathcal{O}(\log n)]$ -hardness of LITERAL  $\sim_{\forall}^{\mathcal{PO}}$  can be easily derived by adapting the hardness proof of Theorem 6.5 from [41]. Indeed, let us consider the following  $\Delta_2^p[\mathcal{O}(\log n)]$ -complete problem from [50] that is called PARITY(SAT):

- \* **Input:** A sequence  $\chi_1, \dots, \chi_n$  of formulas from  $PROP_{PS}$  s.t.  $\chi_i$  is inconsistent implies that  $\chi_{i+1}$  is inconsistent ( $i \in 1..n-1$ ).
- \* **Query:** Is the maximum index  $i$  ( $i \in 1..n$ ) s.t.  $\chi_i$  is consistent an odd number?

Let us associate to each instance of this problem<sup>13</sup> a SBB  $B = (\Delta_1, \dots, \Delta_{n+1})$  and a literal  $new \in PS$ , where

$$\Delta_1 = \left\{ \bigwedge_{i=1}^n (new_i \Rightarrow \chi_i), new \Leftrightarrow \left( \bigvee_{k=0}^{(n-1)/2} \left( \bigwedge_{l=1}^{2k+1} new_l \wedge \bigwedge_{l=2k+2}^n \neg new_l \right) \right) \right\}$$

and each  $\Delta_i$  ( $i \in 2..n+1$ ) is s.t.  $\Delta_i = \{new_{i-1}\}$ . Each variable  $new_i$  ( $i \in 1..n$ ) is a new variable, not occurring in  $\chi_1, \dots, \chi_n$ . This is also the case for  $new$ . Clearly enough, both  $B$  and  $new$  can be computed in time polynomial in the size of  $\chi_1, \dots, \chi_n$ . Now,  $\chi_1, \dots, \chi_n$  is a “yes”-instance of PARITY(SAT) iff  $B \sim_{\forall}^{\mathcal{PO}} new$  holds.

- $\mathcal{LO}$ : In order to prove the complexity results, it is sufficient to show that FORMULA  $\sim_{\forall}^{\mathcal{LO}}$  belongs to  $\Delta_2^p$  and that its restriction LITERAL  $\sim_{\forall}^{\mathcal{LO}}$  is  $\Delta_2^p$ -hard.
  - \* The membership of FORMULA  $\sim_{\forall}^{\mathcal{LO}}$  to  $\Delta_2^p$  is a direct consequence of Theorem 8 from [40] (or Corollary 1 from [14]) since inference from a SBB  $(\Delta_2, \dots, \Delta_k)$  revised by  $\Delta_1$  w.r.t. the linear-base revision scheme corresponds exactly to FORMULA  $\sim_{\forall}^{\mathcal{LO}}$ .
  - \* For the  $\Delta_2^p$ -hardness of LITERAL  $\sim_{\forall}^{\mathcal{LO}}$ , see Corollary 1 from [14].
- $\mathcal{IP}$ : In order to prove the complexity results, it is sufficient to prove that FORMULA  $\sim_{\forall}^{\mathcal{IP}}$  belongs to  $\Pi_2^p$  and that its restriction LITERAL  $\sim_{\forall}^{\mathcal{IP}}$  is  $\Pi_2^p$ -hard.

<sup>13</sup> We assume without loss of generality that  $n$  is odd - if it is not the case, we add to the sequence the formula  $\chi_{n+1} = false$ .



- \* The membership of FORMULA  $\sim_{\forall}^{\mathcal{I}\mathcal{P}}$  to  $\Pi_2^p$  is a direct consequence of Theorem 20 from [39] since inference from a SBB  $(\Delta_2, \dots, \Delta_k)$  revised by  $\Delta_1$  w.r.t. the prioritized base revision scheme corresponds exactly to FORMULA  $\sim_{\forall}^{\mathcal{I}\mathcal{P}}$ .
- \* For the  $\Pi_2^p$ -hardness of LITERAL  $\sim_{\forall}^{\mathcal{I}\mathcal{P}}$ , we slightly modify the proof of Theorem 14 from [38] so that it still holds when queries are limited to literals. Let  $G = \forall a_1 \dots \forall a_n \exists b_1 \dots \exists b_m F$  a 2- $\overline{QBF}$  formula where  $Var(F) = \{a_1, \dots, a_n, b_1, \dots, b_m\}$ . To  $G$  we associate the SBB  $B = (\Delta_1, \Delta_2)$  where  $\Delta_1 = \{new \Rightarrow F\}$  and  $\Delta_2 = \{a_1, \dots, a_n, \neg a_1, \dots, \neg a_n, new\}$  with  $new \in PS \setminus Var(F)$ , and the literal  $new$ . Clearly enough, both  $B$  and  $new$  can be computed in time polynomial in the size of  $G$ . Now,  $G$  is valid iff  $B \sim_{\forall}^{\mathcal{I}\mathcal{P}} new$  holds. Since the hardness result holds for  $k = 2$ , it also holds for supernormal default theories without stratification.
- $\mathcal{L}\mathcal{E}$ : In order to prove the complexity result, it is sufficient to prove that FORMULA  $\sim_{\forall}^{\mathcal{L}\mathcal{E}}$  belongs to  $\Delta_2^p$  and that its restriction LITERAL  $\sim_{\forall}^{\mathcal{L}\mathcal{E}}$  is  $\Delta_2^p$ -hard.
  - \* The membership of FORMULA  $\sim_{\forall}^{\mathcal{L}\mathcal{E}}$  to  $\Delta_2^p$  is a direct consequence of Theorem 5.17 from [41] since inference from a SBB  $B = (\Delta_2, \dots, \Delta_n)$  revised by  $\Delta_1$  w.r.t. the lexicographic base revision scheme corresponds exactly to  $\sim_{\forall}^{\mathcal{L}\mathcal{E}}$ .
  - \* For the LITERAL  $\sim_{\forall}^{\mathcal{L}\mathcal{E}}$ -hardness, see [14].

□

### Proposition 10

The results reported in the following table hold. **U** (resp. **F**) means that the inference problem is unflexible (resp. flexible). **W** (resp. **S**) means that the expected compilability strength is weak (resp. strong). "no $\downarrow$  2" means "no" unless the polynomial hierarchy collapses at the second level, while "no $\downarrow$  3" means "no" unless the polynomial hierarchy collapses at the third level.

*Proof of Proposition 10.*

- **U - W.**
  - \*  $\mathcal{P}\mathcal{O}$  and  $\mathcal{L}\mathcal{O}$ . The unique preferred subbase of the SBB  $(\Delta, \leq)$  w.r.t. the policy is by definition a subset of  $\Delta$ . Hence, its size is polynomial in the size of  $\Delta$ . By construction, this (conjunctively-interpreted) preferred subbase is a formula query-equivalent to  $(\Delta, \leq)$  (skeptically) interpreted under the corresponding policy.

Table 8  
 Compilability of skeptical inference from SBBs.

$\mathcal{P}$	U - W	F - W	U - S	F - S
$\mathcal{PO}$	yes	no↓ 3	no↓ 2	no↓ 2
$\mathcal{LO}$	yes	no↓ 3	no↓ 2	no↓ 2
$\mathcal{IP}$	no↓ 3	no↓ 3	no↓ 2	no↓ 2
$\mathcal{LE}$	yes	no↓ 3	no↓ 2	no↓ 2

- \*  $\mathcal{IP}$ . This results comes from the fact that (1) it is not possible to associate a circumscribed formula  $CIRC(\Sigma)$  to a query-equivalent polyspace formula unless the polynomial hierarchy collapses at the third level (see Theorem 7 from [13]) and that (2)  $CIRC(\Sigma)$  is equivalent over  $Var(\Sigma)$  to the two-strata SBB  $(\{\Sigma\}, \{\neg x \mid x \in Var(\Sigma)\})$  skeptically interpreted under the  $\mathcal{IP}$  policy (see e.g., [27]).
- \*  $\mathcal{LE}$ . Given a SBB  $B = (\Delta_1, \dots, \Delta_k)$ , we first compute the profile of  $B$ , i.e., a vector consisting of  $k$  numbers  $p_1, \dots, p_k$  (in binary notation), where  $p_i$  ( $i \in 1 \dots k$ ) is the number of formulas from  $\Delta_i$  that belong to *every* preferred subbase of  $B$  w.r.t.  $\mathcal{LE}$ . Once this is done, we associate to  $B$  the propositional formula  $COMP(B) =$

$$\bigwedge_{i=1}^k \left( \bigwedge_{j=1}^{card(\Delta_i)} (holds_{i,j} \Rightarrow \phi_{i,j}) \right) \wedge EXA(p_i, \bigcup_{j=1}^{card(\Delta_i)} \{holds_{i,j}\})$$

where  $holds_{i,j}$  are new variables, not occurring in  $B$ . Each  $holds_{i,j}$  is just a name given to the  $j^{th}$  formula from  $\Delta_i$ . Let  $S$  be any subbase of  $B$ ; it is easy to show that  $S$  and the formula  $\bigwedge_{i=1}^k (\bigwedge_{j=1}^{card(\Delta_i)} (holds_{i,j} \Rightarrow \phi_{i,j})) \wedge \bigwedge_{\phi_{i,j} \in S} holds_{i,j}$  are query-equivalent, i.e., equivalent over  $Var(B)$  (the proof is similar to the one of Proposition 15). The claim is that  $COMP(B)$  is query-equivalent to  $B$  skeptically interpreted under  $\mathcal{LE}$  and the size of  $COMP(B)$  is polynomial in the size of  $B$ . Indeed, each conjunct  $EXA(p_i, \bigcup_{j=1}^{card(\Delta_i)} \{holds_{i,j}\})$  encodes a boolean function that evaluates to true iff *exactly*  $p_i$  variables in  $\bigcup_{j=1}^{card(\Delta_i)} \{holds_{i,j}\}$  are set to true. The point is that a polyspace formula encoding this function can easily be generated (just combine a  $card(\Delta_i)$ -bit adder with a comparator, see e.g., [11]).

- **F - W.**

\*  $\mathcal{PO}$ ,  $\mathcal{LO}$  and  $\mathcal{LE}$ . Let  $\Sigma \in 3\text{-SAT}_n$  be a 3-CNF formula built up from the set of variables  $\{x_1, \dots, x_n\}$ . Let  $\Sigma_n^{\text{max}}$  be the 4-CNF formula  $\bigwedge_{\gamma_i \in 3\text{-}C_n} \neg \text{new}_i \vee \gamma_i$  where  $3\text{-}C_n$  is the set of all 3-literal clauses that can be generated from  $\{x_1, \dots, x_n\}$  and the  $\text{new}_i$  are new variables, not among  $\{x_1, \dots, x_n\}$ . Obviously enough, each element  $\Sigma$  from  $3\text{-SAT}_n$  can be identified by the subset  $S_\Sigma$  of the variables  $\text{new}_i$  s.t.  $\gamma_i$  is a clause of  $\Sigma$  iff  $\text{new}_i \in S_\Sigma$ . Now, to the entire family of  $3\text{-SAT}_n$  instances, we associate the belief base  $\Delta_n = \bigcup_{\gamma_i \in 3\text{-}C_n} \{\text{new}_i\} \cup \{\Sigma_n^{\text{max}} \wedge \text{new}\}$  where  $\text{new}$  is a new variable, and to each element  $\Sigma$  of the family, we associate the query  $\text{new}$ <sup>14</sup> and the total pre-order  $\leq_\Sigma$  over  $\Delta_n$  s.t. the first stratum is  $S_\Sigma$ , the second stratum is the singleton  $\{\Sigma_n^{\text{max}} \wedge \text{new}\}$  and the third (and last) stratum is the complement of  $S_\Sigma$  into  $\bigcup_{\gamma_i \in 3\text{-}C_n} \{\text{new}_i\}$ . It is easy to show that  $\Sigma$  is satisfiable iff  $(\Delta_n, \leq_\Sigma) \vdash_{\mathcal{P}}^{\forall} \text{new}$ , whatever the selection policy  $\mathcal{P}$  may be among  $\mathcal{PO}$ ,  $\mathcal{LO}$  and  $\mathcal{LE}$ . Clearly enough, the fixed part  $\Delta_n$  is of size polynomial in  $n$  (since  $|\Sigma_n^{\text{max}}|$  is in  $\mathcal{O}(n^3)$ ). Moreover, the variable part  $\langle \text{new}, \leq_\Sigma \rangle$  can be computed in time polynomial in  $|\Sigma|$  (if  $\gamma_i$  occurs in  $\Sigma$  then put  $\text{new}_i$  in the first stratum, otherwise put it into the third one). Assume that there exists a polynomial  $p$  s.t. for every  $n > 0$ , there exists a data structure  $COMP(\Delta_n)$  whose size is  $p(n)$  and s.t. for any query  $\gamma$  over  $\{x_1, \dots, x_n\}$  and for any total pre-ordering  $\leq$  over  $\Delta_n$ , determining whether  $\gamma$  follows from  $COMP(\Delta_n)$  given  $\leq$  is in  $\text{coNP}$  and this is the case iff  $(\Delta_n, \leq_\Sigma) \vdash_{\mathcal{P}}^{\forall} \gamma$ , where  $\mathcal{P}$  is any selection policy among  $\mathcal{PO}$ ,  $\mathcal{LO}$  and  $\mathcal{LE}$ . Then we would be able to determine whether any instance  $\Sigma \in 3\text{-SAT}_n$  is satisfiable using a non-deterministic Turing machine with a polynomial advice  $A$ : if  $|Var(\Sigma)| = n$ , then the machine loads  $A(n) = COMP(\Delta_n)$ . Once this is done, it computes  $\leq_\Sigma$  in time polynomial in  $|\Sigma|$  and it checks whether  $\text{new}$  follows from  $COMP(\Delta_n)$  given  $\leq$ , which is in  $\text{coNP}$ . Since 3-SAT is complete for  $\text{NP}$ , this would imply  $\text{NP} \subseteq \text{coNP}/\text{poly}$ , and, as a consequence, the polynomial hierarchy would collapse at the third level.

\*  $\mathcal{IP}$ . This comes directly from the non-compilability result in the case **UW** presented above.

- **U - S** and **F - S**. The two rightmost columns are immediate consequences from the fact that (1) it is not possible to associate any consistent propositional formula  $\Sigma$  to a query-equivalent polyspace formula from a  $\models$ -tractable class

<sup>14</sup> This shows that the query could be considered as fixed without questioning the result.

unless the polynomial hierarchy collapses at the second level (cf. for instance the proof of Theorem 2 from [12]) and that (2) classical entailment from a consistent  $\Sigma$  is logically equivalent to skeptical inference from the one-stratum SBB ( $\{\Sigma\}$ ) whatever the selection policy among  $\mathcal{PO}$ ,  $\mathcal{LO}$ ,  $\mathcal{IP}$  and  $\mathcal{LE}$ .  $\square$

**Proposition 15**

Let  $B = (\Delta_1, \dots, \Delta_k)$  be an SBB and let  $COMP$  be any equivalence-preserving compilation function (for clause entailment).  $COMP(B)$  is a  $COMP$ -normal SBB equivalent to  $B$  on  $Var(B)$  w.r.t.  $\mathcal{P} \in \{\mathcal{PO}, \mathcal{LO}, \mathcal{IP}, \mathcal{LE}\}$ .

*Proof of Proposition 15.* Since for any  $\mathcal{P} \in \{\mathcal{PO}, \mathcal{LO}, \mathcal{IP}\}$  and every SBB  $B = (\Delta_1, \dots, \Delta_k)$ , every element  $S$  of  $B_{\mathcal{P}}$  always contains  $\Delta_1$  as a subset, the following bijection  $\beta$  does the job:  $\beta(\Delta_1) = \chi_1$ , and for  $\phi \in \bigcup_{i=2}^k \Delta_i$  ( $\phi$  is the  $j^{th}$  assumption of  $\Delta_i$ ),  $\beta(\{\phi\}) = \{holds_{i,j}\}$  and we define for every subbase  $S$  of  $B_{\mathcal{P}}$ ,  $\beta(S) = \chi_1 \cup \bigcup_{\phi \in S \setminus \Delta_1} \beta(\{\phi\})$ . It is clear that  $\chi_1$  is consistent; by assumption  $\Delta_1$  is consistent, hence it has a model which can be extended into a model of  $\chi_1$  by assigning every variable  $holds_{i,j}$  to 0). Now,  $S(\supseteq \Delta_1)$  is a consistent subbase of  $B$  iff  $\beta(S)(\supseteq \chi_1)$  is a consistent subbase of  $COMP(B)$ . Moreover,  $S$  and  $\beta(S)$  have the same consequences on  $Var(\Delta)$  since every formula  $\phi$  from stratum  $\Delta_i$  ( $i \geq 2$ ) is equivalent on  $Var(\Delta)$  to its name  $holds_{i,j}$  whenever  $\chi_1$  holds. Finally, the preferential information conveyed by the stratification of  $B$  is preserved in  $COMP(B)$  since every formula  $\phi$  from stratum  $\Delta_i$  ( $i \geq 2$ ) is replaced by its name  $holds_{i,j}$  in stratum  $\chi_i$ .  $\square$

**Proposition 17**

The complexity of  $\text{FORMULA} \vdash_{\mathcal{V}}^{\mathcal{P}}$  and of its restrictions to clause and literal inference for  $\mathcal{P} \in \{\mathcal{PO}, \mathcal{LO}, \mathcal{IP}, \mathcal{LE}\}$  from a  $\mathcal{C}$ -normal SBB is as reported in the following table.

*Proof of Proposition 17.*

- $\mathcal{PO}$ : When  $B = (\Delta_1, \dots, \Delta_k)$  is  $\mathcal{C}$ -normal, every formula  $\phi$  from  $\bigcup_{i=2}^k \Delta_i$  is a literal;  $B_{\mathcal{PO}}$  can be computed in time polynomial in  $|B|$  thanks to the following algorithm:

Since  $\Delta_1$  belongs to a  $\models$ -tractable class whenever  $B$  is  $\mathcal{C}$ -normal and since  $\bigvee_{j=2}^k (\bigvee_{\phi \in \Delta_j} \neg \phi)$  is a clause, the given algorithm runs in time polynomial in the size of  $B$ . Now, for every formula  $\psi$  of  $PROP_{PS}$ , we have  $B \vdash_{\mathcal{V}}^{\mathcal{PO}} \psi$  iff  $B_{\mathcal{PO}} \models \psi$  iff  $\Delta_1 \models (\bigvee_{\phi \in B_{\mathcal{PO}} \setminus \Delta_1} \neg \phi) \vee \psi$ . This shows that skeptical infer-

Table 9

Complexity of skeptical inference from  $\mathcal{C}$ -normal SBBs (upper bounds).

$\mathcal{P}$	FORMULA $\sim_{\forall}^{\mathcal{P}}$	CLAUSE / LITERAL $\sim_{\forall}^{\mathcal{P}}$
$\mathcal{PO}$	in coNP	in P
$\mathcal{LO}$	in coNP	in P
$\mathcal{IP}$	in coNP	in coNP
$\mathcal{LE}$	in $\Delta_2^p$	in $\Delta_2^p$

**for**  $i = 2$  **to**  $k$  **do**  
  **if**  $\Delta_1 \models \bigvee_{j=2}^i (\bigvee_{\phi \in \Delta_j} \neg \phi)$  **then**  
    **return**  $(\{\bigcup_{j=1}^{i-1} \Delta_j\})$   
  **end if**  
**end for**  
**return**  $(\{\bigcup_{j=1}^k \Delta_j\})$

ence w.r.t. the possibilistic policy can be polynomially many-one reduced to classical inference. Since classical inference belongs to coNP and since coNP is closed under polynomial reduction, the membership of FORMULA  $\sim_{\forall}^{\mathcal{PO}}$  to coNP follows.

In the specific case  $\psi$  is a clause,  $(\bigvee_{\phi \in B_{\mathcal{PO}} \setminus \Delta_1} \neg \phi) \vee \psi$  is a clause as well, and since  $\Delta_1$  belongs to a  $\models$ -tractable class, CLAUSE  $\sim_{\forall}^{\mathcal{PO}}$  belongs to P.

- $\mathcal{LO}$ : The proof is close to the  $\mathcal{PO}$  one. When  $B$  is  $\mathcal{C}$ -normal,  $B_{\mathcal{LO}}$  can be computed in time polynomial in  $|B|$ , thanks to the following algorithm:

$I \leftarrow \{1\}$   
**for**  $i = 2$  **to**  $k$  **do**  
  **if**  $\Delta_1 \not\models \bigvee_{j \in I \cup \{i\}} (\bigvee_{\phi \in \Delta_j} \neg \phi)$  **then**  
     $I \leftarrow I \cup \{i\}$   
  **end if**  
**end for**  
**return**  $(\{\bigcup_{j \in I} \Delta_j\})$

Since  $\Delta_1$  belongs to a  $\models$ -tractable class whenever  $B$  is  $\mathcal{C}$ -normal and since  $\bigvee_{j \in I \cup \{i\}} (\bigvee_{\phi \in \Delta_j} \neg \phi)$  is a clause, the algorithm above runs in polynomial time. Now, for every formula  $\psi$  of  $PROP_{PS}$ , we have  $B \sim_{\forall}^{\mathcal{LO}} \psi$  iff

$\Delta_1 \models (\bigvee_{\phi \in B_{\mathcal{L}\mathcal{O}} \setminus \Delta_1} \neg\phi) \vee \psi$ . This shows that skeptical inference w.r.t. the linear order policy can be polynomially many-one reduced to classical inference. Since classical inference belongs to **coNP** and since **coNP** is closed under polynomial reductions, the membership of **FORMULA**  $\sim_{\forall}^{\mathcal{L}\mathcal{O}}$  to **coNP** follows.

In the specific case when  $\psi$  is a clause,  $(\bigvee_{\phi \in B_{\mathcal{L}\mathcal{O}} \setminus \Delta_1} \neg\phi) \vee \psi$  is also a clause and since  $\Delta_1$  belongs to a  $\models$ -tractable class, **CLAUSE**  $\sim_{\forall}^{\mathcal{L}\mathcal{O}}$  belongs to **P**.

- **$\mathcal{IP}$** : We show the membership to **coNP** by exhibiting a preferred-models characterization of skeptical inference w.r.t. the inclusion-preference policy. Given  $I$  and  $J$  two models of  $\Delta_1$ , we note  $I < J$  iff there exists  $i \in 2 \dots k$  s.t.  $\{\phi \in \Delta_i \mid I \models \phi\} \supset \{\phi \in \Delta_i \mid J \models \phi\}$  holds and for every  $j \in 2 \dots i - 1$ ,  $\{\phi \in \Delta_j \mid I \models \phi\} = \{\phi \in \Delta_j \mid J \models \phi\}$ .  $<$  is a strict order (i.e., an irreflexive and transitive relation) over the set of models of  $\Delta_1$ . Now, we call preferred models of  $B$  the models of  $\Delta_1$  that are minimal w.r.t.  $<$ . Clearly enough,  $\{\phi \in \Delta \mid I \models \phi\}$  is a preferred subbase of  $B$  iff  $I$  is a preferred model of  $B$ . Accordingly, we have  $B \sim_{\forall}^{\mathcal{IP}} \psi$  iff every preferred model of  $B$  is a model of  $\psi$ . Let us now show that determining whether  $B \not\sim_{\forall}^{\mathcal{IP}} \psi$  holds belongs to **NP** when  $\psi$  is any formula. It is sufficient to prove that checking whether a given interpretation  $I$  is a preferred model of  $B$  can be achieved in time polynomial in  $|B|$ . Indeed, it is sufficient to guess  $I$ , then check in (deterministic) polynomial time that  $I$  is a preferred model of  $B$  and check that  $I \not\models \psi$  holds. Now, in order to check whether  $I$  is a preferred model of  $B$ , it is sufficient to check that  $S = \{\phi \in \Delta \mid I \models \phi\}$  (which can be computed in polynomial time given  $B$  and  $I$ ) is a preferred subbase of  $B$  w.r.t. to the inclusion-preference policy. This can be done thanks to the following algorithm:

```

S = (S1, S2, ..., Sk)
for For  $i = 2$  to  $k$  do
  for  $\phi \in \Delta_i \setminus S_i$  do
    if  $\bigcup_{j=1}^i S_j \cup \{\phi\}$  is consistent then
      return (not preferred)
    end if
  end for
end for
return (preferred)

```

Since  $S_1 = \Delta_1$  belongs to a  $\models$ -tractable class and since  $\neg(\bigwedge_{j=2}^i S_j \wedge \phi)$  is a clause, this algorithm runs in polynomial time.

- $\mathcal{LE}$ : The membership of formula  $\vdash_{\forall}^{\mathcal{IP}}$  to  $\Delta_2^p$  is a direct consequence of the general case (cf. Proposition 6).

□

### Proposition 18

The complexity of FORMULA  $\vdash_{\forall}^{\mathcal{IP}}$  and of its restrictions to clause and literal inference from *COMP*-normal SBBs is as reported in the following table.

Table 10  
Complexity of skeptical inference w.r.t.  $\mathcal{IP}$  from *COMP*-normal SBBs.

<i>COMP</i>	FORMULA $\vdash_{\forall}^{\mathcal{IP}}$	CLAUSE / LITERAL $\vdash_{\forall}^{\mathcal{IP}}$
<b>Blake</b>	coNP-complete	coNP-complete
<b>DNF</b>	coNP-complete	in P
<b>Horn cover</b>	coNP-complete	coNP-complete
<b>renamable Horn cover</b>	coNP-complete	coNP-complete

*Proof of Proposition 18.* All the membership results reported in the Table are direct consequences of the membership results of the general case (Proposition 17), except for the CLAUSE  $\vdash_{\forall}^{\mathcal{IP}}$  membership to P when the  $\mathcal{C}$ -normal base is a DNF formula. Hence, we shall focus on this result and on the hardness results.

- FORMULA  $\vdash_{\forall}^{\mathcal{IP}}$ : coNP-hardness can be proved through the reduction already used for showing the coNP-hardness of FORMULA  $\vdash_{\forall}^{\mathcal{PO}}$  and FORMULA  $\vdash_{\forall}^{\mathcal{LO}}$ : a formula  $\psi$  from *PROP<sub>PS</sub>* is valid iff  $B \vdash_{\forall}^{\mathcal{IP}} \psi$  holds, where  $B = (\{true\})$  is a  $\mathcal{C}$ -normal, one-stratum, belief base. Accordingly, this is a polynomial reduction from the validity problem of propositional logic (well-known to be coNP-complete) to skeptical inference w.r.t. the inclusion-preference policy, in the restricted case when  $\Delta_1$  is a Blake, DNF, Horn cover or renamable Horn cover formula.
- CLAUSE  $\vdash_{\forall}^{\mathcal{IP}}$ :

\* DNF: From points (1) and (2) of Lemma 21, we know that when

$B = (\Delta_1, \dots, \Delta_k)$  is s.t.  $\Delta_1$  is a DNF formula and  $\bigcup_{i=2}^k \Delta_i$  is a consistent

set of literals,  $B_{\subseteq}$  can be computed in polynomial time and every element  $S$  of  $B_{\subseteq}$  can be turned into a DNF formula in polynomial time. Moreover, it is obvious that filtering out  $B_{\mathcal{IP}}$  from  $B_{\subseteq}$  can be done in polynomial time. Since clausal entailment can be achieved in polynomial time from a DNF formula, the conclusion follows.

- \* Blake, Horn cover, renamable Horn cover: The membership of  $\text{CLAUSE} \sim_{\forall}^{\mathcal{IP}}$  to  $\text{coNP}$  is a consequence from  $\text{FORMULA} \sim_{\forall}^{\mathcal{IP}}$ .

It remains to show that  $\text{LITERAL} \sim_{\forall}^{\mathcal{IP}}$  is  $\text{coNP-hard}$ . It is well-known that extended closed world reasoning can be reduced to skeptical inference from some specific default theories [27,32]. Indeed, given a formula  $\Phi \in \text{PROP}_{PS}$  and a partition  $\langle P, Q, Z \rangle$  of  $\text{Var}(\Sigma)$ , for every formula  $\Psi \in \text{PROP}_{PS}$ , we have  $\text{ECWA}(\Phi, \langle P, Q, Z \rangle) \models \Psi$  iff  $\Psi$  is a skeptical consequence of the default theory  $\langle \Phi, \Delta \rangle$  where  $\Delta = \{\frac{\text{true}:l}{\perp} \mid l \in L_Q \text{ or } l \in L_{\bar{P}}\}$ . Equivalently,  $(\{\Phi\}, L_Q \cup L_{\bar{P}}) \sim_{\forall}^{\mathcal{IP}} \Psi$ . Clearly enough, this reduction can be achieved in polynomial time. Note that  $(\{\Phi\}, L_Q \cup L_{\bar{P}})$  is not a  $\mathcal{C}$ -normal SBB (even when  $\Phi$  belongs to a  $\models$ -tractable class) since  $L_Q$  is by definition an inconsistent set when  $Q \neq \emptyset$ . Nevertheless, it is known that literal inference from  $\Phi$  interpreted under  $\text{ECWA}$  is  $\text{coNP-hard}$  whenever  $\Phi$  is a Blake formula and  $Q = \emptyset$  (see Proposition 14 from [18]). Accordingly, the above reduction can be used to show that  $\text{LITERAL} \sim_{\forall}^{\mathcal{IP}}$  is  $\text{coNP-hard}$  when  $\phi$  is a Blake formula (even in the case when the number  $k$  of strata is 2). Actually, the  $\text{coNP-hardness}$  result still holds in the case when  $\Delta_1$  is a Krom formula.

Concerning the Horn cover class or the renamable Horn cover one, let us first prove the following lemma:

**Lemma 23.** Let  $B = (\Delta_1, \Delta_2, \dots, \Delta_k)$  be a SBB s.t.  $\bigcup_{i=2}^k \Delta_i$  is a subset of  $L_{PS}$  (not necessarily a consistent one) and  $\Delta_1$  is from the Horn cover class (resp. the renamable Horn cover one). Then we can compute in time polynomial in  $|B|$  a SBB  $B' = (\Delta'_1, \Delta'_2, \dots, \Delta'_k)$  which is equivalent to  $B$  on  $\text{Var}(\bigcup_{i=1}^k \Delta_i)$  w.r.t.  $\mathcal{P} \in \{\mathcal{PO}, \mathcal{LO}, \mathcal{IP}, \mathcal{LE}\}$  and s.t.  $\Delta'_1$  is from the Horn cover class (resp. the renamable Horn cover class) and  $\bigcup_{i=2}^k \Delta'_i$  is a set of positive literals (hence consistent).

*Proof of Lemma 23.* Let us consider the SBB  $(\Delta'_1, \Delta'_2, \dots, \Delta'_k)$  defined in the following way. Every  $\Delta'_i$  is s.t.  $i \geq 2$  is derived from  $\Delta_i$  by replacing every negative literal  $\phi_{i,j}$  by a new positive literal  $\text{holds}_{i,j}$  and  $\Delta'_1$  is obtained



by conjoining each disjunct of  $\Delta_1$  with the clauses  $\neg holds_{i,j} \vee \phi_{i,j}$  for every new literal which is introduced (hence, at most  $|\bigcup_{i=2}^k \Delta_i|$  clauses must be considered (which is bounded by  $|B|$ )). Clearly enough, the resulting base is equivalent to  $B$  on  $Var(\bigcup_{i=1}^k \Delta_i)$  w.r.t.  $\mathcal{P} \in \{\mathcal{PO}, \mathcal{LO}, \mathcal{IP}, \mathcal{LE}\}$  (the proof is similar to the one of Proposition 15). Conjoining every disjunct of the Horn cover formula  $\Delta_1$  with the conjunction of the Horn clauses  $\neg holds_{i,j} \vee \phi_{i,j}$  clearly leads to a Horn cover formula. As to the renamable Horn cover case, it is important to note that none of the variables  $holds_{i,j}$  that are introduced appears positively in  $\Delta'_1$ . Accordingly, when the renamable Horn cover class is considered, it is sufficient to conjoin every disjunct of the renamable Horn cover formula  $\Delta_1$  with the conjunction of the clauses  $\neg holds_{i,j} \vee \phi_{i,j}$  to obtain a renamable Horn cover formula (any substitution used to show any disjunct as a Horn CNF formula can still be used to show the Horn renamability of the resulting disjunct).  $\square$

This lemma shows that, when Horn cover formulas or renamable Horn cover formulas are considered, the case when  $\bigcup_{j=2}^k \Delta_j$  is a consistent set can be recovered (without loss of generality). Combining both polynomial reductions shows that LITERAL  $\vdash_{\forall}^{\mathcal{IP}}$  is coNP-hard when the formula of the first stratum belongs to the Horn cover class or the renamable Horn cover class (even in the case when the number  $k$  of strata is 2).  $\square$

### Proposition 19

The complexity of FORMULA  $\vdash_{\forall}^{\mathcal{LE}}$  and of its restrictions to clause and literal inference from *COMP*-normal SBBs is as reported in the following table.

Table 11  
Complexity of skeptical inference w.r.t.  $\mathcal{LE}$  from *COMP*-normal SBBs.

<i>COMP</i>	FORMULA $\vdash_{\forall}^{\mathcal{LE}}$	CLAUSE / LITERAL $\vdash_{\forall}^{\mathcal{LE}}$
<b>Blake</b>	$\Delta_2^p$ -complete	$\Delta_2^p$ -complete
<b>DNF</b>	coNP-complete	in P
<b>Horn cover</b>	$\Delta_2^p$ -complete	$\Delta_2^p$ -complete
<b>renamable Horn cover</b>	$\Delta_2^p$ -complete	$\Delta_2^p$ -complete

*Proof of Proposition 19.*

- Blake, Horn cover, renamable Horn cover: We have to show that FORMULA  $\sim_{\forall}^{\mathcal{L}\mathcal{E}}$  belongs to  $\Delta_2^p$  and that LITERAL  $\sim_{\forall}^{\mathcal{L}\mathcal{E}}$  is  $\Delta_2^p$ -hard.
- \* The membership of FORMULA  $\sim_{\forall}^{\mathcal{L}\mathcal{E}}$  to  $\Delta_2^p$  comes from the general case (see Proposition 17).
- \* It remains to show the hardness result for LITERAL  $\sim_{\forall}^{\mathcal{L}\mathcal{E}}$ .
- \* Blake:

We focus on the specific case when  $\Delta_1$  is a Krom formula. It is well-known that the set of prime implicates of a Krom formula can be easily computed in polynomial time (e.g., a resolution-based algorithm like Tison's version of old good Quine-McCluskey's algorithm achieves the goal). Hence showing the hardness result in the situation where  $\Delta_1$  is a Krom formula is sufficient.

We first need a preferred-models characterization of skeptical inference w.r.t. the lexicographic policy. Given  $I$  and  $J$  two models of  $\Delta_1$ , we note  $I < J$  iff there exists  $i \in 2 \dots k$  s.t.  $\text{card}(\{\phi \in \Delta_i \mid I \models \phi\}) > \text{card}(\{\phi \in \Delta_i \mid J \models \phi\})$  holds and for every  $j \in 2 \dots i - 1$ ,  $\text{card}(\{\phi \in \Delta_i \mid I \models \phi\}) = \text{card}(\{\phi \in \Delta_i \mid J \models \phi\})$ .  $<$  is a strict order (i.e., an irreflexive and transitive relation) over the set of models of  $\Delta_1$ . Now, we call preferred models of  $B$  the models of  $\Delta_1$  that are minimal w.r.t.  $<$  in their set. Clearly enough,  $\{\phi \in \Delta \mid I \models \phi\}$  is a preferred subbase of  $B$  iff  $I$  is a preferred model of  $B$ . Accordingly, we have  $B \sim_{\forall}^{\mathcal{L}\mathcal{E}} \psi$  iff every preferred model of  $B$  is a model of  $\psi$ .

We now need the following lemma showing that the set of models of any consistent 3-CNF formula  $\Sigma$  s.t.  $\text{Var}(\Sigma) = \{v_1, \dots, v_n\}$  can be associated to the set of minimum-cardinality models of a Krom formula computable in time polynomial in  $|\Sigma|$ .

Let  $\Sigma = \bigwedge_{i=1}^m (l_{i,1} \vee l_{i,2} \vee l_{i,3})$  be a consistent 3-CNF formula s.t.  $\text{Var}(\Sigma) = \{v_1, \dots, v_n\}$  (each  $l_{i,j}$  with  $i \in 1 \dots m$  and  $j \in 1 \dots 3$  denotes a literal built up from  $\text{Var}(\Sigma)$ ). To  $\Sigma$  we associate the formula  $Krom(\Sigma) = \bigwedge_{i=1}^m (x_{i,1} \vee x_{i,2}) \wedge (x_{i,1} \vee x_{i,3}) \wedge (x_{i,2} \vee x_{i,3}) \wedge \bigwedge_{i=1}^m \bigwedge_{j=1}^3 (x_{i,j} \vee l_{i,j}) \wedge \bigwedge_{i=1}^n (v_i \vee u_i) \wedge (\neg v_i \vee \neg u_i)$ . We have  $\text{Var}(Krom(\Sigma)) = \{v_1, \dots, v_n, u_1, \dots, u_n, x_{1,1}, x_{1,2}, x_{1,3}, x_{2,1}, x_{2,2}, x_{2,3}, \dots, x_{m,1}, x_{m,2}, x_{m,3}\}$ . As mentioned in [51], this mapping is close to the reduction from 3-SAT to VERTEX COVER (given in Theorem 3.3 of [28]). We derived:

**Lemma 24.**  $Krom(\Sigma)$  is a consistent Krom formula. It can be computed in time polynomial in  $|\Sigma|$  and for every interpretation  $I$  over  $Var(\Sigma)$ ,  $I$  is a model of  $\Sigma$  iff there exists a minimum-cardinality model  $I'$  of  $Krom(\Sigma)$  (i.e., a model for which the number of variables assigned to 1 is minimum) over  $Var(Krom(\Sigma))$  s.t.  $\forall i \in 1 \dots m \ I'(v_i) = I(v_i)$ .

*Proof of Lemma 24.* The fact that  $Krom(\Sigma)$  is a consistent Krom formula computable in time polynomial in  $|\Sigma|$  is obvious. Now, in  $Krom(\Sigma)$ , the conjunct  $\bigwedge_{i=1}^m (x_{i,1} \vee x_{i,2}) \wedge (x_{i,1} \vee x_{i,3}) \wedge (x_{i,2} \vee x_{i,3})$  ensures that every model of  $Krom(\Sigma)$  assigns at least  $2m$  variables to 1 and the conjunct  $\bigwedge_{i=1}^n (v_i \vee u_i) \wedge (\neg v_i \vee \neg u_i)$  ensures that in every model of  $Krom(\Sigma)$  exactly one of  $v_i$  and  $u_i$  are assigned to 1 for any  $i \in 1 \dots m$ . Accordingly, the cardinality of every model of  $Krom(\Sigma)$  is  $\geq 2m + n$ . The conjunct  $\bigwedge_{i=1}^m \bigwedge_{j=1}^3 (x_{i,j} \vee l_{i,j})$  ensures that to every model  $I$  of  $\Sigma$  we can associate at least one model  $I'$  of  $Krom(\Sigma)$  that extends  $I$  and assigns exactly  $2m + n$  variables to 1. Moreover, the converse also holds. Indeed, assume that there exists a minimum-cardinality model  $I'$  of  $Krom(\Sigma)$  whose restriction  $I$  to  $Var(\Sigma)$  is not a model of  $\Sigma$ . Then there exists a clause  $(l_{i,1} \vee l_{i,2} \vee l_{i,3})$  of  $\Sigma$  s.t.  $I(l_{i,1}) = I(l_{i,2}) = I(l_{i,3}) = 0$ . The conjunct  $\bigwedge_{i=1}^m \bigwedge_{j=1}^3 (x_{i,j} \vee l_{i,j})$  requires that  $I'(x_{i,1}) = I'(x_{i,2}) = I'(x_{i,3}) = 1$ . But in this case  $I'$  assigns at least  $2m + n + 1$  variables to 1, hence it is not a minimum-cardinality model of  $Krom(\Sigma)$ .  $\square$

We can now give a polynomial many-one reduction from  $MAX\text{-}SAT\text{-}ASG_{odd}$  to  $LITERAL \vdash_{\forall}^{\mathcal{L}\mathcal{E}}$  from a Blake-normal SBB.  $MAX\text{-}SAT\text{-}ASG_{odd}$  is the following decision problem:

- + **Input:** A propositional formula  $\Sigma$  over  $\{s_1, \dots, s_n\}$ .
- + **Query:** Is the maximal model  $I$  of  $\Sigma$  w.r.t. the lexicographic ordering induced by  $s_1 < \dots < s_n$  s.t.  $I(s_n) = 1$ ?

Let us first note that the  $\Delta_2^p$ -hardness of  $MAX\text{-}SAT\text{-}ASG_{odd}$  still holds in the restricted case when the formula  $\Sigma$  under consideration is a consistent 3-CNF formula (see e.g., [23]). Let  $Var(Krom(\Sigma)) = \{y_1, \dots, y_p\}$ . To  $\Sigma$  we associate the Blake-normal SBB  $B = (Blake(Krom(\Sigma)) \wedge \bigwedge_{y_i \in Var(Krom(\Sigma))} (\neg y_i \vee z_i) \wedge (\neg z_i \vee y_i)), \{\neg z_1, \dots, \neg z_p\}, \{s_1\}, \dots, \{s_n\})$  where each  $z_i$  is a new variable ( $i \in 1 \dots p$ ). This reduction can be

achieved in polynomial time since  $Krom(\Sigma) \wedge \bigwedge_{y_i \in Var(Krom(\Sigma))} (\neg y_i \vee z_i) \wedge (\neg z_i \vee y_i)$  is a Krom formula that can be computed in time polynomial in  $|\Sigma|$ , and the prime implicates of a Krom formula can be computed in polynomial time. Now  $\Sigma$  is a positive instance of MAX-SAT-ASG<sub>odd</sub> iff  $B \vdash_{\mathcal{V}}^{\mathcal{LE}} z_p$ . Indeed, to every variable  $y_i$  of  $Var(Krom(\Sigma))$  is associated an equivalent variable  $z_i$ , so that every model of  $Krom(\Sigma) \wedge \bigwedge_{y_i \in Var(Krom(\Sigma))} (\neg y_i \vee z_i) \wedge (\neg z_i \vee y_i)$  is fully characterized by its restriction to  $\{y_1, \dots, y_p\}$ . Moreover, minimum cardinality is preserved in the sense that the restriction of a minimum-cardinality model of  $Krom(\Sigma) \wedge \bigwedge_{y_i \in Var(Krom(\Sigma))} (\neg y_i \vee z_i) \wedge (\neg z_i \vee y_i)$  to  $\{y_1, \dots, y_p\}$  is a minimum-cardinality model of  $Krom(\Sigma)$ . Requiring that the second stratum of  $B$  is  $\{\neg z_1, \dots, \neg z_p\}$  forces the preferred models of  $B$  w.r.t.  $\mathcal{LE}$  to be among the minimum-cardinality models of  $Krom(\Sigma) \wedge \bigwedge_{y_i \in Var(Krom(\Sigma))} (\neg y_i \vee z_i) \wedge (\neg z_i \vee y_i)$ . Accordingly, using Lemma 24, every model  $I$  of  $\Sigma$  over  $\{s_1, \dots, s_n\}$  can be extended into (at least) one preferred model of  $(Blake(Krom(\Sigma) \wedge \bigwedge_{y_i \in Var(Krom(\Sigma))} (\neg y_i \vee z_i) \wedge (\neg z_i \vee y_i)), \{\neg z_1, \dots, \neg z_p\})$  w.r.t.  $\mathcal{LE}$ . By taking into account the  $n$  following strata  $\{s_1\}, \dots, \{s_n\}$  of  $B$ , we restrict the resulting set of preferred models to those extending the maximal model of  $\Sigma$  w.r.t. the lexicographic ordering induced by  $s_1 < \dots < s_n$ . Thus,  $s_n$  is assigned to 1 in this model iff it is assigned to 1 in every preferred model of  $B$  w.r.t.  $\mathcal{LE}$ .

Clearly enough, Lemma 24 can be used to reduce the problem LITERAL  $\models$  to LITERAL  $\vdash_{\mathcal{V}}^{\mathcal{LE}}$  in the restricted case when  $B$  is a Blake-normal SBB with two strata (the models of  $\Sigma$  are extended into the minimum-cardinality models of a Krom formula); this shows that LITERAL  $\vdash_{\mathcal{V}}^{\mathcal{LE}}$  from a Blake-normal SBB remains coNP-hard when  $k \geq 2$ .

- \* Horn cover, renamable Horn cover: The hardness result can be easily derived from the reduction given in the proof of Theorem 5.18 from [41] since inference from a SBB  $B = (\Delta_1, \dots, \Delta_n)$  revised by  $\phi$  (consistent) w.r.t. the lexicographic base revision scheme corresponds exactly to  $\vdash_{\mathcal{V}}^{\mathcal{LE}}$ . In this proof,  $\phi$  is a variable and the most priority stratum (named  $A_{n+2}$ ) consists of a single Horn CNF formula. Since  $\phi$  is consistent with this formula, the revised base is equivalent on the set of its variables w.r.t.  $\mathcal{LE}$  to the SBB obtained by expanding the most priority stratum with  $\phi$ . Now, since the formulas from the other strata are positive literals,

they form a consistent set and the resulting base is a compiled one in which  $\Delta_1$  is a Horn CNF formula. Since every Horn CNF formula is a Horn cover formula and *a fortiori* a renamable Horn cover formula, the expected result follows. Note that the  $\Delta_2^p$ -hardness of LITERAL  $\sim_{\forall}^{\mathcal{L}\mathcal{E}}$  still holds in the restricted case when the number of strata  $k = 2$  (this is a direct consequence of Theorem 5.15 from [41]).

- DNF:

\* FORMULA  $\sim_{\forall}^{\mathcal{L}\mathcal{E}}$ : We first give the following lemma:

**Lemma 21**

Let  $B = (\Delta_1, \dots, \Delta_k)$  be an SBB with  $\Delta = \bigcup_{i=1}^k \Delta_i$ . We recall that  $B_{\subseteq} = (\Delta_1, \dots, \Delta_k)_{\subseteq}$  is the set of all maximal (w.r.t.  $\subseteq$ ) consistent subbases of  $B = \bigcup_{i=1}^k \Delta_i$  containing  $\Delta_1$ . We have:

1. If  $\Delta_1 = \{\alpha_1 \vee \dots \vee \alpha_n\}$  where each  $\alpha_i$  ( $i \in 1 \dots n$ ) is a formula from  $PROP_{PS}$ ,  $B_{\subseteq} = \max_{\subseteq}(\{\Delta_1 \cup (S \cap \Delta) \mid S \in \bigcup_{i=1}^n (\{\alpha_i\}, \bigcup_{j=2}^k \Delta_j)_{\subseteq}\})$ .
2. If  $\alpha$  is a term and  $\bigcup_{j=2}^k \Delta_j$  is a consistent set of literals, then  $(\{\alpha\}, \bigcup_{j=2}^k \Delta_j)_{\subseteq}$  is the singleton  $\{\{\alpha\} \cup \{\phi \in \bigcup_{j=2}^k \Delta_j \mid \alpha \not\models \neg\phi\}\}$ .
3. If  $\alpha$  is a Horn CNF formula and  $\bigcup_{j=2}^k \Delta_j$  contains only negative literals, then  $(\{\alpha\}, \bigcup_{j=2}^k \Delta_j)_{\subseteq}$  is the singleton  $\{\{\alpha\} \cup \{\phi \in \bigcup_{j=2}^k \Delta_j \mid \alpha \not\models \neg\phi\}\}$ .

*Proof of Lemma 21.*

1. By definition, every element of  $B_{\subseteq}$  is of the form  $\Delta_1 \cup S$  where  $S$  is a maximal (w.r.t.  $\subseteq$ ) subset of  $\bigcup_{j=2}^k \Delta_j$  which is consistent with  $\Delta_1$ . Now,  $S$  is consistent with  $\Delta_1 = \alpha_1 \vee \dots \vee \alpha_n$  iff there exists  $i \in 1 \dots n$  s.t.  $S$  is consistent with  $\alpha_i$ . Hence, every  $S$  is a maximal (w.r.t.  $\subseteq$ ) subset of  $\bigcup_{j=2}^k \Delta_j$  which is consistent with at least one  $\alpha_i$  ( $i \in 1 \dots n$ ). This is equivalent to state that  $\{\alpha_i\} \cup S \in (\{\alpha_i\}, \bigcup_{j=2}^k \Delta_j)_{\subseteq}$  for at least one  $i \in 1 \dots n$ . Accordingly, we have  $B_{\subseteq} = \max_{\subseteq}(\{\Delta_1 \cup S \mid \exists i \in 1 \dots n \{\alpha_i\} \cup S \in (\{\alpha_i\}, \bigcup_{j=2}^k \Delta_j)_{\subseteq}\})$ . There are two possibilities for  $S$  as a subset of  $\bigcup_{j=2}^k \Delta_j$ :

- \* either  $\alpha_i \in S$ . In this case,  $\alpha_i \in \bigcup_{j=2}^k \Delta_j$ .
- \* or  $\alpha_i \notin S$ . In this case,  $\alpha_i \notin \bigcup_{j=2}^k \Delta_j$  since we could add  $\alpha_i$  to  $S$  while preserving consistency with  $\alpha_i$  if it were not the case. But then the maximality (w.r.t.  $\subseteq$ ) requirement on  $S$  would be violated.

Thus, for every  $S' \in (\{\alpha_i\}, \bigcup_{j=2}^k \Delta_j)_{\subseteq}$  s.t.  $S' = \{\alpha_i\} \cup S$ , we have:

- \*  $S = S'$  if  $\alpha_i \in \bigcup_{j=2}^k \Delta_j$ .
- \*  $S = S' \setminus \{\alpha_i\}$  otherwise.

In any case, we have  $S = S' \cap \bigcup_{j=2}^k \Delta_j$ . Thus, for every  $S' \in (\{\alpha_i\}, \bigcup_{j=2}^k \Delta_j)_{\subseteq}$ , we have  $\Delta_1 \cup (S' \cap \Delta)$   
 $= \Delta_1 \cup (S' \cap (\Delta_1 \cup \bigcup_{j=2}^k \Delta_j))$   
 $= \Delta_1 \cup ((S' \cap \Delta_1) \cup (S' \cap \bigcup_{j=2}^k \Delta_j))$   
 $= \Delta_1 \cup (S' \cap \bigcup_{j=2}^k \Delta_j)$ .

Hence, we have

$$B_{\subseteq} = \max_{\subseteq} (\{\Delta_1 \cup (S \cap \Delta) \mid S \in \bigcup_{i=1}^n (\{\alpha_i\}, \bigcup_{j=2}^k \Delta_j)_{\subseteq}\}).$$

2. When  $\alpha$  is a term and  $\bigcup_{j=2}^k \Delta_j$  is a consistent set of literals, for every subset  $S$  of  $\bigcup_{j=2}^k \Delta_j$ ,  $\{\alpha\} \cup S$  is consistent iff  $\{\alpha\} \cup \{l\}$  is consistent for every  $l \in S$ . Indeed,  $S$  does not contain a complementary pair of literals, hence  $\alpha$  is consistent with the conjunction of two literals of  $S$  iff it is consistent with each of them. Thus, it is sufficient to gather the literals of  $\bigcup_{j=2}^k \Delta_j$  with which  $\alpha$  is consistent to generate the (necessarily unique) maximal (w.r.t.  $\subseteq$ ) subset  $S$  of  $\bigcup_{j=2}^k \Delta_j$  with which  $\alpha$  is consistent.
3. If  $\alpha$  is a Horn CNF formula and  $\bigcup_{j=2}^k \Delta_j$  is a set of negative literals (hence a consistent set), then for every subset  $S = \{l_1, \dots, l_p\}$  of  $\bigcup_{j=2}^k \Delta_j$ ,  $\{\alpha\} \cup S$  is consistent iff  $\alpha \not\models \neg l_1 \vee \dots \vee \neg l_p$  iff  $\alpha \not\models \neg l_i$  for any  $i \in 1 \dots p$ . Indeed, the clause  $\neg l_1 \vee \dots \vee \neg l_p$  is a positive clause (it consists of positive literals only) and the positive prime implicates of a Horn CNF formula are unit clauses. Here again, it is sufficient to gather the literals of  $\bigcup_{j=2}^k \Delta_j$  with which  $\alpha$  is consistent to generate the (necessarily unique) maximal (w.r.t.  $\subseteq$ ) subset  $S$  of  $\bigcup_{j=2}^k \Delta_j$  with which  $\alpha$  is consistent.

□

We use the results of Lemma 21 in order to prove the membership of FORMULA  $\vdash_{\forall}^{\mathcal{L}\mathcal{E}}$  to coNP. We can compute  $B_{\subseteq}$  in polynomial time (see Lemma 21). Then, from  $B_{\subseteq}$ , we can derive  $B_{\mathcal{L}\mathcal{E}}$  in polynomial time. Now, for every formula  $\Psi \in PROP_{PS}$ , we have  $B \vdash_{\forall}^{\mathcal{L}\mathcal{E}} \Psi$  iff  $\forall S \in B_{\mathcal{L}\mathcal{E}}, S \models \Psi$ . This shows that skeptical inference w.r.t. the lexicographic policy from a DNF formula can be polynomially reduced to classical inference. Since classical inference

belongs to  $\text{coNP}$  and since  $\text{coNP}$  is closed under polynomial reduction, the membership to  $\text{coNP}$  follows.

It remains to show that  $\text{FORMULA} \sim_{\forall}^{\mathcal{L}\mathcal{E}}$  is  $\text{coNP}$ -hard. A formula  $\Psi$  from  $\text{PROP}_{PS}$  is valid iff  $B \sim_{\forall}^{\mathcal{L}\mathcal{E}} \Psi$  holds with  $B = (\{true\})$  being a  $\mathcal{C}$ -normal, one-stratum belief base. Accordingly, this is a polynomial reduction from the validity problem of propositional logic (known to be  $\text{coNP}$ -complete) to skeptical inference w.r.t. the lexicographic policy.

\*  $\text{CLAUSE} \sim_{\forall}^{\mathcal{L}\mathcal{E}}$ : similar to the proof for the DNF form in Proposition 18, replacing  $B_{\mathcal{I}\mathcal{P}}$  by  $B_{\mathcal{L}\mathcal{E}}$ . □

**Proposition 20**

$\text{CLAUSE} \sim_{\forall}^{\mathcal{I}\mathcal{P}}$  and  $\text{CLAUSE} \sim_{\forall}^{\mathcal{L}\mathcal{E}}$  from a  $\mathcal{C}$ -normal SBB  $B = (\Delta_1, \dots, \Delta_k)$  where  $\Delta_1$  is a Horn cover formula and  $\bigcup_{i=2}^k \Delta_i$  contains only negative literals are in  $\text{P}$ .

*Proof of Proposition 20.* From points (1) and (3) of Lemma 21, we know that when  $B = (\Delta_1, \dots, \Delta_k)$  is s.t.  $\Delta_1$  is a Horn cover formula and  $\bigcup_{i=2}^k \Delta_i$  contains only negative literals,  $B_{\subseteq}$  can be computed in polynomial time and every element  $S$  of  $B_{\subseteq}$  can be turned into a Horn cover formula in polynomial time. Moreover, it is obvious that filtering out  $B_{\mathcal{I}\mathcal{P}}$  (or  $B_{\mathcal{L}\mathcal{E}}$ ) from  $B_{\subseteq}$  can be done in polynomial time. Since clausal entailment can be achieved in polynomial time from a Horn cover formula, the conclusion follows. □