Compiling Stratified Belief Bases

Sylvie Coste-Marquis[1] and **Pierre Marquis**[2]

Abstract. Many coherence-based approaches to inconsistency handling within propositional belief bases have been proposed so far. They consist in selecting one or several preferred consistent subbases of the given (usually inconsistent) stratified belief base (SBB), then using classical inference from some of the selected subbases. Unfortunately, deciding the corresponding inference relations is typically hard from the computational complexity point of view. In this paper, we show how some knowledge compilation techniques for classical inference can be used to circumvent the intractability of such sophisticated inference relations. For several families of compiled SBBs and several selection policies, the complexity of skeptical inference is identified. Interestingly, some tractable restrictions are exhibited.

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. To cope with this problem, we adhere to the *coherence-based approach to inconsistency handling*. Pieces of information are represented by *propositional stratified belief bases* (SBB for short), i.e., finite sets of propositional formulas equipped with a total pre-order which represents the available preferences over the given beliefs.

Following [20], 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. Clearly enough, there are many ways to extend the given total preorder over formulas into a preference relation over sets of beliefs. In this paper, four important subbases selection policies are considered [1], namely the *possibilistic* policy, the *linear order* policy, the *inclusion-preference* policy and the *lexicographic* policy. Additionally, several entailment principles can be defined [20, 3]; 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 finally (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 relation [9]. Consequently, the rest of the paper focuses on this relation.

A major drawback of inference from a SBB lays on its computational cost which makes it impractical for many instances. Thus an important question is: how to circumvent the intractability of inference from a 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 a 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 [6] 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 these techniques can ensure that the objective of enhancing inference is reached in the worst case (because the size of the compiled form can be exponentially larger than the size of the original knowledge base), experiments have shown such approaches valuable in many practical situations.

In the following, we show how such compilation techniques for classical inference from knowledge bases can be used to possibly improve sophisticated nonmonotonic inference from SBBs. Interestingly, any equivalence-preserving knowledge compilation technique can be used and the given stratification of beliefs can change without requiring the SBB to be re-compiled from scratch. Clearly enough, such a compilation approach can prove helpful 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. We achieve it, focusing on four different knowledge compilation functions found in the literature.

2 Formal Preliminaries

$PROP_{PS}$ denotes the propositional language built up from a denumerable set PS of symbols, the boolean constants $true$ and $false$, and the connectives in the standard way. $Var(\Sigma)$ denotes the set of propositional variables occurring in Σ. The size of a formula Σ from $PROP_{PS}$, noted $|\Sigma|$, is the number of signs (symbols and connectives) used to write it. For every subset V of PS, L_V is the set of literals built up from the propositional symbols of V. A negative literal is a literal of the form $\neg x$, where $x \in PS$.

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

[1] CRIL et IUT de Lens, Université d'Artois, rue de l'Université, SP 16, 62307 Lens cedex, FRANCE, email: coste@cril.univ-artois.fr

[2] CRIL, Université d'Artois, rue de l'Université, SP 16, 62307 Lens cedex, FRANCE, email: marquis@cril.univ-artois.fr

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 Δ_k^p, Σ_k^p and Π_k^p of the polynomial hierarchy PH (see [19] for details). $\Delta_2^p[\mathcal{O}(log\ n)]$ is the class of problems which can be decided using only logarithmically many calls to an NP oracle. Let us recall that a decision problem is said at the k^{th} level of PH iff it belongs to Δ_{k+1}^p, and is either Σ_k^p-hard or Π_k^p-hard. It is strongly believed that PH does not collapse (at any level), i.e., is a truly infinite hierarchy (for every integer k, $\mathsf{PH} \neq \Sigma_k^p$).

3 Inference from Stratified Belief Bases

Let us first define what a SBB is:

Definition 3.1 (stratified belief bases) *A* stratified belief base *(SB-B) B is an ordered pair $B = \langle\Delta, \leq\rangle$, where $\Delta = \{\phi_1, \ldots, \phi_n\}$ is a finite set of formulas from $PROP_{PS}$ and $\leq$ is a total pre-order over Δ. Every subset S of Δ is a* subbase *of B.*

It is equivalent to define B as a finite sequence $(\Delta_1, \ldots, \Delta_k)$ of subbases of Δ, where each Δ_i ($i \in 1 .. k$) is the non-empty set which contains all the minimal elements of $\Delta \setminus (\bigcup_{j=1}^{i-1} \Delta_j)$[3] w.r.t. $\leq$. Clearly enough, $\{\Delta_1, \ldots, \Delta_k\}$ is a partition of Δ. Each subset Δ_i ($i \in 1 .. k$) is called a *stratum* of B, and i is the priority level of each formula of Δ_i. Intuitively, the lowest the priority level of a formula the highest its plausibility. Given a subbase S of B, we note S_i ($i \in 1 .. k$) the subset of S defined by $S_i = S \cap \Delta_i$.

In the following, we assume that Δ_1 is a consistent set containing all the certain beliefs (i.e., the pieces of *knowledge*) of Δ. This assumption can be done without loss of generality since when no certain beliefs are available, it is sufficient to add $true$ to Δ as its unique minimal element w.r.t. $\leq$. Accordingly, a SBB $B = (\Delta_1, \ldots, \Delta_k)$ 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 [5].

There are several ways to use the information given by a SBB corresponding to several epistemic attitudes. Following Pinkas and Loui’s analysis [20], inference from a 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 to every SBB B 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 3.2 (selection policies)
Let $B = (\Delta_1, \ldots, \Delta_k)$ be a 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 Δ'_i ($i \in 1..k$) is defined by $\Delta'_i = \Delta_i$ if $\Delta_i \cup \bigcup_{j=1}^{i-1} \Delta'_j$ is consistent, $\emptyset$ otherwise.*
- *The set $B_{\mathcal{IP}}$ of all the preferred subbases of B w.r.t. the inclusion-preference policy is $\{S \subseteq \Delta$ s.t. S is consistent and $\forall S' \subseteq \Delta$ s.t. S' is consistent, $\forall i \in 1..k$ $((\forall j < i(S'_j = S_j)) \Rightarrow S_i \not\subset S'_i)\}$.*
- *The set $B_{\mathcal{LE}}$ of all the preferred subbases of B w.r.t. the lexicographic policy is $\{S \subseteq \Delta$ s.t. S is consistent and $\forall S' \subseteq \Delta$ s.t. S' is consistent, $\forall i \in 1..k$ $((\forall j < i(card(S'_j) = card(S_j))) \Rightarrow card(S_i) \not< card(S'_i))\}$.*

All preferred subbases S of B (w.r.t. any of the above policy) are (by construction) consistent sets. Moreover, since Δ_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 $B_{\mathcal{LE}} \subseteq B_{\mathcal{IP}} \subseteq B_{\subseteq}$, where $B_{\subseteq} = \{S \subseteq \Delta$ s.t. S is consistent, $\Delta_1 \subseteq S$, and $\forall \phi \in \Delta \setminus S$, $S \cup \{\phi\}$ is inconsistent$\}$ is the set of all maximal (w.r.t. $\subseteq$) consistent subbases of B. It is worth noting that 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). The elements of $B_{\mathcal{IP}}$ correspond to the so-called preferred subtheories of [5].

Given a selection policy, several entailment principles can be considered, especially credulous inference, argumentative inference, skeptical inference. Among them, we specifically focus on skeptical reasoning which is the most rational one [9].

Definition 3.3 (skeptical inference) *Let $B = (\Delta_1, \ldots, \Delta_k)$ be a SBB, $\mathcal{P}$ a policy for the generation of preferred subbases, and Ψ a formula from $PROP_{PS}$. Ψ is a* skeptical consequence *of B w.r.t. $\mathcal{P}$, noted $B \mid\!\sim_{\forall}^{\mathcal{P}} \Psi$, iff $\forall S \in B_{\mathcal{P}}$, $S \models \Psi$.*

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

Definition 3.4 (FORMULA $\mid\!\sim_{\forall}^{\mathcal{P}}$**)**
Let $\mid\!\sim_{\forall}^{\mathcal{P}}$ be any inference relation among $\{\mid\!\sim_{\forall}^{\mathcal{PO}}, \mid\!\sim_{\forall}^{\mathcal{LO}}, \mid\!\sim_{\forall}^{\mathcal{IP}}, \mid\!\sim_{\forall}^{\mathcal{LE}}\}$. FORMULA $\mid\!\sim_{\forall}^{\mathcal{P}}$ *is the following decision problem:*

- **Input:** *A SBB $B = (\Delta_1, \ldots, \Delta_k)$ and a formula Ψ from $PROP_{PS}$.*
- **Query:** *Does $B \mid\!\sim_{\forall}^{\mathcal{P}} \Psi$ hold?*

CLAUSE $\mid\!\sim_{\forall}^{\mathcal{P}}$ (resp. LITERAL $\mid\!\sim_{\forall}^{\mathcal{P}}$) is the restriction of FORMULA $\mid\!\sim_{\forall}^{\mathcal{P}}$ to the case where Ψ is required to be a CNF formula (resp. a term).

The following complexity results can be found in the literature[4] (see Theorem 8 from [17] (or Corollary 1 from [8]), Theorems 5.17 and 6.5 from [18], Theorem 15 from [16]).

Proposition 3.1 (skeptical inference from SBBs)
The complexity of FORMULA $\mid\!\sim_{\forall}^{\mathcal{P}}$ *from a SBB and of its restrictions to clause and literal inference for $\mathcal{P} \in \{\mathcal{PO}, \mathcal{LO}, \mathcal{IP}, \mathcal{LE}\}$ is reported in Table 1.*

$\mathcal{P}$	FORMULA / CLAUSE / LITERAL $\mid\!\sim_{\forall}^{\mathcal{P}}$
$\mathcal{PO}$	$\Delta_2^p[\mathcal{O}(log\ n)]$-complete
$\mathcal{LO}$	Δ_2^p-complete
$\mathcal{IP}$	Π_2^p-complete
$\mathcal{LE}$	Δ_2^p-complete

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

4 Knowledge Compilation

Knowledge compilation (see [6] for a survey) gathers several techniques which prove helpful in the objective of improving inference, in particular clause deduction [23], but also diagnosis, planning, belief revision, etc [14]. In the following, we focus on knowledge compilation techniques for improving classical inference, i.e., for making the following decision problem easier:

[3] By convention, $\bigcup_{j=1}^{0} \Delta_j = \emptyset$.

[4] Actually, previous complexity results typically concern the FORMULA $\mid\!\sim_{\forall}^{\mathcal{P}}$ 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 $\mid\!\sim_{\forall}^{\mathcal{P}}$ and LITERAL $\mid\!\sim_{\forall}^{\mathcal{P}}$ problems.

Definition 4.1 (FORMULA $\models$**)**
FORMULA $\models$ *is the following decision problem:*
- **Input:** *Two formulas* Σ *and* Ψ *from* $PROP_{PS}$
- **Query:** *Does* $\Sigma \models \Psi$ *hold?*

CLAUSE $\models$ (resp. LITERAL $\models$) is the restriction of FORMULA $\models$ to the case where Ψ 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., [7, 14]). Indeed, if the size of the compiled form is exponentially larger than the size of the original KB Σ, 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, LITERAL $\models$ is compilable while both FORMULA $\models$ and CLAUSE $\models$ are (probably) not compilable[5].

The second category contains works that are much more oriented towards the design of compilation algorithms and their empirical evaluations. Thus, among others, [21, 13, 15, 22, 4, 12] present equivalence-preserving knowledge compilation methods for clause deduction. All these methods aim at computing a formula $COMP(\Sigma)$ equivalent to Σ, and from which CLAUSE $\models$ belongs to P. Stated otherwise, compiling Σ consists in turning it into a formula belonging to a tractable class for clause deduction.

Abusing words, a formula of $PROP_{PS}$ is said $\models$-tractable when it belongs to such a tractable class of formulas. Considering $\models$-tractable KB Σ is helpful for the CLAUSE $\models$ problem, since determining whether a clause is entailed by a $\models$-tractable KB Σ can be achieved in polynomial time, while the problem is coNP-complete when Σ 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 is the set of formulas given in prime implicates normal form,
- the *DNF* class is the set of formulas given in disjunctive normal form (DNF),
- the *Horn cover* class is the set of disjunctions of Horn CNF formulas,
- the *renamable Horn cover* class is the set of disjunctions of renamable Horn CNF formulas.

The Blake class (resp. the DNF class) is the target class of the compilation function described in [21] (resp. in [22]). The Horn cover class and the renamable Horn cover class are target classes for the tractable covers compilation functions given in [4]. Of course, all these compilation functions $COMP$ are subject to the limitation explained above: in the worst case, the size of the compiled form $COMP(\Sigma)$ is exponential in the size of Σ. Nevertheless, there is some empirical evidence that some of these approaches can prove computationally valuable for many instances of the CLAUSE $\models$ problem (see e.g., the experimental results given in [22, 4]).

5 Compiling Stratified Belief Bases

In the following, we will only consider compiled SBBs, i.e., SBBs in which the certain beliefs form a $\models$-tractable formula and all the remaining beliefs are represented by literals:

Definition 5.1 (compiled SBBs) *A SBB* $B = (\Delta_1, \ldots, \Delta_k)$ *is* compiled *iff* Δ_1 *is* $\models$*-tractable and* $\bigcup_{i=2}^{k} \Delta_i \subseteq L_{PS}$.

Interestingly, for every SBB, there exists an equivalent compiled SBB with equivalence defined as:

Definition 5.2 (equivalence of SBBs) *Let* $B = (\Delta_1, \ldots, \Delta_k)$ *and* $B\prime = (\Delta\prime_1, \ldots, \Delta\prime_l)$ *be two SBBs. Let* V *be a subset of* PS *and* $\mathcal{P}$ *a selection policy.* B *and* $B\prime$ *are* equivalent *on* V *w.r.t.* $\mathcal{P}$ *iff there exists a bijection* β *from* $B_{\mathcal{P}}$ *to* $B\prime_{\mathcal{P}}$ *s.t. for every* $S \in B_{\mathcal{P}}$ *and every formula* Ψ *from* $PROP_V$, $S \models \Psi$ *iff* $\beta(S) \models \Psi$.

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

Definition 5.3 (compiling SBBs) *Let* $B = (\Delta_1, \ldots, \Delta_k)$ *be a SBB (with* $\Delta = \bigcup_{i=1}^{k} \Delta_i$*) and let* $COMP$ *be any equivalence-preserving compilation function (for clause deduction). Without loss of generality, let us assume that every stratum* Δ_i *(*$i \in 1 .. k$*) of* B *is totally ordered (w.r.t. any order) and let us note* $\phi_{i,j}$ *the* j^{th} *formula of* Δ_i *w.r.t. this order.*
The SBB $COMP(B) = (\chi_1, \ldots, \chi_k)$ *where* $\chi_i = \{new_{i,1}, \ldots, new_{i,card(\Delta_i)}\}$ *for* $i \in 2 .. k$*, each* $new_{i,j} \in L_{PS} \setminus L_{Var(\Delta)}$*, and* $\chi_1 = COMP(\Delta_1 \cup (\bigcup_{i=2}^{k} \{\bigwedge_{j=1}^{card(\Delta_i)} (new_{i,j} \Rightarrow \phi_{i,j})\}))$ *is the* compilation *of* B *w.r.t.* $COMP$.

This transformation basically consists in giving a name (under the form of a new literal) to each assumption of Δ and in storing the correspondance assumption/name with the certain beliefs *before* compiling them for clause deduction. As an important fact, our compilation approach does not question equivalence on the original language.

Proposition 5.1 (equivalence preservation) *Let* $B = (\Delta_1, \ldots, \Delta_k)$ *be a SBB (with* $\Delta = \bigcup_{i=1}^{k} \Delta_i$*) and let* $COMP$ *be any equivalence-preserving compilation function (for clause deduction).* $COMP(B)$ *is a compiled SBB equivalent to* B *on* $Var(\Delta)$ *w.r.t.* $\mathcal{P} \in \{\mathcal{PO}, \mathcal{LO}, \mathcal{IP}, \mathcal{LE}\}$.

The motivation for our definition of compiled SBBs B relies on the fact that making $\models$-tractable every formula of Δ is not sufficient for improving CLAUSE $\mid\sim^{\mathcal{P}}_{\forall}$ in the general case. Indeed, forming preferred subbases of B requires to check the consistency of conjunctions of such formulas and $\models$-tractable formulas 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. More specifically, tractable classes of formulas are typically not closed under conjunction (especially, for all 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, because every assumption from $\bigcup_{i=2}^{k} \Delta_i$ is a literal, and whatever the compilation function used to compile Δ_1 is, the consistency of any subbase of a compiled SBB B which contains Δ_1 can be checked in polynomial time. Thus, any equivalence-preserving compilation function can be used for compiling a SBB. Since many of the existing compilation functions have no comparable computational behaviours (each of them performs better than the others on some instances), such a flexibility is a major advantage.

[5] The existence of an equivalence-preserving compilation function $COMP$ s.t. it is guaranteed that for every propositional CNF formula Σ, FORMULA $\models$ (resp. CLAUSE $\models$) from $COMP(\Sigma)$ is in P and $|COMP(\Sigma)|$ is polynomially bounded in $|\Sigma|$ would make P = NP (just because determining whether a formula is valid is coNP-complete) (resp. the polynomial hierarchy to collapse at the second level (see [23, 6] for more details)).

6 Complexity of Inference from Compiled SBBs

The purpose of compiling a SBB is to enhance inference from it. This objective can be achieved only if (1) the size of the compiled SBB is not exponentially larger than the size of the original SBB, and (2) inference from the compiled SBB is easier than inference from the original SBB. Because every inference relation considered in this paper is supra-classical (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 compilation of a SBB remains polynomial in the size of the original SBB, whatever the compilation function is. 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 their computational value, we can nevertheless expect computational benefits for many instances. In this section, we show the extent to which (2) can be achieved, depending on the inference relation under consideration, the nature of the query (formula, clause, literal) and the compilation function $COMP$ used.

We have identified the following complexity results:

Proposition 6.1 (skeptical inference from compiled SBBs) *The complexity of* FORMULA $\mid\!\sim_{\forall}^{\mathcal{P}}$ *and of its restrictions to clause and literal inference for $\mathcal{P} \in \{\mathcal{PO}, \mathcal{LO}, \mathcal{IP}, \mathcal{LE}\}$ from a compiled SBB is reported in Table 2.*

$\mathcal{P}$	FORMULA $\mid\!\sim_{\forall}^{\mathcal{P}}$	CLAUSE / LITERAL $\mid\!\sim_{\forall}^{\mathcal{P}}$
$\mathcal{PO}$	coNP-complete	in P
$\mathcal{LO}$	coNP-complete	in P
$\mathcal{IP}$	coNP-complete	coNP-complete
$\mathcal{LE}$	Δ_2^p-complete	Δ_2^p-complete

Table 2. Complexity of skeptical inference from compiled SBBs.

Proposition 6.1 shows that compiling a SBB can actually make inference computationally easier. Actually, compiling makes all inference relations considered in this paper easier, except $\mid\!\sim_{\forall}^{\mathcal{LE}}$.

Within Proposition 6.1, no assumption on the nature of the compiled SBB has been done. In order to possibly obtain tractability results for inference w.r.t. the $\mathcal{IP}$ policy and the $\mathcal{LE}$ policy, restricted compiled SBBs must be considered. In the following, we focus on compiled SBBs of the form $COMP(B)$ where $COMP$ is a compilation function which maps any propositional formula into a Blake, DNF, Horn cover or renamable Horn cover formula.

Proposition 6.2 (skeptical inference w.r.t. $\mathcal{IP}$ from $COMP(B)$) *The complexity of* FORMULA $\mid\!\sim_{\forall}^{\mathcal{IP}}$ *and of its restrictions to clause and literal inference from compiled SBBs $COMP(B)$ is reported in Table 3.*

$COMP$	FORMULA $\mid\!\sim_{\forall}^{\mathcal{IP}}$	CLAUSE / LITERAL $\mid\!\sim_{\forall}^{\mathcal{IP}}$
Blake	coNP-complete	coNP-complete
DNF	coNP-complete	in P
Horn cover	coNP-complete	coNP-complete
renamable Horn cover	coNP-complete	coNP-complete

Table 3. Complexity of skeptical inference w.r.t. $\mathcal{IP}$ from $COMP(B)$.

Proposition 6.3 (skeptical inference w.r.t. $\mathcal{LE}$ from $COMP(B)$) *The complexity of* FORMULA $\mid\!\sim_{\forall}^{\mathcal{LE}}$ *and of its restrictions to clause and literal inference from compiled SBBs $COMP(B)$ is reported in Table 4.*

$COMP$	FORMULA $\mid\!\sim_{\forall}^{\mathcal{LE}}$	CLAUSE / LITERAL $\mid\!\sim_{\forall}^{\mathcal{LE}}$
Blake	Δ_2^p-complete	Δ_2^p-complete
DNF	coNP-complete	in P
Horn cover	Δ_2^p-complete	Δ_2^p-complete
renamable Horn cover	Δ_2^p-complete	Δ_2^p-complete

Table 4. Complexity of skeptical inference w.r.t. $\mathcal{LE}$ from $COMP(B)$.

Tractability is only achieved for compiled SBBs for which Δ_1 is a DNF formula and queries are restricted to CNF formulas. Especially, all the hardness results presented in Tables 3 and 4 still holds in the specific case in which the number of strata under consideration satisfies $k \geq 2$. Intractability results w.r.t. both $\mid\!\sim_{\forall}^{\mathcal{IP}}$ and $\mid\!\sim_{\forall}^{\mathcal{LE}}$ still hold when Δ_1 is a consistent Krom formula (such formulas are renamable Horn and can be turned in polynomial time into Blake normal form), or when Δ_1 is a Horn CNF formula.

Interestingly, imposing some restrictions on the literals used to name assumptions enables us to derive tractable restrictions for both CLAUSE $\mid\!\sim_{\forall}^{\mathcal{IP}}$ and CLAUSE $\mid\!\sim_{\forall}^{\mathcal{LE}}$ from a compiled SBB where Δ_1 is a Horn cover formula. Indeed, we have:

Proposition 6.4 (tractable restrictions) CLAUSE $\mid\!\sim_{\forall}^{\mathcal{IP}}$ *and* CLAUSE $\mid\!\sim_{\forall}^{\mathcal{LE}}$ *from a compiled SBB $B = (\Delta_1, \ldots, \Delta_k)$ where Δ_1 is a Horn cover formula and $\bigcup_{i=2}^{k} \Delta_i$ contains only negative literals are in* P.

Due to space limitations, we cannot give all complexity proofs[6]. So let us just focus on tractability results. Actually, in all the tractable cases listed above, $B_{\subseteq}$ can be computed in time polynomial in $|B|$ thanks to the following lemma:

Lemma 6.1 *Let $B = (\Delta_1, \ldots, \Delta_k)$ be a SBB with $\Delta = \bigcup_{i=1}^{k} \Delta_i$. We have:*

- *If $\Delta_1 = \{\alpha_1 \vee \ldots \vee \alpha_n\}$ where each α_i ($i \in 1 .. n$) is a formula from $PROP_{PS}$, then $B_{\subseteq} = \{\Delta_1 \cup (S \cap \Delta) \mid S \in max_{\subseteq}(\bigcup_{i=1}^{n}(\{\alpha_i\}, \bigcup_{j=2}^{k} \Delta_j)_{\subseteq})$.*
- *If α is a term and $\bigcup_{j=2}^{k} \Delta_j$ contains only literals, or α 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 $\{\{\alpha\} \cup \{\phi \in \bigcup_{j=2}^{k} \Delta_j \mid \alpha \not\models \neg\phi\}\}$.*

When $B = (\Delta_1, \ldots, \Delta_k)$ is s.t. Δ_1 is a DNF (resp. a Horn cover formula) and $\bigcup_{i=2}^{k} \Delta_i$ contains only literals (resp. negative literals), every element S of $B_{\subseteq}$ can be turned into a DNF (resp. a Horn cover formula) in polynomial time. Moreover, filtering out $B_{\mathcal{IP}}$ (or $B_{\mathcal{LE}}$) from $B_{\subseteq}$ can be done in polynomial time.

Since the transformation reported in Definition 5.3 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 $\mid\!\sim_{\forall}^{\mathcal{IP}}$ and CLAUSE $\mid\!\sim_{\forall}^{\mathcal{LE}}$ tractable from the compiled form. Of course, this is already achieved by only requiring Δ_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 class can prove much more compact as a representation formalism (some DNF formulas can be represented by Horn cover formulas the sizes of which are logarithmically lower but the converse does not hold[7]).

Let us ask Omer the emu for an illustration of Lemma 6.1 (Omer is an emu, every emu is a bird, normally, emus do not fly, normally, birds fly). Formally, let $B = (\Delta_1, \Delta_2, \Delta_3)$ with:

[6] Some of them are easy consequences of results reported in [10, 18, 11].

[7] For instance, the size of the smallest DNF formula equivalent to the Horn cover formula $\bigwedge_{i=1}^{m}(\neg x_2 i \vee \neg x_{2i+1})$ is $\Omega(2^m)$.

$\Delta_1 = \{emu(Omer), (emu(Omer) \Rightarrow bird(Omer))\}$,
$\Delta_2 = \{emu(Omer) \Rightarrow \neg fly(Omer)\}$, 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). B can be turned into the following compiled SBB $B' = (\Delta'_1, \Delta'_2, \Delta'_3)$ using Horn cover compilation:

$\Delta'_1 = \{(fly(Omer) \wedge emu(Omer) \wedge$
$(emu(Omer) \Rightarrow bird(Omer)) \wedge$
$(emu(Omer) \Rightarrow Emusfly(Omer))) \vee$
$(\neg fly(Omer) \wedge emu(Omer) \wedge$
$(emu(Omer) \Rightarrow bird(Omer)) \wedge$
$(bird(Omer) \Rightarrow Birdsdon'tfly(Omer))\}$
$\Delta'_2 = \{\neg Emusfly(Omer)\}$,
$\Delta'_3 = \{\neg Birdsdon'tfly(Omer)\}$.

Here, $\neg Emusfly(Omer)$ and $\neg Birdsdon'tfly(Omer)$ are the new literals used to name (uncertain) beliefs before compilation. From this compiled SBB, $B'_{\subseteq}$ can be derived in polynomial time as:

$\{\Delta'_1 \cup \{\neg Birdsdon'tfly(Omer)\}, \Delta'_1 \cup \{\neg Emusfly(Omer)\}\}$.

By construction, each of the two elements of $B'_{\subseteq}$ is a Horn CNF 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 Related Work and Conclusion

In this paper, we have shown how knowledge compilation techniques can be used to compile SBBs in order to make skeptical inference more efficient. Through a complexity analysis, 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 approach for compiling a SBB B can be favourably compared with the basic compilation approach that consists in computing $B_{\mathcal{P}}$ (reducing inference to deduction, hence making it "only" coNP-complete in the general case). Like ours, this approach 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. On the one hand, many knowledge compilation functions can be used within it (and some of them may achieve the objective of keeping the size "small enough"). On the other hand, $B_{\mathcal{P}}$ cannot be computed incrementally in the general case since removed pieces of belief can reappear later on; indeed, starting from $B_{\mathcal{P}}$ only, it is not always possible to compute the preferred subbases of a SBB B extended with a new formula. Our approach does not suffer from this drawback. In the same vein, re-partitionning[8] the SBB requires $B_{\mathcal{P}}$ to be re-computed (which is very time-consuming in general). No re-compilation is mandatory in our approach. Finally, it is obvious that, in the general case, there is no guarantee that every element of $B_{\mathcal{P}}$ is $\models$-tractable, while this is ensured by our approach.

There are many works concerned with reasoning from an inconsistent SBB, and our approach is related to many of them. Among the closer approaches is [10] which provides several complexity results for inference from SBBs (and we used some of them in our hardness proofs). This paper also gives a BDD-based algorithm for $\mid\!\sim_{\forall}^{\mathcal{LE}}$ inference; since a BDD is nothing but a compact representation of a DNF formula, Lemma 6.1 shows how such an algorithm could be extended to deal with other selection policies based on $B_{\subseteq}$. Let us finally mention [2] which presents a compilation approach for SBB. This approach consists in turning the given SBB into an equivalent one which has only one preferred subbase (not necessarily $\models$-tractable). This makes this approach complementary to ours.

[8] When designing a SBB, it is not always easy to put each piece of belief into the right stratum without making some adjustments. Hence, the capacity of re-partionning a SBB "for free" is valuable.

Acknowledgements

Many thanks to Prof. Dr. Klaus Wagner for providing us with a Δ_2^p-complete problem that can be easily reduced to LITERAL $\mid\!\sim_{\forall}^{\mathcal{LE}}$ when Δ_1 is a Blake formula. Many thanks as well to Dr. Salem Benferhat for many discussions about the compilation of SBBs.

The authors have been partly supported by a "Contrat d'objectifs de la Région Nord/Pas-de-Calais".

REFERENCES

[1] S. Benferhat, C. Cayrol, D. Dubois, J. Lang, and H. Prade, 'Inconsistency management and prioritized syntax-based entailment', in *IJCAI'93*, pp. 640–645, (1993).
[2] 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 *KR'98*, pp. 466–477, (1998).
[3] S. Benferhat, D. Dubois, and H. Prade, 'How to infer from inconsistent beliefs without revising', in *IJCAI'95*, pp. 1449–1455, (1995).
[4] Y. Boufkhad, E. Grégoire, P. Marquis, B. Mazure, and L. Saïs, 'Tractable cover compilations', in *IJCAI'97*, pp. 122–127, (1997).
[5] G. Brewka, 'Preferred subtheories: an extended logical framework for default reasoning', in *IJCAI'89*, pp. 1043–1048, (1989).
[6] M. Cadoli and F.M. Donini, 'A survey on knowledge compilation', *AI Communications*, **10**, 137–150, (1997). printed in 1998.
[7] M. Cadoli, F.M. Donini, and M. Schaerf, 'Is intractability of non-monotonic reasoning a real drawback?', *Artificial Intelligence*, **88**, 215–251, (1996).
[8] C. Cayrol and M.-C. Lagasquie-Schiex, 'On the complexity of non-monotonic entailment in syntax-based approaches', in *Proc. of the ECAI'94 Workshop on Algorithms, Complexity and Commonsense Reasoning*, (1994).
[9] C. Cayrol and M.-C. Lagasquie-Schiex, 'Non-monotonic syntax-based entailment: a classification of consequence relations', in *ECSQARU'95*, pp. 107–114, (1995).
[10] C. Cayrol, M.-C. Lagasquie-Schiex, and Th. Schiex, 'Nonmonotonic reasoning: from complexity to algorithms', *Annals of Mathematics and Artificial Intelligence*, **22**, (1998).
[11] S. Coste-Marquis and P. Marquis, 'Complexity results for propositional closed world reasoning and circumscription from tractable knowledge bases', in *IJCAI'99*, pp. 24–29, (1999).
[12] A. Darwiche, 'Compiling devices into decomposable negation normal form', in *IJCAI'99*, pp. 284–289, (1999).
[13] A. del Val, 'Tractable databases: how to make propositional unit resolution complete through compilation', in *KR'94*, pp. 551–561, (1994).
[14] P. Liberatore, 'On the compilability of diagnosis, planning, reasoning about actions, belief revision, etc.', in *KR'98*, pp. 144–155, (1998).
[15] P. Marquis, 'Knowledge compilation using theory prime implicates', in *IJCAI'95*, pp. 837–843, (1995).
[16] B. Nebel, 'Belief revision and default reasoning: syntax-based approaches', in *KR'91*, pp. 417–428, (1991).
[17] B. Nebel, 'Base revision operations and schemes: semantics, representation and complexity', in *ECAI'94*, pp. 341–345, (1994).
[18] B. Nebel, *Belief Revision*, volume 3 of *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, chapter How hard is it to revise a belief base?, 77–145, Kluwer Academic, 1998.
[19] Ch. Papadimitriou, *Computational complexity*, Addison–Wesley, 1994.
[20] G. Pinkas and R.P. Loui, 'Reasoning from inconsistency: a taxonomy of principles for resolving conflict', in *KR'92*, pp. 709–719, (1992).
[21] R. Reiter and J. de Kleer, 'Foundations of assumption-based truth maintenance systems: preliminary report', in *AAAI'87*, pp. 183–188, (1987).
[22] R. Schrag, 'Compilation for critically constrained knowledge bases', in *AAAI'96*, pp. 510–515, (1996).
[23] B. Selman and H.A. Kautz, 'Knowledge compilation and theory approximation', *Journal of the Association for Computing Machinery*, **43**, 193–224, (1996).