

Compiling propositional weighted bases[☆]

Adnan Darwiche^a, Pierre Marquis^{b,*}

^a Computer Science Department, University of California, Los Angeles, CA 90095, USA

^b CRIL-CNRS/Université d'Artois, rue de l'Université, S.P. 16, 62307 Lens, France

Received 28 December 2002; accepted 14 April 2004

Abstract

In this paper, we investigate the extent to which knowledge compilation can be used to improve model checking and inference from propositional weighted bases. We first focus on the compilability issue for both problems, deriving mainly non-compilability results in the case preferences are subject to change. Then, we present a general notion of *C*-normal weighted base that is parametrized by a tractable class *C* for the clausal entailment problem. We show how every weighted base can be turned (“compiled”) into a query-equivalent *C*-normal base whenever *C* is a complete class for propositional logic. Both negative and positive results are presented. On the one hand, complexity results are identified, showing that the inference problem from a *C*-normal weighted base is as difficult as in the general case, when the prime implicates, Horn cover or renamable Horn cover classes are targeted. On the other hand, we show that both the model checking and the (clausal) inference problem become tractable whenever *DNNF*-normal bases are considered. Moreover, we show that the set of all preferred models of a *DNNF*-normal weighted base can be computed in time polynomial in the output size, and as a consequence, model checking is also tractable for such bases. Finally, we sketch how our results can be used in model-based diagnosis in order to compute the most likely diagnoses of a system.

© 2004 Published by Elsevier B.V.

Keywords: Knowledge representation; Belief bases; Penalty logic; Knowledge compilation

[☆] This is an extended and revised version of a paper that appeared in the Proceedings of the 9th International Workshop on Non-Monotonic Reasoning (NMR'02), Toulouse, France, 2002, pp. 6–14.

* Corresponding author.

E-mail addresses: darwiche@cs.ucla.edu (A. Darwiche), marquis@cril.univ-artois.fr (P. Marquis).

1. Introduction

Penalty logic is a logical framework developed by Pinkas [45,46] and by Dupin de St Cyr, Lang and Schiex [27]. It enables the representation of propositional weighted bases. A *weighted base* is a finite set

$$W = \{\langle \phi_1, k_1 \rangle, \dots, \langle \phi_n, k_n \rangle\}.$$

Each ϕ_i is a propositional sentence, and k_i is its corresponding weight, i.e., the price to be paid if the sentence is violated. In penalty logic, weights are positive integers¹ or $+\infty$ and they are *additively* aggregated.

A weighted base can be considered as a compact, implicit encoding of a total pre-ordering over a set Ω of propositional worlds. Indeed, given a weighted base W , the weight of each world ω can be defined as follows:

$$K_W(\omega) \stackrel{\text{def}}{=} \sum_{\langle \phi_i, k_i \rangle \in W, \omega \models \neg \phi_i} k_i.$$

That is, the weight of a world is the sum of all weights associated with sentences violated by the world. One can extend the weight function K_W to arbitrary sentences α :

$$K_W(\alpha) \stackrel{\text{def}}{=} \min_{\omega \models \alpha} K_W(\omega).$$

Finally, for every set S of worlds, $\min_W(S)$ denotes the most preferred worlds in S , i.e., those having minimal weight:

$$\min_W(S) \stackrel{\text{def}}{=} \{\omega \mid \omega \in S, \forall \omega' \in S, K_W(\omega) \leq K_W(\omega')\}.$$

The weight of a base W , denoted $K(W)$, is the weight of the worlds in $\min_W(\Omega)$. Obviously enough, we have $K(W) = K_W(\text{true})$, and $\omega \in \min_W(\Omega)$ if and only if $K_W(\omega) = K(W)$.

Example 1.1. Let $W = \{\langle a \wedge b, 2 \rangle, \langle \neg b, 1 \rangle\}$ be a weighted base. Let us consider the following four worlds over the variables appearing in W , $\text{Var}(W)$:

- $\omega_1 = (a, b)$;
- $\omega_2 = (a, \neg b)$;
- $\omega_3 = (\neg a, b)$;
- $\omega_4 = (\neg a, \neg b)$.

We then have $K_W(\omega_1) = 1$, $K_W(\omega_2) = 2$, $K_W(\omega_3) = 3$, and $K_W(\omega_4) = 2$. Accordingly, we have $K(W) = 1$ and $\min_W(\Omega) = \{\omega_1\}$.

All sentences ϕ_i associated with finite weights in a weighted base are called *soft constraints*, while those associated with the weight $+\infty$ are called *hard constraints*.

¹ Floating numbers can also be used; what is important is the fact that sum is a total function over the set of (totally ordered) numbers under consideration, and that it can be computed in polynomial time.

As it is the case for many logic-based representation formalisms, we are typically interested in two main decision problems, MODEL CHECKING and (CLAUSAL) INFERENCE,² defined as follows:

- MODEL CHECKING
 - **Input:** a weighted base W and a world ω .
 - **Question:** is ω a preferred world given W , i.e., is $K_W(\omega) = K(W)$?
- (CLAUSAL) INFERENCE
 - **Input:** a weighted base W and a CNF sentence α .
 - **Question:** is α a consequence of W (noted $\vdash_W \alpha$), i.e., is every preferred world from $\min_W(\Omega)$ a model of α ?

Penalty logic has some valuable connections with possibilistic logic, as well as with Dempster–Shafer theory (see [27] for details) and Kappa Calculus. In Kappa Calculus, one has a function κ which maps every world ω into an ordinal $\kappa(\omega)$ [23,53]. The kappa function is extended to propositional sentences α using $\kappa(\alpha) = \min_{\omega \models \alpha} \kappa(\omega)$. One way to construct kappa functions is by using a belief network, which is a directed acyclic graph over propositional symbols [17,32]. For every instantiation α of a network variable, and every instantiation β of its parents, we provide a value κ for the pair (α, β) , which represents a penalty for a world that satisfies $\alpha \wedge \beta$. This is why the kappa value is called a *degree of surprise* in this case. The kappa value of a world is then the addition of all degrees of surprise contributed by the network variables. Kappa functions constructed in this fashion satisfy some very interesting properties, most of which can be revealed by examining the belief network topology.

Penalty logic is also closely connected to the optimization problem WEIGHTED-MAX-SAT considered in operations research (see, e.g., [6,33,52]). Indeed, the input of the (function) problem WEIGHTED-MAX-SAT is a weighted base W in which every weighted sentence ϕ_i is restricted to be a clause, and the output of WEIGHTED-MAX-SAT is any element of $\min_W(\Omega)$.

Several proposals for the use of weighted bases can be found in the AI literature. One of them concerns the compact representation of preferences in a decision making setting. Indeed, in some decision making problems, models (and sentences) can be used to encode decisions. Hard constraints are used to characterize the set of alternatives (possible decisions), while the soft ones enable one to encode preferences, and the weight of a model represents the disutility of a decision, and a weighted base can be viewed as an implicit representation of the set of all decisions of an agent, totally ordered w.r.t. their (dis)utility. Lafage and Lang [35] take advantage of such an encoding for group decision making. A key issue here from a computational point of view is the problem consisting in determining whether a given world ω (encoding a decision) is undominated, i.e., it is an element of $\min_W(\Omega)$; as introduced before, this is just the model checking problem for penalty logic.

² In this paper, inference is to be considered as a short for clausal inference: by default, we assume that queries are CNF sentences. The literal inference problem is the restriction of (CLAUSAL) INFERENCE, where queries are conjunctions of literals.

Of course, the corresponding function and enumeration problems (computing one versus all most preferred world(s)) are also of major interest in such a setting.

Another suggested use of penalty logic concerns inference from inconsistent belief bases. Here, hard constraints are used to encode pieces of knowledge (i.e., beliefs that must be true), while soft constraints are used to represent more or less uncertain pieces of beliefs. Based on the preference information given by K_W , several inference relations from a weighted base W can be defined. Among them is skeptical inference where $\alpha \vdash_W \beta$ if and only if every world ω that is of minimal weight among the models of α is a model of β . In this framework, propositional sentences represent pieces of (explicit) belief. The inference relation $\vdash_W$ is interesting for at least two reasons. On the one hand, it is a *comparative* inference relation, i.e., a rational inference relation satisfying supraclassicality [27]. On the other hand, weighted bases can be used to encode some well-known forms of inference from stratified belief bases $B = (B_1, \dots, B_k)$ [2,4,47]. In particular, the so-called skeptical *lexicographic inference* $B \vdash_{lex}$ can be recovered as a specific case of $true \vdash_{W_B}$ for some weighted base W_B .

Example 1.2. Let $B = (B_1, B_2)$ be a stratified belief base skeptically interpreted under the lexicographic policy, where $B_1 = \{a \vee b \vee c\}$ (the most reliable stratum) and $B_2 = \{\neg a \wedge c, \neg b \wedge c, \neg c\}$. We can associate with B the weighted base

$$W_B = \{\langle a \vee b \vee c, 4 \rangle, \langle \neg a \wedge c, 1 \rangle, \langle \neg b \wedge c, 1 \rangle, \langle \neg c, 1 \rangle\}.$$

The unique most preferred world for W_B is $(\neg a, \neg b, c)$ that is also the only lexicographically-preferred model of B .

Weighted bases enable more flexibility than stratified belief bases. For example, violating two sentences of weight 5 is worse than violating a single sentence of weight 9, but this cannot be achieved through a simple stratification.³

Up to now, weighted bases have been investigated from a theoretical point of view only. Despite their promise, we are not aware of any industrial application of weighted bases, except the one reported in [43,44] (discussed in Section 7) which illustrates how interesting such a notion can be from the application point of view. There is a simple (but partial) explanation of this fact: MODEL CHECKING and INFERENCE from weighted bases are intractable. On the one hand, it is not hard to prove that the model checking problem is coNP-complete even if the weighted base contains two weighted sentences, none of them being a hard constraint. On the other hand, the inference problem is known as Δ_2^P -complete [26], even in the restricted case queries are literals. Furthermore, it is not hard to show that computing a preferred world from $\min_W(\Omega)$ is $F\Delta_2^P$ -complete. This implies that any of the two problems is very likely to require an unbounded polynomial number of calls to an NP oracle to be solved in polynomial time on a deterministic Turing machine.

In this paper, we investigate the extent to which *knowledge compilation* [8] can be used to improve model checking and inference from weighted bases. The key idea of

³ Since lexicographic inference also includes inference from consistent subbases that are maximal w.r.t. cardinality as a subcase (to achieve it, just put every sentence of the belief base into a single stratum), the latter can also be recovered as a specific case of inference from a weighted base.

compilation is pre-processing the fixed part of the decision problem under consideration (the one that does not change frequently) so as to improve on-line complexity. Existing work on knowledge compilation can be roughly partitioned into two classes, the one gathering results on compilability (most of them are from Cadoli and his colleagues) and the other one gathering compilation functions, typically aiming at improving the (clausal) inference problem for classical logic from the practical side.

Roughly speaking, a decision problem is said to be compilable to a given complexity class C if it is in C once the fixed part of any instance has been pre-processed, i.e., turned off-line into a poly-size data structure. The fact that the pre-processing must be achieved in polynomial space is crucial. In order to formalize such a notion of compilability, Cadoli and his colleagues introduced many new classes (compilability classes) and the corresponding reductions (see mainly [9–12,37]). This enables one to classify many AI problems as compilable to a class C , or as not compilable to C (usually under standard assumptions of complexity theory—the fact that the polynomial hierarchy PH does not collapse). Thus, the (clausal) inference problem for classical logic is known as non-compilable to P unless PH collapses.

Because this negative result concerns the worst case only, it does not necessarily prevent knowledge compilation from being practically useful in order to improve clausal entailment. Accordingly, many knowledge compilation functions dedicated to the clausal entailment problem have been pointed out so far (e.g., [7,16,19,29,30,39,49–51]). In these approaches, the input sentence is turned into a compiled one during an off-line compilation phase and the compiled form is used to answer the queries on-line. Assuming that the sentence does not often change and that answering queries from the compiled form is computationally easier than answering them from the input sentence, the compilation time can be balanced over a sufficient number of queries. Thus, the complexity of classical inference falls from coNP-complete to P under the restrictions that clausal queries are considered and the input sentence has been compiled. While none of the techniques listed above can ensure the objective of enhancing inference to be reached in the worst case (because the size of the compiled form can be exponentially larger than the size of the original sentence—this coheres with the fact that the clausal entailment problem is not compilable to P [8,51]), experiments have shown such approaches valuable in many practical situations [7,22,50].

In the following, we consider both aspects of knowledge compilation for penalty logic: the compilability issue and the design of compilation functions for both the model checking and the inference problems.

On the one hand, we show that the complexity of *MODEL CHECKING* (respectively *INFERENCE*) can be reduced to P (respectively coNP) through pre-processing given that the preferences (weights) are available at the off-line stage, and that such compilability results do not hold any longer (under the standard assumptions of complexity theory) when preferences belong to the varying part of the problem. Thus, the unique problem among those considered here that can be rendered tractable (i.e., in P) through compilation is *MODEL CHECKING* assuming that the weights do not change with time.

On the other hand, we show how compilation functions for clausal entailment from classical sentences can be extended to clausal inference from weighted bases. Any equivalence-preserving knowledge compilation function can be considered in our framework. Interestingly, the corresponding notion of compiled base is flexible w.r.t.

preference handling in the sense that re-compiling a weighted base is useless whenever the weights associated to soft constraints change with time. Unfortunately, for many target classes C for such functions, including the prime implicates, Horn cover and renamable Horn cover classes, we show that the inference problem from a C -normal base remains Δ_2^P -complete, even for very simple queries (literals). Accordingly, in this situation, there is no guarantee that compiling a weighted base using any of the corresponding compilation functions may help. Then we focus on *DNNF*-normal bases, considering the *DNNF* class introduced in [19,21]. This case is much more favourable since both the model checking problem and the clausal inference problem become tractable. More, we show that the preferred models of a *DNNF*-normal weighted base can be enumerated in output polynomial time.

Finally, we sketch how our results can be used in the model-based diagnosis framework in order to compute the most likely diagnoses of a system.

2. Formal preliminaries

In the following, we consider a propositional language $PROP_{PS}$ defined inductively from a finite set PS of propositional symbols, the boolean constants *true* and *false* and the standard connectives in the usual way. L_{PS} is the set of literals built up from PS . If l is a positive literal $x \in PS$, then its complementary literal $\sim l$ is $\neg x$; if l is a negative literal $\neg x$, then $\sim l$ is x . For every sentence ϕ from $PROP_{PS}$, $Var(\phi)$ denotes the symbols of PS occurring in ϕ . As mentioned before, if $W = \{\langle \phi_1, k_1 \rangle, \dots, \langle \phi_n, k_n \rangle\}$ is a weighted base, then $Var(W) = \bigcup_{i=1}^n Var(\phi_i)$. The size $|\phi|$ of a sentence ϕ is the number of occurrences of propositional symbols and connectives used to write ϕ . Numbers k are represented in binary notation and $+\infty$ is a specific symbol, so that the size $|k|$ of any finite number k is the number of binary digits used to write it, and the size of $+\infty$ is 1. The size $|W|$ of a weighted base $W = \{\langle \phi_1, k_1 \rangle, \dots, \langle \phi_n, k_n \rangle\}$ is then $\sum_{i=1}^n |\phi_i| + |k_i|$.

Sentences are interpreted in a classical way. Ω denotes the set of all interpretations built up from PS . Every interpretation (world) $\omega \in \Omega$ is represented as a tuple of literals. $\models$ denotes classical entailment and $\equiv$ denotes logical equivalence. $Mod(\phi)$ is the set of all models of ϕ ; that is, worlds in Ω that satisfy ϕ .

As usual, every finite set of sentences is considered as the conjunctive sentence whose conjuncts are the elements of the set. A *CNF* sentence is a (finite) conjunction of clauses, where a clause is a (finite) disjunction of literals. A tractable class for clausal entailment is a subset C of propositional sentences⁴ whose clausal consequences can be decided in polynomial time. There are many such classes, including the *Blake* one, the *Horn CNF* one, the *renamable Horn CNF* one, the *DNNF* one. A sentence ϕ is *Blake* if and only if it is a *CNF* sentence where each prime implicate⁵ of ϕ appears as a conjunct (one representative per equivalence class). A sentence is *Horn CNF* if and only if it is a *CNF* sentence s.t. every clause in it contains at most one positive literal. A sentence ϕ is *renamable Horn CNF* if

⁴ We do not make any distinction here between the “flat” classes and the “nested” ones, as in [25].

⁵ A *prime implicate* π of a sentence ϕ is one of the logically strongest clauses entailed by ϕ , i.e., we have $\phi \models \pi$ and for every clause π' s.t. $\phi \models \pi'$ and $\pi' \models \pi$, we also have $\pi \equiv \pi'$.

and only if $\sigma(\phi)$ is a *Horn CNF* sentence, where σ is a substitution from L_{PS} to L_{PS} s.t. $\sigma(l) = l$ for every literal l of L_{PS} except those of a set L , and for every literal l of L , $\sigma(l) = \sim l$ and $\sigma(\sim l) = l$.

We assume that the reader familiar with the complexity classes P, NP, coNP and Δ_2^P of the polynomial hierarchy. $F\Delta_2^P$ denotes the class of function problems associated to Δ_2^P ; see [42] for details.

3. Compilability results for model checking and inference from weighted bases

Before considering any specific compilation approach for weighted bases, it is important to identify the feasibility of improving inference (and model checking) through pre-processing, i.e., to determine whether or not the decision problem under consideration is compilable. Indeed, a non-compilability result shows that whatever the compilation approach, no computational gain is to be expected in the worst case from pre-processing. Hence, one can either abandon the compilation approach, or develop compilation functions even though they may lead to compiled forms of weighted bases that are exponentially larger than the original bases.

In this section, we investigate the compilability of model checking and inference from weighted bases. We first give a few definitions and then report our results.

3.1. Some definitions

Let us first make precise what “compilable to C” means, recalling some key definitions proposed by Cadoli and his colleagues (many more definitions and results about compilability can be found in [11,37]).

First of all, in order to address the compilability of a decision problem, we need to consider it as a language of pairs $\langle x, y \rangle$: the *fixed part* x will be subject to pre-processing, while the remaining *varying part* y will not. For instance, considering MODEL CHECKING (respectively INFERENCE), a standard partition consists in taking W as the fixed part and ω (respectively α) as the varying one; this just reflects the fact that the base typically changes less often than the queries. Accordingly, the decision problems under consideration are represented as *languages of pairs* of strings $\langle x, y \rangle$.

While several families of classes can be considered as candidates to represent what “compilable to C” means, the most general one gathers the nu-compC classes [11]. Thus, “compilable to C” is formalized as membership to the compilability class nu-compC:

Definition 3.1 (nu-compC). Let C be a complexity class closed under polynomial many-one reductions and admitting complete problems for such reductions. A language of pairs L belongs to nu-compC if and only if there exists a binary poly-size function⁶ f and a language of pairs $L' \in C$ such that for all $\langle x, y \rangle$, we have:

$$\langle x, y \rangle \in L \quad \text{if and only if} \quad \langle f(x, |y|), y \rangle \in L'.$$

⁶ A function f is *poly-size* whenever there exists a polynomial p s.t. for all strings x it holds $|f(x)| \leq p(|x|)$.

Here “nu” stands for “non-uniformly”, which indicates that the compiled form of x may also depend on the size of the varying part y . As for usual complexity classes, the most difficult problems w.r.t. a compilability class nu-compC are those to which any problem from nu-compC can be reduced. The right notion of reduction for such compilability classes is the $\leq_{\text{nu-comp-one}}$ [11]:

Definition 3.2 ($\leq_{\text{nu-comp}}$ reduction). A $\leq_{\text{nu-comp}}$ reduction from a language of pairs L to a language of pairs L' is a triple $\langle f_1, f_2, g \rangle$ where f_1 and f_2 are binary poly-size functions and g is a binary polynomial function s.t. for any pair of strings $\langle x, y \rangle$, we have:

$$\langle x, y \rangle \in L \quad \text{if and only if} \quad \langle f_1(x, |y|), g(f_2(x, |y|), y) \rangle \in L'.$$

Inclusion of compilability classes similar to those holding in the polynomial hierarchy exist (see [11]). It is also strongly believed that the compilability hierarchy is proper: if it collapses, then the polynomial hierarchy collapses at well (cf. Theorem 2.12 from [11]). For instance, if the clausal entailment problem (that is $\text{nu-compcoNP-complete}$) is in nu-compP , then the polynomial hierarchy collapses at the third level.

3.2. Compilability results

Let us first consider the case W is the fixed part of the compilation problem, while ω (respectively α) is the varying part. We have identified the following compilability results:

Proposition 3.1.

- MODEL CHECKING with fixed W and varying ω is in nu-compP .
- INFERENCE with fixed W and varying α is $\text{nu-compcoNP-complete}$.

Proof.

- MODEL CHECKING. It is sufficient to compute off-line $K(W)$ and to store it (formally, we define $f(W, |\omega|)$ as $K(W)$ —which is independent of $|\omega|$); at the on-line stage, given a world $\omega \in \Omega$, we compute $K_W(\omega)$ in (deterministic) polynomial time and we compare it to the stored number $K(W)$: ω is a preferred world if and only if $K_W(\omega) = K(W)$.
- INFERENCE
 - Membership: in order to show that INFERENCE is in nu-compcoNP , it is sufficient to exhibit a propositional sentence $f(W, |\alpha|) = \Sigma$ of size polynomial in $|W|$ that is query equivalent to W . Such a sentence Σ can be computed (off-line) from $W = \{\langle \phi_1, k_1 \rangle, \dots, \langle \phi_n, k_n \rangle\}$ as follows:
 - (1) check whether $\widehat{W} = \bigwedge_{\langle \phi_i, +\infty \rangle \in W} \phi_i$ is inconsistent;
 - if it is the case, we have $K(W) = +\infty$, hence every world is preferred: set Σ to *true*, and goto (4);
 - otherwise, remove each pair $\langle \phi_i, +\infty \rangle$ from W to obtain a new base W' ;

- (2) compute $K(W')$ for $W' = \{\langle \phi_1, k_1 \rangle, \dots, \langle \phi_m, k_m \rangle\}$;
- (3) generate the sentence

$$\Sigma = \widehat{W} \wedge \bigwedge_{i=1}^m (\text{holds}_i \Rightarrow \phi_i) \wedge \text{PREF}(\text{holds}_1, \dots, \text{holds}_m);$$

- (4) return Σ .

At step (1), it is obvious that every world is preferred when $\widehat{W}$ is inconsistent; in the remaining case, every preferred world is a model of $\widehat{W}$ and it must be preferred given the weighted base obtained by removing each pair $\langle \phi_i, +\infty \rangle$ from W . Accordingly, the value $K(W')$ computed at step (2) coincides with $K(W)$ whenever $\widehat{W}$ is consistent. At step (3), $\text{PREF}(\text{holds}_1, \dots, \text{holds}_m)$ is a sentence encoding a boolean function depending on $\text{holds}_1, \dots, \text{holds}_m$ that evaluates to true if and only if $\sum_{i=1, \dots, m | \text{holds}_i = \text{false}} k_i = K(W')$. Indeed, the assignments of truth values to $\text{holds}_1, \dots, \text{holds}_m$ that make $\text{PREF}(\text{holds}_1, \dots, \text{holds}_m)$ true characterize exactly the preferred subbases of W' , i.e., those subsets of W' whose constraints are satisfied by a preferred world (see Corollary 7.1 from [27]). The last point is that a polyspace sentence $\text{PREF}(\text{holds}_1, \dots, \text{holds}_m)$ can be generated by combining adders and a comparator (see [9] for a similar proof).

- As to hardness, in order to show that INFERENCE is nu-compcoNP-hard, it is sufficient to observe that clausal entailment from a propositional sentence Σ is a specific case of inference from a weighted base $\{\langle \Sigma, 1 \rangle\}$, and to take advantage of known results showing that clausal entailment is nu-compcoNP-complete (see Theorem 2.10 from [11]). $\square$

It can be observed that our proofs actually show MODEL CHECKING belonging to compP and INFERENCE belonging to compcoNP. Note also that compP (respectively compcoNP) is a subset of nu-compP (respectively nu-compcoNP); see [11] for details.

These results show that compilation can prove helpful when preferences are fixed since the on-line complexities of MODEL CHECKING and of INFERENCE are reduced (from coNP to P for MODEL CHECKING, and from Δ_2^P to coNP for INFERENCE). However, it is very unlikely that INFERENCE can be rendered tractable through poly-size pre-processing.

In some situations, the constraints encoded by sentences from weighted bases are shared by a number of agents, while each agent has her own preferences; hence constraints (at least the soft ones) may have different weights, depending on the agent. May pre-processing help in such a situation? In order to address this issue in formal terms, we need to keep the constraints apart from the corresponding penalties, i.e., to identify every weighted base $W = \{\langle \phi_1, k_1 \rangle, \dots, \langle \phi_n, k_n \rangle\}$ with the pair $\langle C, P \rangle$, where C is the n -vector of sentences (constraints) $\langle \phi_1, \dots, \phi_n \rangle$ and P is the n -vector of penalties (positive integers or $+\infty$ represented as a specific symbol) $\langle k_1, \dots, k_n \rangle$. Obviously, every W can be associated to such a pair $\langle C, P \rangle$ in polynomial time, and the converse also holds.

This time, the compilability of MODEL CHECKING (respectively INFERENCE) is considered for the language of pairs $\langle C, \langle P, \omega \rangle \rangle$ (respectively $\langle C, \langle P, \alpha \rangle \rangle$). We have identified the following results:

Proposition 3.2.

- MODEL CHECKING with fixed C and varying $\langle P, \omega \rangle$ is nu-compcoNP-complete.
- INFERENCE with fixed C and varying $\langle P, \alpha \rangle$ is nu-comp Δ_2^P -complete.

Proof.

- MODEL CHECKING
 - Membership is a direct consequence of the fact that the original model checking problem (where the whole input is variable) is in coNP, and that $\text{coNP} \subset \text{nu-compcoNP}$ (see Theorem 2.13 from [11]).
 - As to hardness, we reduce the $\star 3\text{UNSAT}$ problem to MODEL CHECKING. $\star 3\text{UNSAT}$ is defined as $\bigcup_{n \geq 0} \star 3\text{UNSAT}_n$, and the language of $\star 3\text{UNSAT}_n$ contains all pairs $\langle \star, \pi_n \rangle$ where π_n belongs to 3UNSAT_n , i.e., it is an inconsistent 3-CNF sentence generated from the variables $x_1, \dots, x_n$ (a 3-CNF sentence is a finite conjunction of 3-clauses, where a 3-clause contains at most 3 literals). $\star$ stands here for any string. It is known that $\star 3\text{UNSAT}$ is nu-compcoNP-hard (a direct consequence of Theorem 2.13 from [37]).

Let Γ_n be the CNF sentence containing all the clauses of the form $\neg \text{holds}_i \vee \gamma_i$ where γ_i is a clause with at most 3 literals built up from variables $x_1, \dots, x_n$ and each holds_i is a new symbol (one for each clause γ_i). Let p be the number of clauses in Γ_n ; we have $p \in O(n^3)$, hence the size of Γ_n is polynomial in n .

To each pair $\langle \star, \pi_n \rangle$ we associate the $p + 2$ -vector of constraints

$$C_n = \langle \text{holds}_1, \dots, \text{holds}_p, \Gamma_n \vee \neg \text{dom}, \text{dom} \rangle,$$

where dom is a new variable (intuitively, dom means “dominated”, i.e., non-preferred). Clearly enough, C_n depends only on n .

Let π_n be any instance of 3UNSAT_n . Each π_n corresponds to an assignment $\text{holds}_1^*, \dots, \text{holds}_p^*$ of truth values to $\text{holds}_1, \dots, \text{holds}_p$ s.t. holds_i is evaluated to true (i.e., $\text{holds}_i^* = \text{true}$) if and only if γ_i belongs to π_n . Obviously, Γ_n and π_n are equivalent under the (partial) assignment $\text{holds}_1^*, \dots, \text{holds}_p^*$.

Now, each π_n can be associated to a pair $\langle P_{\pi_n}, \omega_{\pi_n} \rangle$ s.t.

For every $i \in 1, \dots, p$, $P_{\pi_n}[i] = 3p + 3$ if γ_i belongs to π_n , and $P_{\pi_n}[i] = 1$ otherwise, and $P_{\pi_n}[p + 1] = P_{\pi_n}[p + 2] = p + 1$.

For every $i \in 1, \dots, n$, $\omega_{\pi_n}(x_i) = \text{false}$, $\omega_{\pi_n}(\text{dom}) = \text{false}$, and for every $i \in 1, \dots, p$, $\omega_{\pi_n}(\text{holds}_i) = \text{true}$ if and only if γ_i belongs to π_n .

The weights are set in such a way every constraint holds_i is among the most prioritary ones whenever γ_i belongs to π_n , and among the less prioritary ones in the remaining case. The two other constraints $\Gamma_n \vee \neg \text{dom}$ and dom are put in an intermediate stratum (no compensation between strata can be achieved here).

By construction, every preferred model ω of $\langle C_n, P_{\pi_n} \rangle$ is such that for every $i \in 1, \dots, p$, $\omega(\text{holds}_i) = \text{true}$ if and only if γ_i belongs to π_n . Now, if π_n is inconsistent, then $\Gamma_n \vee \neg \text{dom}$ is equivalent to $\neg \text{dom}$ under the partial assignment $\text{holds}_1^*, \dots, \text{holds}_p^*$ of truth values associated to π_n , and since $\Gamma_n \vee \neg \text{dom}$ and dom are associated to the same weight, ω_{π_n} is undominated (the world that

coincides with ω_{π_n} except for variable dom has the same weight as ω_{π_n}). Otherwise, π_n is consistent, so it has a model ω over $x_1, \dots, x_n$. Let us consider the extension of ω over $x_1, \dots, x_n, holds_1, \dots, holds_p, dom$ s.t. for every $i \in 1, \dots, p$, $\omega(holds_i) = true$ if and only if γ_i belongs to π_n , and $\omega(dom) = true$. It is clear that $K_{(C_n, P_{\pi_n})}(\omega) < K_{(C_n, P_{\pi_n})}(\omega_{\pi_n})$, hence ω_{π_n} is dominated. Thus, $(\star, \pi_n)$ is a positive instance of $\star 3UNSAT_n$ if and only if $\langle C_n, \langle P_{\pi_n}, \omega_{\pi_n} \rangle \rangle$ is a positive instance of MODEL CHECKING (with fixed C and varying $\langle P, \alpha \rangle$), and this completes the proof.

- INFERENCE

- Membership is a direct consequence of the fact that the original inference problem (where the whole input is variable) is in Δ_2^P , and that $\Delta_2^P \subset \text{nu-comp}\Delta_2^P$ (see Theorem 2.13 from [11]).

- As to hardness, we reduce the $\star MAX\text{-SAT-}ASG_{\text{odd}}$ problem to INFERENCE.

We start from the $MAX\text{-SAT-}ASG_{\text{odd}}$ problem as defined in [54]. The set of positive instances of this decision problem consists of all propositional sentences Σ where Σ is a consistent 3-*CNF* sentence generated from the variables $x_1, \dots, x_n$ s.t. the model ω_{max} of Σ that is maximal w.r.t the lexicographic ordering induced by $x_1 < \dots < x_n$ satisfies $\omega_{\text{max}}(x_n) = true$. We first show that the restriction of $MAX\text{-SAT-}ASG_{\text{odd}}$ where Σ is a 3-*CNF* formula remains Δ_2^P -complete. Here is a reduction close to the one typically used to show that 3SAT is NP-hard, starting from the general problem (i.e., the satisfiability problem for unconstrained propositional formulas). Given a propositional formula Σ containing m occurrences of a connective, we introduce a new variable y per occurrence and we generate in polynomial time a 3-*CNF* formula Σ' encoding the corresponding equivalences. For instance, to $\Sigma = ((x_1 \Rightarrow_1 x_2) \vee_2 \neg_3 x_3) \Rightarrow_4 x_4$, we first associate the following conjunction of equivalences: $(y_1 \Leftrightarrow (x_1 \Rightarrow x_2)) \wedge (y_3 \Leftrightarrow (\neg x_3)) \wedge (y_2 \Leftrightarrow (y_1 \vee y_3)) \wedge (y_4 \Leftrightarrow (y_2 \Rightarrow x_4))$. The subscripts associated to each occurrence i of a connective are just used to indicate the corresponding variable y_i . Then, we turn such a conjunction of equivalences into a 3-*CNF* formula $\Sigma' = (\neg y_1 \vee \neg x_1 \vee x_2) \wedge (x_1 \vee y_1) \wedge (\neg x_2 \vee y_1) \wedge (\neg y_3 \vee \neg x_3) \wedge (y_3 \vee x_3) \wedge (\neg y_2 \vee y_1 \vee y_3) \wedge (\neg y_1 \vee y_2) \wedge (\neg y_3 \vee y_2) \wedge (\neg y_4 \vee \neg y_2 \vee x_4) \wedge (y_2 \vee y_4) \wedge (\neg x_4 \vee y_4)$. Σ' can be computed in time polynomial in the size of Σ . It is easy to show that every model ω of Σ over $\text{Var}(\Sigma)$ is extended by a unique model ω' of Σ' over $\text{Var}(\Sigma) \cup \{y_1, \dots, y_m\}$: every new variable y_i ($i \in 1, \dots, m$) is defined in Σ' from $\text{Var}(\Sigma) = \{x_1, \dots, x_n\}$ [36]. Now, to every instance $\langle \Sigma, x_1 < \dots < x_n \rangle$ of $MAX\text{-SAT-}ASG_{\text{odd}}$, we associate in polynomial time the instance $\langle \Sigma' \wedge (\neg x_n \vee y_{m+1}) \wedge (x_n \vee \neg y_{m+1}), x_1 < \dots < x_n < y_1 < \dots < y_m < y_{m+1} \rangle$ of its restriction to the case the formula is a 3-*CNF* one. By construction, the model ω_{max} of Σ over $\text{Var}(\Sigma)$ that is maximal w.r.t the lexicographic ordering induced by $x_1 < \dots < x_n$ is s.t. $\omega_{\text{max}}(x_n) = true$ if and only if the model ω'_{max} of Σ' over $\text{Var}(\Sigma')$ that is maximal w.r.t the lexicographic ordering induced by $x_1 < \dots < x_n < y_1 < \dots < y_m < y_{m+1}$ is s.t. $\omega'_{\text{max}}(y_{m+1}) = true$.

We now define $\star MAX\text{-SAT-}ASG_{\text{odd}} = \bigcup_{n \geq 0} \star MAX\text{-SAT-}ASG_{\text{odd}_n}$, where the language of $\star MAX\text{-SAT-}ASG_{\text{odd}_n}$ contains all pairs $\langle \star, \pi_n \rangle$ where π_n belongs to $MAX\text{-SAT-}ASG_{\text{odd}_n}$, i.e., it is a consistent 3-*CNF* sentence generated from the variables $x_1, \dots, x_n$ s.t. the model ω_{max} of π_n that is maximal w.r.t the lexicographic

ordering induced by $x_1 < \dots < x_n$ satisfies $\omega_{\max}(x_n) = \text{true}$. It is known that $\star\text{MAX-SAT-ASG}_{\text{odd}}$ is nu-comp Δ_2^p -hard (a direct consequence of Theorem 2.13 from [37] since the corresponding $\text{MAX-SAT-ASG}_{\text{odd}}$ problem is hard for Δ_2^p , as we just proved).

We use the same notation Γ_n as in the proof above.

Let π_n be any instance of $\text{MAX-SAT-ASG}_{\text{odd}_n}$. Each π_n corresponds to an assignment $\text{holds}_1^*, \dots, \text{holds}_p^*$ of truth values to $\text{holds}_1, \dots, \text{holds}_p$ s.t. holds_i is evaluated to true (i.e., $\text{holds}_i^* = \text{true}$) if and only if γ_i belongs to π_n . Obviously, Γ_n and π_n are equivalent under the (partial) assignment $\text{holds}_1^*, \dots, \text{holds}_p^*$.

To each pair $\langle \star, \pi_n \rangle$ we associate the $p + n + 1$ -vector of constraints

$$C_n = \langle \text{holds}_1, \dots, \text{holds}_p, \Gamma_n \wedge \text{con}, x_1, \dots, x_n \rangle,$$

where con is a new variable (which means “consistent”). C_n depends only on n .

We observe that each π_n can be associated to a pair $\langle P_{\pi_n}, \alpha_{\pi_n} \rangle$ s.t.

For every $i \in 1, \dots, p$, $P_{\pi_n}[i] = 2^{n+1}(p+1)$ if γ_i belongs to π_n , and $P_{\pi_n}[i] = 1$ otherwise, $P_{\pi_n}[p+1] = 2^n(p+1)$, and for every $i \in 1, \dots, n$, $P_{\pi_n}[p+1+i] = 2^{n-i}(p+1)$.

$$\alpha_{\pi_n} = x_n \wedge \text{con}.$$

Remind here that numbers are represented in binary notation and that the size of π_n is at least n .

The weights are set in such a way every constraint holds_i is among the most prioritary ones whenever γ_i belongs to π_n , and among the less prioritary ones in the remaining case. The remaining constraints $\Gamma_n \wedge \text{con}$ and $x_1, \dots, x_n$ are put into $n+1$ strata in a decreasing order of priority (no compensation between strata can be achieved).

By construction, every preferred model ω of $\langle C_n, P_{\pi_n} \rangle$ is such that for every $i \in 1, \dots, p$, $\omega(\text{holds}_i) = \text{true}$ if and only if γ_i belongs to π_n . Now, if π_n is inconsistent, then $\Gamma_n \wedge \text{con}$ is inconsistent as well under the partial assignment $\text{holds}_1^*, \dots, \text{holds}_p^*$ of truth values associated to π_n . Therefore, under this partial assignment, C_n is independent from con when π_n is inconsistent, hence $x_n \wedge \text{con}$ cannot be a consequence of $\langle C_n, P_{\pi_n} \rangle$ (if a preferred model ω of $\langle C_n, P_{\pi_n} \rangle$ satisfies $\omega(\text{con}) = \text{true}$, then the model that coincides with ω except for variable con also is a preferred model of $\langle C_n, P_{\pi_n} \rangle$). Otherwise, π_n is consistent and the constraint $\Gamma_n \wedge \text{con}$ imposes that every preferred model ω of $\langle C_n, P_{\pi_n} \rangle$ satisfies $\omega(\text{con}) = \text{true}$. The remaining strata w.r.t. the priority order induced by the weights concern $x_1, \dots, x_n$ and they lead to select as a unique preferred model of $\langle C_n, P_{\pi_n} \rangle$ the maximal model $\omega_{\max}$ of π_n w.r.t. the lexicographic ordering induced by $x_1 < \dots < x_n$. Thus, $\langle \star, \pi_n \rangle$ is a positive instance of $\star\text{MAX-SAT-ASG}_{\text{odd}_n}$ if and only if $\langle C_n, \langle P_{\pi_n}, \alpha_{\pi_n} \rangle \rangle$ is a positive instance of INFERENCE (with fixed C and varying $\langle P, \alpha \rangle$), and this completes the proof. $\square$

It must be noted that the compilability results above still hold whenever the hard constraints are known at the off-line stage (especially, when there is no hard constraint), and, as to INFERENCE, when the queries α are restricted to literals (while INFERENCE is

obviously in nu-compP when queries are limited to literals and preferences are in the fixed part since there is only a polynomial number of literals—see Theorem 2.1 from [11]).

These results simply show that neither the on-line complexity of **MODEL CHECKING** nor the on-line complexity of **INFERENCE** can be lowered by a poly-size pre-processing; this just reflects in formal terms the basic intuition according to which no useful computation can be done off-line when preferences are not available (provided that the size of the compiled form remains polynomial in the input size).

To sum up, the compilability results we derived are mainly negative ones; in particular, they show that improving inference or model checking in the worst case through pre-processing is very unlikely when preferences are not fixed (it would lead the polynomial hierarchy to collapse). Accordingly, rendering on-line inference tractable cannot be achieved in the worst case, unless the poly-size requirement on the compiled form is relaxed (or the standard complexity assumptions do not hold). Nevertheless, since non-compilability results concern the worst case only, they do not prevent a compilation approach from giving some computational benefits in practice, at least for some weighted bases.

4. Compiling weighted bases

In this section, we first show how knowledge compilation techniques for improving clausal entailment can be used in order to compile weighted bases. Then, we present some complexity results showing that compiling a weighted base is not always a good idea, since the complexity of inference from a compiled base does not necessarily decrease. We specifically focus on prime implicates [49], and Horn covers, and renamable Horn covers compilations [7].

4.1. A framework for weighted bases compilation

Let $W = \{\langle \phi_1, k_1 \rangle, \dots, \langle \phi_n, k_n \rangle\}$ be a weighted base. In the case $\bigwedge_{i=1}^n \phi_i$ is consistent, then $K(W) = 0$ and $\min_W(\Omega)$ is the set of all models of $\bigwedge_{i=1}^n \phi_i$. Accordingly, in this situation, inference $\vdash_W$ is classical entailment, so it is possible to directly use any knowledge compilation function and compiling W comes down to compile $\bigwedge_{i=1}^n \phi_i$. However, this situation is very specific and out of the ordinary when weighted bases are considered (otherwise, weights would be useless). A difficulty is that, in the situation $\bigwedge_{i=1}^n \phi_i$ is inconsistent, we cannot compile directly this sentence using any equivalence-preserving knowledge compilation function (otherwise, trivialization would not be avoided). Indeed, in this situation, $\vdash_W$ is not classical entailment any longer, so a more sophisticated approach is needed.

In order to compile weighted bases, it is helpful to consider weighted bases in normal form.

Definition 4.1 (*Weighted bases in normal form*). A belief base $W = \{\langle \phi_1, k_1 \rangle, \dots, \langle \phi_n, k_n \rangle\}$ is in normal form if and only if for every $i \in 1, \dots, n$, either $k_i = +\infty$ or ϕ_i is a literal.

Every weighted base can be turned into a query-equivalent base in normal form.

Definition 4.2 (*V-equivalence of weighted bases*). Let W_1 and W_2 be two weighted bases and let $V \subseteq PS$. W_1 and W_2 are *V-equivalent* if and only if for every pair of sentences α and β in $PROP_V$, we have $\alpha \vdash_{W_1} \beta$ precisely when $\alpha \vdash_{W_2} \beta$.

Accordingly, two *V-equivalent* weighted bases must agree on queries built up from the symbols in V . Note that a stronger notion of equivalence can be defined by requiring that both bases induce the same weight function, i.e., $K_{W_1} = K_{W_2}$ [27]. Finally, note that if K_{W_1} and K_{W_2} agree on the sentences in $PROP_V$, then W_1 and W_2 must be *V-equivalent*.

Definition 4.3 (*Normalization of a weighted base*). Let $W = \{\langle \phi_1, k_1 \rangle, \dots, \langle \phi_n, k_n \rangle\}$ be a weighted base. The *normalization* of W is the weighted base $W \downarrow \stackrel{\text{def}}{=} H \cup S$ defined as follows:

$$H \stackrel{\text{def}}{=} \{ \langle \phi_i, +\infty \rangle \mid \langle \phi_i, +\infty \rangle \in W \},$$

$$S \stackrel{\text{def}}{=} \{ \langle \text{holds}_i \Rightarrow \phi_i, +\infty \rangle, \langle \text{holds}_i, k_i \rangle \mid \langle \phi_i, k_i \rangle \in W \text{ and } k_i \neq +\infty \},$$

where $\{\text{holds}_1, \dots, \text{holds}_n\} \subseteq PS \setminus \text{Var}(W)$.

Obviously, the normalization of a weighted base is in normal form. Intuitively, the variable holds_i is guaranteed to be false in any world that violates the sentence ϕ_i and, hence, that world is guaranteed to incur the penalty k_i . We now have the following equivalence result between a weighted base and its normalization.

Proposition 4.1. *Let W be a weighted base and let $W \downarrow$ be its normalization. Then K_W and $K_{W \downarrow}$ agree on all weights of sentences in $PROP_{\text{Var}(W)}$ and, hence, $W \downarrow$ is $\text{Var}(W)$ -equivalent to W .*

Proof. It is immediate to show that if K_W and $K_{W \downarrow}$ agree on all weights of sentences in $PROP_{\text{Var}(W)}$, then $W \downarrow$ is $\text{Var}(W)$ -equivalent to W . Hence, it remains to show that K_W and $K_{W \downarrow}$ agree on all weights of sentences in $PROP_{\text{Var}(W)}$, or, equivalently, to show that for every world $\omega \in 2^{\text{Var}(W)}$, we have $K_W(\omega) = K_{W \downarrow}(\omega)$. Let ω be any world from $2^{\text{Var}(W)}$. By definition, we have

$$K_{W \downarrow}(\omega) = \min_{\omega' \models \omega} K_{W \downarrow}(\omega'),$$

where ω' is a world from $2^{\text{Var}(W \downarrow)}$. It is sufficient then to show that:

- $K_W(\omega') \leq K_{W \downarrow}(\omega')$ for all $\omega' \models \omega$, and
- $K_W(\omega') = K_{W \downarrow}(\omega')$ for some $\omega' \models \omega$,

which is quite easy. Given the above two conditions, we get

$$K_{W \downarrow}(\omega) = \min_{\omega' \models \omega} K_{W \downarrow}(\omega') = \min_{\omega' \models \omega} K_W(\omega') = K_W(\omega). \quad \square$$

Table 1
The weight function $K_W \downarrow$ for Example 1.1

World	$K_W \downarrow$
$a, b, \text{holds}_1, \text{holds}_2$	$+\infty$
$a, b, \text{holds}_1, \neg \text{holds}_2$	1
$a, b, \neg \text{holds}_1, \text{holds}_2$	$+\infty$
$a, b, \neg \text{holds}_1, \neg \text{holds}_2$	3
$a, \neg b, \text{holds}_1, \text{holds}_2$	$+\infty$
$a, \neg b, \text{holds}_1, \neg \text{holds}_2$	$+\infty$
$a, \neg b, \neg \text{holds}_1, \text{holds}_2$	2
$a, \neg b, \neg \text{holds}_1, \neg \text{holds}_2$	3
$\neg a, b, \text{holds}_1, \text{holds}_2$	$+\infty$
$\neg a, b, \text{holds}_1, \neg \text{holds}_2$	$+\infty$
$\neg a, b, \neg \text{holds}_1, \text{holds}_2$	$+\infty$
$\neg a, b, \neg \text{holds}_1, \neg \text{holds}_2$	3
$\neg a, \neg b, \text{holds}_1, \text{holds}_2$	$+\infty$
$\neg a, \neg b, \text{holds}_1, \neg \text{holds}_2$	$+\infty$
$\neg a, \neg b, \neg \text{holds}_1, \text{holds}_2$	2
$\neg a, \neg b, \neg \text{holds}_1, \neg \text{holds}_2$	3

Example 1.1 (Continued). The weighted base W of Example 1.1 can be normalized as follows:

$$W \downarrow = \{ \langle \text{holds}_1 \Rightarrow (a \wedge b), +\infty \rangle, \\ \langle \text{holds}_2 \Rightarrow \neg b, +\infty \rangle, \langle \text{holds}_1, 2 \rangle, \langle \text{holds}_2, 1 \rangle \}.$$

The normalized weighted base $W \downarrow$ induces the weight function given in extension in Table 1. We have $K(W \downarrow) = 1$ and

$$\min_{W \downarrow}(\Omega) = \{ (a, b, \text{holds}_1, \neg \text{holds}_2) \}.$$

Moreover, K_W and $K_W \downarrow$ agree on all sentences constructed from variables in $\{a, b\}$.

Let us now focus on some specific weighted bases in normal form:

Definition 4.4 (*C-normal weighted base*). Let C be any subset of $PROP_{PS}$. A belief base W is said to be *C-normal* if and only if it is in normal form and the unique hard constraint ϕ s.t. $\langle \phi, +\infty \rangle$ satisfies $\phi \in C$.

It is obvious that every weighted base in normal form can be turned into an equivalent *C-normal* one, whenever C is any complete propositional fragment, i.e., for every sentence $\phi \in PROP_{PS}$, there exists a sentence $\phi_C \in C$ s.t. $\phi \equiv \phi_C$ holds. Indeed, the weight functions of $W \downarrow = H \cup S$ and $\{ \langle \bigwedge_{\langle \phi_i, +\infty \rangle \in H} \phi_i, +\infty \rangle \} \cup S$ coincide.⁷

⁷ Together with Proposition 4.1, this simple property also shows any weighted base W can be turned into a weighted base W_{CNF} in which every weighted sentence is a clause and K_W and $K_{W_{CNF}}$ agree on every sentence of $PROP_{Var(W)}$. Computing such a W_{CNF} can be achieved in time linear in $|W|$ as soon as every weighted

We now have all the ingredients required to present our compilation approach. Given a weighted base W , the basic idea is to compute a C -normal base that is query-equivalent to W (i.e., V -equivalent to it, where $V = \text{Var}(W)$), where C is the target class of an equivalence-preserving compilation function $COMP$ for clausal entailment. Slightly abusing word, we identify the propositional fragment C with any equivalence-preserving compilation function $COMP$ having C as a target class.

From here on, we will use $\widehat{W}$ to denote the conjunction of all hard constraints of W , i.e., sentences in the weighted base W that have $+\infty$ weights:

$$\widehat{W} \stackrel{\text{def}}{=} \bigwedge_{\langle \phi_i, +\infty \rangle \in W} \phi_i.$$

Definition 4.5 (*Compilation of a weighted base*). Let $W = \{\langle \phi_1, k_1 \rangle, \dots, \langle \phi_n, k_n \rangle\}$ be a weighted base. Let $COMP$ be any equivalence-preserving knowledge compilation function. The $COMP$ -compilation of W is the $COMP$ -normal weighted base

$$W \downarrow_{COMP} \stackrel{\text{def}}{=} \{ \langle COMP(\widehat{W} \downarrow), +\infty \rangle \} \cup \{ \langle \text{holds}_i, k_i \rangle \mid \langle \text{holds}_i, k_i \rangle \in W \downarrow \text{ and } k_i \neq +\infty \}.$$

That is, to compile a weighted base W , we perform three steps. First, we compute a normal form $W \downarrow$ according to Proposition 4.1, which is guaranteed to be $\text{Var}(W)$ -equivalent to W . Next, we combine all of the hard constraints of $W \downarrow$ into a single hard constraint $\widehat{W} \downarrow$. Finally, we compile $\widehat{W} \downarrow$ using the function $COMP$.

Example 1.1 (*Continued*). We have

$$\widehat{W} \downarrow = (\neg \text{holds}_1 \vee (a \wedge b)) \wedge (\neg \text{holds}_2 \vee \neg b).$$

Accordingly, the *Blake*-compilation⁸ of W is

$$\{ \langle (\neg \text{holds}_1 \vee a) \wedge (\neg \text{holds}_1 \vee b) \wedge (\neg \text{holds}_2 \vee \neg b) \wedge (\neg \text{holds}_1 \vee \neg \text{holds}_2), +\infty \rangle, \langle \text{holds}_1, 2 \rangle, \langle \text{holds}_2, 1 \rangle \}.$$

Note that if $\widehat{W}$ is consistent, then $\widehat{W} \downarrow$ is consistent as well (any model ω of $\widehat{W}$ over $\text{Var}(W)$ can be extended to a model $\omega \downarrow$ of $\widehat{W} \downarrow$ by setting $\omega \downarrow \models \neg \text{holds}_i$ for $i \in 1, \dots, n$). Accordingly, if $K(W) \neq +\infty$, then $K(W \downarrow) \neq +\infty$.

Given Proposition 4.1, and since $COMP$ is equivalence-preserving, we have:

Corollary 4.1. Let $W = \{\langle \phi_1, k_1 \rangle, \dots, \langle \phi_n, k_n \rangle\}$ be a weighted base. Let $COMP$ be any equivalence-preserving knowledge compilation function. $K_{W \downarrow_{COMP}}$ and K_W agree on the sentences in $\text{PROP}_{\text{Var}(W)}$. Hence, W and $W \downarrow_{COMP}$ are $\text{Var}(W)$ -equivalent.

sentence in W is in conjunctive normal form; this transformation enables one to take advantage of algorithms for WEIGHTED-MAX-SAT for computing elements from $\min_W(\Omega)$ for any W .

⁸ That is, the $COMP_{\text{Blake}}$ -compilation of W where $COMP_{\text{Blake}}$ is any algorithm for computing prime implicates.

It is important to observe here that $COMP(\widehat{W})$ is *independent* from the weights associated to the soft constraints. This gives a lot of flexibility to our approach since it renders possible to change the weights without requiring a re-compilation (as long as soft constraints do not become hard ones, of course). Thus, assuming that a *COMP*-compilation of a weighted base W has been computed and that *INFERENCE* is tractable from such a compilation ($COMP = DNNF$ works for it as we will see), Proposition 4.1 shows that clausal inference from any *COMP*-normal weighted base obtained by modifying the weights of some soft constraints (keeping them finite) is still feasible in polynomial time.

4.2. Some complexity results

We next consider a number of tractable classes of sentences, which are target classes for some existing equivalence-preserving compilation functions *COMP*:

- The *Blake* class is the set of sentences given in prime implicants normal form;
- The *Horn cover* class is the set of disjunctions of *Horn CNF* sentences;
- The *renamable Horn cover* class (*r. Horn cover* for short) is the set of the disjunctions of *renamable Horn CNF* sentences.

The *Blake* class is the target class of the compilation function $COMP_{Blake}$ described in [49]. The *Horn cover* class and the *renamable Horn cover* class are target classes for the tractable covers compilation functions given in [7]. We shall note respectively $COMP_{Horn\ cover}$ and $COMP_{r.\ Horn\ cover}$ the corresponding compilation functions.

Accordingly, a *Blake* (respectively *Horn cover*, *r. Horn cover*)-normal weighted base W is defined as a weighted base in normal form whose unique hard constraint belongs to the *Blake* (respectively *Horn cover*, *r. Horn cover*) class.

In the next section, we will also focus on the *DNNF* class. We consider it separately because—unlike the other classes—it makes clausal inference from the corresponding normal bases tractable.

Of course, all these compilation functions *COMP* are subject to the limitation mentioned 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 clausal entailment problem (see e.g., the experimental results given in [7,22,50]).

As indicated previously, knowledge compilation can prove helpful only if inference from the compiled form is computationally easier than direct inference. Accordingly, it is important to identify the complexity of inference from a compiled weighted base if we want to draw some conclusions about the usefulness of knowledge compilation in this context.

When no restriction is put on W , *INFERENCE* is known as Δ_2^P -complete [26], even in the restricted case queries α are literals. Now, what if W is a *C*-normal weighted base? We have identified the following results:

Proposition 4.2. *The complexity of INFERENCE and of its restrictions to literal inference, is Δ_2^P -complete when W is a Blake, Horn cover, or r. Horn cover-normal weighted base.*

Proof.

- Membership: all results come directly from the membership to Δ_2^P of the more general problem INFERENCE [26].
- Hardness: all results come from the Δ_2^P -hardness of literal (skeptical) inference $\vdash_{lex}$ from a compiled stratified belief base [15] interpreted under the lexicographic policy. Indeed, if m is the maximum number of sentences belonging to any stratum B_i of $B = (B_1, \dots, B_k)$, then let $W_B = \{\langle \phi, (m+1)^{k-i} \rangle \mid \phi \in B_i\}$ (see Example 1.2 for an illustration). Clearly enough, W_B can be computed in time polynomial in $|B|$. We have $B \vdash_{lex} \alpha$ if and only if $true \vdash_{W_B} \alpha$. $\square$

According to the complexity results in Proposition 4.2, there is no guarantee that compiling a belief base using the *Blake* (or the *Horn cover* or the *r. Horn cover*) compilation function leads to improve inference since its complexity from the corresponding compiled bases is just as hard as the complexity of INFERENCE in the general case.

Fortunately, it is not the case that such negative results hold for every compilation function. As we will see in the next section, *DNNF*-normal weighted bases exhibit a much better behaviour.

5. Compiling weighted bases using *DNNF*

In this section, we focus on *DNNF*-compilations of weighted bases. After a brief review of what *DNNF* sentences are, we show that *DNNF*-compilations support several computational tasks in polynomial time: preferred model enumeration (hence model checking) and (clausal) inference.

5.1. A glimpse at the *DNNF LANGUAGE*

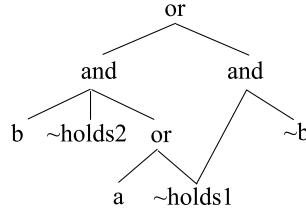
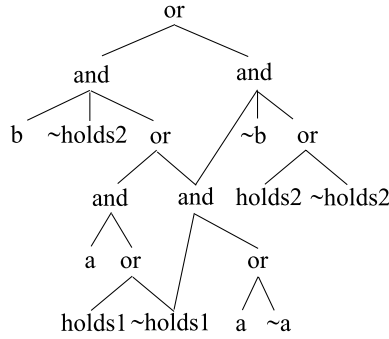
The *DNNF* language is a subset of the (DAG)-*NNF* one:

Definition 5.1 (*NNF*). Let PS be a finite set of propositional variables. A sentence in (DAG)-*NNF* is a rooted, directed acyclic graph (DAG) where each leaf node is labeled with *true*, *false*, x or $\neg x$, $x \in PS$; each internal node is labeled with $\wedge$ or $\vee$ and can have arbitrarily many children.

The *DNNF* language contains exactly the *NNF* sentences satisfying the *decomposability* property [19,20]:

Definition 5.2 (*DNNF*). A sentence in *DNNF* (for “Decomposable *NNF*”) is a *NNF* sentence satisfying the *decomposability* property: for each conjunction C in the sentence, the conjuncts of C do not share variables.

Fig. 1 depicts a *DNNF* of the hard constraint $\widehat{W\downarrow}$, where W is the weighted base given in Example 1.1. Note here that $W\downarrow$ is the normal form constructed from W according to Proposition 4.1, and $\widehat{W\downarrow}$ is the conjunction of all hard constraints in $W\downarrow$.

Fig. 1. A sentence in *DNNF*.Fig. 2. A sentence in smooth *DNNF*.

Algorithms for translating *CNF* sentences into equivalent *DNNF* sentences can be found in [20,22].

An interesting subset of *DNNF* is the set of *smooth DNNF* sentences [20,21]:

Definition 5.3 (*Smooth DNNF*). A *DNNF* sentence satisfies the *smoothness* property if and only if for each disjunction C in the sentence, each disjunct of C mentions the same variables.

Interestingly, every *DNNF* sentence can be turned into an equivalent, smooth one in polynomial time [21].

For instance, Fig. 2 depicts a smooth *DNNF* which is equivalent to the *DNNF* in Fig. 1. Note that for readability reasons some leaf nodes are duplicated in the figure.

Among the various tasks that can be achieved in a tractable way from a smooth *DNNF* sentence are *conditioning*, *clausal entailment*, *forgetting* and *model enumeration* [20,24, 25].

5.2. Tractable queries

Given a weighted base W , and given a *DNNF*-compilation of W , we now show how the compilation can be used to represent the preferred models of W as a *DNNF* in polynomial time.

Definition 5.4 (*Minimization of a weighted base*). A *minimization* of a weighted base W is a propositional sentence Δ where the models of Δ are $\min_W(\mathcal{Q})$.

Note that this notion generalizes the notion of minimization of a propositional sentence ϕ reported in [19], for which the preferred models are those containing a maximal number of variables assigned to true. Such a minimization can be easily achieved in a weighted base setting by considering the base

$$W = \{\langle \phi, +\infty \rangle\} \cup \bigcup_{x \in \text{Var}(\phi)} \{\langle x, 1 \rangle\}.$$

Given a partition $\{P, Q, Z\}$ of PS , our notion of minimization is also sufficient to capture the cardinality-based circumscription $NCIRC((P, Q, Z))(\phi)$ of a propositional sentence ϕ in the restricted case there is no fixed variables ($Q = \emptyset$) [38] [40]. Indeed, we have $NCIRC((P, \emptyset, Z))(\phi) \equiv \min_W(\Omega)$, with

$$W = \{\langle \phi + \infty \rangle\} \cup \bigcup_{p \in P} \{\langle p, 1 \rangle\}.$$

Let us now explain how to compute recursively the weight and a minimization of a given *DNNF*-normal weighted base; we first need the two following definitions:

Definition 5.5 (*Weight of DNNF-normal weighted base*). Let W be a *DNNF*-normal weighted base. Let $\langle \alpha, +\infty \rangle$ be the single hard constraint in W , where α is a smooth *DNNF* sentence. We define $k(\alpha)$ inductively as follows:

- $k(\text{true}) \stackrel{\text{def}}{=} 0$ and $k(\text{false}) \stackrel{\text{def}}{=} +\infty$.
- If α is a literal:
 - If $\langle \alpha, k \rangle \in W$ with $k \neq +\infty$, then $k(\alpha) \stackrel{\text{def}}{=} k$.
 - Otherwise, $k(\alpha) \stackrel{\text{def}}{=} 0$.
- $k(\alpha = \bigvee_i \alpha_i) \stackrel{\text{def}}{=} \min_i k(\alpha_i)$.
- $k(\alpha = \bigwedge_i \alpha_i) \stackrel{\text{def}}{=} \sum_i k(\alpha_i)$.

Definition 5.6 (*Minimization of DNNF-normal weighted base*). Let W be a *DNNF*-normal weighted base. Let $\langle \alpha, +\infty \rangle$ be the single hard constraint in W , where α is a smooth *DNNF* sentence. We define $\min(\alpha)$ inductively as follows:

- If α is a literal or a boolean constant, then $\min(\alpha) \stackrel{\text{def}}{=} \alpha$.
- $\min(\alpha = \bigvee_i \alpha_i) \stackrel{\text{def}}{=} \bigvee_{k(\alpha_i)=k(\alpha)} \min(\alpha_i)$.
- $\min(\alpha = \bigwedge_i \alpha_i) \stackrel{\text{def}}{=} \bigwedge_i \min(\alpha_i)$.

We have the following result:

Proposition 5.1. *Let W be a DNNF-normal weighted base. Let $\langle \alpha, +\infty \rangle$ be the single hard constraint in W , where α is a consistent, smooth DNNF sentence. Then $\min(\alpha)$ is a smooth DNNF and is a minimization of W .*

Proof. The fact that $\min(\alpha)$ is a smooth DNNF is easy to be proved by induction. The fact that it is a minimization of W is a direct generalization of Theorems 10 and 11 from [20].

The proof is as follows. Let W^* be the weighted base $W \setminus \{\langle \alpha, +\infty \rangle\}$. We first prove two lemmata about the subsentences α' of α . For every subsentence α' of α , we have:

Lemma 1. $k(\alpha') = K_{W^*}(\alpha')$.

Lemma 2. $\text{Mod}(\min(\alpha'))$ is the set of models of the projections on $\text{Var}(\alpha')$ of the models of $\min_{W^*}(\text{Mod}(\alpha'))$.

Those lemmata can be proved by structural induction on α' .

Proof of Lemma 1. By structural induction:

- Base case.
 - $\alpha' = \text{true}$. Let ω be any interpretation s.t. for every soft constraint ϕ_i (where ϕ_i is a literal), we have $\omega \models \phi_i$. By construction, ω is a model of every sentence from W^* . Hence, $K_{W^*}(\omega) = 0$. Since 0 is the least possible penalty and ω is a model of true , we have $K_{W^*}(\text{true}) = 0 = k(\text{true})$.
 - $\alpha' = \text{false}$. $K_{W^*}(\text{false}) = \min_{\omega \models \text{false}} K_{W^*}(\omega)$. There are no models of false . Since $\min$ is associative and the neutral element for $\min$ is $+\infty$, we have $\min_{\omega \models \text{false}} K_{W^*}(\omega) = +\infty$. Hence, $K_{W^*}(\text{false}) = +\infty$.
 - $\alpha' = \neg\phi_i$ where ϕ_i is a soft constraint. Let ω be any interpretation s.t. $\omega \models \neg\phi_i$ and for every other soft constraint ϕ_j (with $j \neq i$), $\omega \models \phi_j$. By construction, $K_{W^*}(\omega) = k_i$. Since $\langle \phi_i, k_i \rangle \in W^*$, every model ω' of $\neg\phi_i$ is s.t. $K_{W^*}(\omega') \geq k_i$. Hence, $K_{W^*}(\neg\phi_i) \geq k_i$. Since the minimal value k_i is reached by the model ω of $\neg\phi_i$, we have $K_{W^*}(\neg\phi_i) = k_i = k(\neg\phi_i)$.
 - α' is any literal l , not equivalent to the negation any soft constraint ϕ_i . Let ω be any interpretation s.t. $\omega \models l$ and for every soft constraint ϕ_i , $\omega \models \phi_i$. By construction, ω is a model of every sentence from W^* . Hence, $K_{W^*}(\omega) = 0$. Since 0 is the least possible penalty and ω is a model of l , we have $K_{W^*}(l) = 0 = k(l)$.
- Inductive step.
 - α' is an and-node $\alpha'_1 \wedge \dots \wedge \alpha'_p$. Since the α'_i do not share any variable, the set of models of α' over $\text{Var}(\alpha')$ is the cross-product of the sets of models of the α'_i over their respective sets of variables $\text{Var}(\alpha'_i)$. As an immediate consequence, the set of preferred (w.r.t. W^*) models of α' over $\text{Var}(\alpha')$ is the cross-product of the sets of preferred models of the α'_i over their respective sets of variables $\text{Var}(\alpha'_i)$. As a consequence,

$$\begin{aligned}
 K_{W^*}(\alpha'_1 \wedge \dots \wedge \alpha'_p) &= \sum_{i=1, \dots, p} K_{W^*}(\alpha'_i) \\
 &= \sum_{i=1, \dots, p} k(\alpha'_i) \quad (\text{by induction hypothesis}) \\
 &= k(\alpha') \quad \text{by construction.}
 \end{aligned}$$

- α' is an or-node $\alpha'_1 \vee \dots \vee \alpha'_p$. By definition,

$$K_{W^*}(\alpha'_1 \vee \dots \vee \alpha'_p) = \min_{\omega \models \alpha'_1 \vee \dots \vee \alpha'_p} (K_{W^*}(\omega)).$$

Since $\text{Mod}(\alpha'_1 \vee \dots \vee \alpha'_p) = \bigcup_{i=1}^p \text{Mod}(\alpha'_i)$, this is also equal to

$$\begin{aligned} & \min_{i \in 1, \dots, p} (\min_{\omega \models \alpha'_i} K_{W^*}(\omega)) \\ &= \min_{i \in 1, \dots, p} (K_{W^*}(\alpha'_i)) \\ &= \min_{i \in 1, \dots, p} (k(\alpha'_i)) \quad (\text{by inductive hypothesis}) \\ &= k(\alpha') \quad \text{by construction.} \quad \square \end{aligned}$$

Proof of Lemma 2. By structural induction:

- Base case.
 - $\alpha' = \text{true}$. Since α' is consistent, it has some models, hence some preferred models for W^* . Forgetting all the variables in a consistent sentence gives a sentence equivalent to $\text{true} = \min(\alpha')$ by construction.
 - $\alpha' = \text{false}$. Since α' has no model, it has no preferred model. Forgetting all the variables in an inconsistent sentence gives a sentence equivalent to $\text{false} = \min(\alpha')$ by construction.
 - α' is any literal l . Since α' is consistent, it has some models, hence some preferred models for W^* . Each of them satisfies l , hence projecting them on $\text{Var}(l)$ gives $l = \min(\alpha')$ by construction.
- Inductive step.
 - α' is an and-node $\alpha'_1 \wedge \dots \wedge \alpha'_p$. As explained in the proof of Lemma 1 above, the set of preferred (w.r.t. W^*) models of α' over $\text{Var}(\alpha')$ is the cross-product of the sets of preferred models of the α'_i over their respective sets of variables $\text{Var}(\alpha'_i)$. Stated otherwise, the set of the projections of the preferred models of α' on $\text{Var}(\alpha')$ is the cross-product of the sets of the projections of the preferred models of the α'_i on the respective $\text{Var}(\alpha'_i)$. By induction hypothesis, it comes that the set of the projections of the preferred models of α' on $\text{Var}(\alpha')$ is the cross-product of the sets of models of $\min(\alpha'_i)$. Since the α'_i do not share any variables, this cross-product set is equal to $\text{Mod}(\bigwedge_{i=1}^p \min(\alpha'_i))$, which is also equal to $\min(\alpha')$ by construction.
 - α' is an or-node $\alpha'_1 \vee \dots \vee \alpha'_p$. Since $\text{Mod}(\alpha'_1 \vee \dots \vee \alpha'_p) = \bigcup_{i=1}^p \text{Mod}(\alpha'_i)$, we have

$$\min_{W^*}(\text{Mod}(\alpha')) = \min_{W^*} \left(\bigcup_{i=1}^p \text{Mod}(\alpha'_i) \right) = \min_{W^*} \left(\bigcup_{i=1}^p \min_{W^*}(\text{Mod}(\alpha'_i)) \right).$$

Among the models of $\bigcup_{i=1}^p \min_{W^*}(\text{Mod}(\alpha'_i))$, the preferred ones (w.r.t. W^*) are by definition those ω of minimal weight, i.e., those for which $K_{W^*}(\omega) = K_{W^*}(\alpha')$. Whenever such an ω belongs to $\min_{W^*}(\text{Mod}(\alpha'_i))$, then $K_{W^*}(\alpha'_i) = K_{W^*}(\alpha')$. Thus, we have

$$\min_{W^*}(\text{Mod}(\alpha')) = \bigcup_{i \in 1, \dots, p \mid K_{W^*}(\alpha'_i) = K_{W^*}(\alpha')} \min_{W^*}(\text{Mod}(\alpha'_i)). \quad (1)$$

Let $\omega[S]$ be the projection of an interpretation ω on the set S of variables, also viewed as a term. By construction,

$$\min(\alpha') = \bigvee_{i \in 1, \dots, p \mid k(\alpha'_i) = k(\alpha')} \min(\alpha'_i);$$

equivalently, we have

$$\text{Mod}(\min(\alpha')) = \bigcup_{i \in 1, \dots, p \mid k(\alpha'_i) = k(\alpha')} \text{Mod}(\min(\alpha'_i)).$$

Lemma 1 shows that $k(\alpha'_i) = K_{W^*}(\alpha'_i)$ and $k(\alpha') = K_{W^*}(\alpha')$. Hence,

$$\text{Mod}(\min(\alpha')) = \bigcup_{i \in 1, \dots, p \mid K_{W^*}(\alpha'_i) = K_{W^*}(\alpha')} \text{Mod}(\min(\alpha'_i)).$$

Now, by induction hypothesis, we have

$$\text{Mod}(\min(\alpha'_i)) = \bigcup_{\omega \in \min_{W^*}(\text{Mod}(\alpha'_i))} \text{Mod}(\omega[\text{Var}(\alpha'_i)]).$$

Since α is a smooth *DNNF*, we have $\text{Var}(\alpha') = \text{Var}(\alpha'_1) = \dots = \text{Var}(\alpha'_p)$ for every subsentence α' of α . Hence, we have

$$\text{Mod}(\min(\alpha'_i)) = \bigcup_{\omega \in \min_{W^*}(\text{Mod}(\alpha'_i))} \text{Mod}(\omega[\text{Var}(\alpha')]).$$

We obtain

$$\text{Mod}(\min(\alpha')) = \bigcup_{i \in 1, \dots, p \mid K_{W^*}(\alpha'_i) = K_{W^*}(\alpha')} \bigcup_{\omega \in \min_{W^*}(\text{Mod}(\alpha'_i))} \text{Mod}(\omega[\text{Var}(\alpha')]).$$

Taking advantage of Eq. (1), we obtain

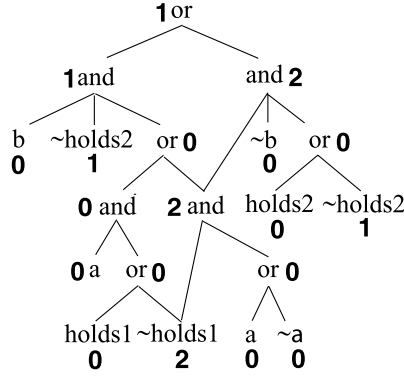
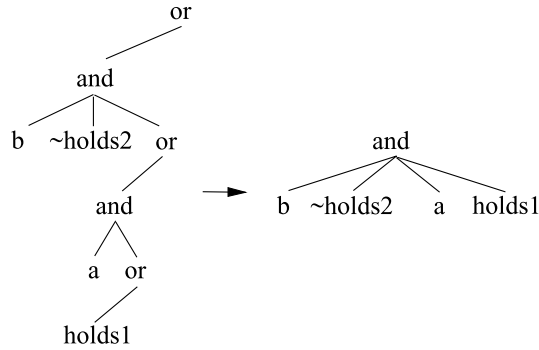
$$\text{Mod}(\min(\alpha')) = \bigcup_{\omega \in \min_{W^*}(\text{Mod}(\alpha'))} \text{Mod}(\omega[\text{Var}(\alpha')]),$$

and this concludes the proof. $\square$

From Lemma 2, we can infer that $\text{Mod}(\min(\alpha))$ is the set of the projections on $\text{Var}(\alpha)$ of the models of $\min_{W^*}(\text{Mod}(\alpha))$. Since $\text{Var}(\alpha) = \text{Var}(W)$, the projection step does not matter here: $\text{Mod}(\min(\alpha)) = \min_{W^*}(\text{Mod}(\alpha))$. Since α is consistent and is the unique hard constraint of W , we also have $\min_{W^*}(\text{Mod}(\alpha)) = \min_W(\Omega)$, which concludes the proof. $\square$

Since every *DNNF*-compilation of a weighted base W is *DNNF*-normal, the previous proposition can be used to derive its minimization represented as a smooth *DNNF* sentence.

Fig. 3 depicts the weight $k(\alpha)$ of every subsentence α of the smooth *DNNF* sentence given in Fig. 2. Fig. 4 (left) depicts the minimization of the *DNNF* in Fig. 3. Fig. 4 (right) depicts a simplification of this minimized *DNNF* which has a single model.

Fig. 3. Weights on a smooth *DNNF* sentence.Fig. 4. Minimization of a smooth *DNNF* sentence.

The consistency requirement on α is not very restricting. As mentioned previously, α is always consistent, except when the original base W is a *DNNF*-compilation of which contains an inconsistent hard constraint. In such a pathological situation, the weight of the base is $+\infty$. Moreover, every world is a preferred model of the weighted base and valid sentences are its only consequences.

Since $\min(\alpha)$ can be computed in time polynomial in the size of *DNNF* α , and since clausal entailment can be done in time linear in the size of α [19,20], we have:⁹

Corollary 5.1. *The clausal inference problem INFERENCE for DNNF-normal weighted bases is in P.*

Since model enumeration can be done in output polynomial time from a smooth *DNNF*, we also have:

⁹ The consistency of α can be tested in time linear in the size of α , since α is a *DNNF* sentence. In the case α is inconsistent, only valid clauses are its clausal consequences (and they can be tested in polynomial time, of course), and every world is a preferred model of W , hence, they can be enumerated in output polynomial time.

Corollary 5.2. *The preferred model enumeration problem for DNNF-normal weighted bases can be solved in output polynomial time.*

Especially, this last corollary trivially shows that the model checking problem for DNNF-normal weighted bases is in P.

6. Application to model-based diagnosis

We now briefly sketch how the previous results can be used to compute the set of most likely diagnoses of a system in time polynomial in the size of system description and the output size. The following results generalize those given in [18,21] to the case where the probability of failure of components is available.

We first need to briefly recall what a consistency-based diagnosis of a system is [48]:

Definition 6.1 (*Consistency-based diagnosis*).

- A *diagnostic system* $\mathcal{P} = \langle SD, OK \rangle$ is a pair consisting of:
 - a sentence SD from $PROP_{PS}$, the system description;
 - a finite set $OK = \{ok_1, \dots, ok_n\} \subseteq PS$ of propositional symbols. “ ok_i is true” means that component i of the system to be diagnosed is not faulty.
- A *diagnostic problem* is a pair $\langle \mathcal{P}, OBS \rangle$, where $\mathcal{P}$ is diagnostic system and OBS is a diagnostic observation, that is, a term with no variables in OK . It is assumed that $SD \wedge OBS$ is consistent, which means that the observations are considered reliable (otherwise, it could be the case that the system could not be diagnosed).
- A *consistency-based diagnosis* Δ for a diagnostic problem $\langle \mathcal{P}, OBS \rangle$ is a complete OK -term (i.e., a conjunction of literals built up from OK in which every ok_i occurs either positively or negatively) s.t. $\Delta \wedge SD \wedge OBS$ is consistent.

Because a system can have a number of diagnoses that is exponential in the number of its components, preference criteria are usually used to limit the number of candidates. The most current ones consist in keeping the diagnoses containing as few negative OK -literals as possible (w.r.t. set inclusion or cardinality).

When the *a priori* probability of failure of components is available (and such probabilities are considered independent), the most likely diagnoses for $\mathcal{P}$ can also be preferred. Such a notion of preferred diagnosis generalizes the one based on minimality w.r.t. cardinality (the latter corresponds to the case the probability of failure of components is uniform and $< \frac{1}{2}$).

Interestingly, the most likely diagnoses for $\mathcal{P}$ can be enumerated in output polynomial time as soon as a smooth DNNF-compilation $\mathcal{P}_{DNNF}$ corresponding to $\mathcal{P}$ has been derived first.

Definition 6.2 (*Compilation of a diagnostic problem*). Let $\mathcal{P}$ be a diagnostic system for which the *a priori* probability of failure p_i of any component i is available.

$$\mathcal{P}_{DNNF} \stackrel{\text{def}}{=} \{ \langle DNNF(SD), +\infty \rangle \} \cup \{ \langle ok_i, \log p_i \rangle \mid ok_i \in OK \}$$

is the smooth *DNNF*-normal weighted base associated with $\mathcal{P}$.

The log transformation performed here enables us to compute the *log* of the probability $P(\Delta) = \prod_{-ok_i \in \Delta} p_i$ of a diagnosis Δ as $\sum_{-ok_i \in \Delta} \log p_i$. Because log is strictly non-decreasing, the induced preference ordering between diagnoses is preserved.

Proposition 6.1.

- $K(\mathcal{P}_{DNNF} \mid OBS)$ is the log of the probability of any most likely diagnosis for $\langle \mathcal{P}, OBS \rangle$.
- The most likely diagnoses for $\langle \mathcal{P}, OBS \rangle$ are the models of $Forget(min(DNNF(SD) \mid OBS), PS \setminus OK)$.

In this proposition, $\alpha \mid OBS$ denotes the conditioning of α on the term *OBS*, i.e., the sentence obtained by replacing in Δ every variable x by *true* (respectively *false*) if x (respectively $\neg x$) is a positive (respectively negative) literal of *OBS*. Moreover, for every sentence ϕ and every set of variables X , $Forget(\phi, X)$ denotes the logically strongest consequence of ϕ that is independent from X , i.e., that can be turned into an equivalent sentence in which no variable from X occurs. Note that forgetting can be applied to sentence in *DNNF* in time polynomial in the sentence size [20].

Proof. Since $OK \cap Var(OBS) = \emptyset$ and $SD \wedge OBS \equiv (SD \mid OBS) \wedge OBS$, the consistency-based diagnoses Δ for $\langle \mathcal{P}, OBS \rangle$ are the projections on *OK* of the models of $SD \mid OBS$. The most likely ones are those in Δ for which $\log P(\Delta) = \sum_{-ok_i \in \Delta} \log p_i$ is minimal.

On the other hand, by construction, the preferred models of the conditioned weighted base $\mathcal{P}_{DNNF} \mid OBS$ are the models ω of $DNNF(SD) \mid OBS$ s.t. $\sum_{-ok_i \in \omega} \log p_i$ is minimal. Now, since *DNNF* is equivalence-preserving, the preferred models of $\mathcal{P}_{DNNF} \mid OBS$ are the models ω of $SD \mid OBS$ s.t. $\sum_{-ok_i \in \omega} \log p_i$ is minimal. Accordingly, $K(\mathcal{P}_{DNNF} \mid OBS)$ is equal to $\sum_{-ok_i \in \omega} \log p_i$ where ω is any preferred model of $SD \mid OBS$. Subsequently, the most likely consistency-based diagnoses Δ for $\langle \mathcal{P}, OBS \rangle$ are the projections on *OK* of the preferred models of $\mathcal{P}_{DNNF} \mid OBS$ and $\log P(\Delta) = K(\mathcal{P}_{DNNF} \mid OBS)$. Finally, Proposition 5.1 shows that the models of $min(DNNF(SD) \mid OBS)$ are the preferred models of $\mathcal{P}_{DNNF} \mid OBS$, so the models of $Forget(min(DNNF(SD) \mid OBS), PS \setminus OK)$ are the projections on *OK* of the models of $min(DNNF(SD) \mid OBS)$, and this completes the proof. $\square$

Since forgetting variables in a *DNNF* sentence can be done in polynomial time [19,20], and the models of a smooth *DNNF* sentence can be generated in time polynomial in the output size [20], we obtain:

Corollary 6.1. *The most likely diagnoses for a diagnostic problem $\langle \mathcal{P}, \text{OBS} \rangle$ can be enumerated in time polynomial in the size of $\mathcal{P}_{\text{DNNF}}$.*

As far as we know, our compilation approach is the first one enabling to derive the most likely diagnoses of a system in output polynomial time once the system description has been pre-processed.

7. Other related work

Our work can be related to other previous work, which can be classified into three categories depending on the main objective: identifying compilability or complexity results, designing compilation techniques for propositional bases, applying such techniques to real-world problems, like diagnosis and configuration.

7.1. Compilability and complexity results

Our compilability results are based on the framework of [9–12]. But although the compilability of circumscription and belief revision have been investigated before [9,13], we know of no previous treatment for the compilability of model checking and inference from weighted bases.

Our results complete in some sense some of the complexity results pointed out in [14, 31,41], where the complexity of inference from stratified belief bases, interpreted under various policies, is identified in the general case and under some restrictions. While [14,41] focused on the *Horn CNF* case, we have considered other restrictions, like the one where the hard constraints are encoded as a *DNNF* sentence while the other pieces of belief are literals (this is what we called a *DNNF*-normal base). Interestingly, we have considered fragments that are complete for propositional logic: any weighted base can be compiled in our framework, even if it is not composed of *Horn CNF* sentences.

7.2. Compilation techniques

Our work is more closely related to approaches focusing on the compilation of stratified belief bases, mainly [5,14,15]. As shown in the paper, every stratified belief base skeptically interpreted under the lexicographic policy (or any restriction of it, especially the cardinality maximisation policy) can be turned in polynomial time into a weighted base. Once this is done, our compilation approach can be used. There is no obvious converse poly-time translation, so we do not know how to use compilation approaches for stratified belief bases whenever some compensations between weights are useful (like in the application to consistency-based diagnosis).

A very basic approach to implement inference from a weighted base W consists in computing the set S of all preferred subbases of W , i.e., those containing the constraints of W satisfied by a preferred model of W . Indeed, a query α is a consequence of W if and only if it is entailed by every element of S ; thus, once S has been computed, inference is reduced to classical entailment, which is “only” coNP-complete in the general case.

Clearly enough, this approach amounts to knowledge compilation: the generation of the set of preferred subbases is the compilation step.

However, the basic approach is generally not interesting for several reasons. First of all, the compiled form S of a weighted base W may easily be exponentially larger than W , while one of our compilability results show that it is possible to derive a polyspace propositional sentence that is query-equivalent to W , when preferences are fixed. Furthermore, S cannot be computed incrementally from W in the general case since some removed pieces of belief can reappear later on; starting from S only, it is not always possible to compute the preferred subbases of W extended with a new sentence (this comes easily from similar results for stratified bases [3]).

In our approach, a compiled base is query-equivalent to the original one: no information is lost, and the compilation step can be done in an incremental way (provided that the compilation function that is used admits sentences from its target class as part of the input). Incrementality is not always a decisive computational advantage but this may be the case in some situations, especially when “small changes” are performed (in this case, updating the compiled form is often less expensive than re-compiling the base from scratch). Besides, our approach is much more flexible than the basic one. Thus, 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). Especially, when $DNNF$ is used as a target class, inference from the compiled form is tractable (while it is not the case from S in the general case); furthermore, in this situation, it is known that the only exponential factor in the (time and space) complexity of the algorithm *dnnf2* (reported in [20]) for generating $DNNF(\hat{W})$ is the width of the decomposition tree generated from a CNF of $\hat{W}$ (see Theorem 16 from [20]). Even if computing such a tree with optimal width is computationally hard, there are poly-time algorithms enabling to compute “good” decomposition trees from the practical side. This permits one to predict an upper bound of the size of $DNNF(\hat{W})$. Our approach also offers the opportunity to modify the weight of any soft constraint “for free” (i.e., without requiring any expensive re-compilation step), as long as it is kept finite, while this is not the case when the basic approach is considered. The possibility to change the penalties given to some pieces of belief is important for at least two reasons. First, when designing a weighted base, some weight adjustments can be necessary, guided by the discrepancy between the set of expected conclusions and the set of achieved ones. Secondly, in a multi-agent setting where all agents are subject to the same hard constraints but may have different preferences (encoded as soft constraints), it is not necessary to handle (and compile) one base per agent but one for the whole group. This situation occurs, for instance, when timetables must be designed (the hard constraints are shared by the agents, but they usually have different preferences).

A more sophisticated compilation-based approach to inference from stratified belief bases is reported in [5]. It aims at computing a propositional sentence equivalent to a stratified belief base, skeptically interpreted under the lexicographic policy. While our compilability results show how to derive a polyspace sentence query-equivalent to the original base, the size of the compiled base computed following the approach presented in [5] may be exponential in the size of the original base. Nevertheless, compared with the basic approach, the approach presented in [5] has the major advantage that no information is lost during the compilation step, which can be achieved in an incremental way. Unlike

our approach, it does not offer the possibility to “change the weights”, i.e., to re-partition the belief base into new strata without requiring a re-compilation of the base; and unlike our approach when *DNNF* is used as a target class, it does not ensure that inference from the resulting base is tractable.

In [15], *C*-compilations of stratified belief bases have been introduced and the complexity of skeptical inference from such bases investigated for several tractable fragments *C*. In this paper, we have exploited some hardness results given in [15] to obtain similar hardness results, but in a different setting (penalty logic). We have also considered other tractable fragments, especially the *DNNF* one, that have not been taken into account in [15]. There is also a tractability result in [15] for skeptical inference from stratified belief bases interpreted under the inclusion-based policy when *C* is the *DNF* fragment, which is strictly less succinct than *DNNF* [20,24]. We did not obtain a similar result in the penalty logic framework (since the inclusion-based policy is not directly relevant to penalty logic) but we conjecture that such a tractability result cannot be extended to the case $C = \text{DNNF}$.

In [14], an approach to compile stratified belief bases skeptically interpreted under the lexicographic policy into a *OBDD* sentence is proposed. From such sentences, inference is shown tractable. This approach (which inspired our work) will first convert the given stratified belief base into an equivalent weighted base in normal form. Especially, the compiled forms that are generated are exactly what we call *OBDD*-compilations in our framework. Our approach extends [14] in two directions. On the one hand, other tractable fragments *C* can be considered as target classes, especially the *DNNF* one, and *DNNF*-normal bases are as tractable as *OBDD*-normal bases as to model checking and inference. Furthermore, since *DNNF* is strictly more succinct than *OBDD* as a propositional fragment (see [20,24]), smaller compiled forms can be expected and this is very important from the practical side. On the other hand, while Proposition 5.1 still holds when *OBDD*-compilations are considered (cf. Theorem 3 from [14]), it cannot be extended to the larger class of *OBDD*-normal weighted bases. The reason is that *OBDD* sentences are non-smooth in the general case. Indeed, consider the weighted base

$$W = \{\langle \text{OBDD}(\neg a \vee \neg b), +\infty \rangle, \langle a, 1 \rangle, \langle b, 2 \rangle\},$$

where $\text{OBDD}(\neg a \vee \neg b)$ is a (reduced) *OBDD* sentence equivalent to $\neg a \vee \neg b$ and obtained by considering variable *a* before variable *b*. *W* is an *OBDD*-normal weighted base. Its minimization, according to the extension of Theorem 3 from [14] to *OBDD*-normal bases, would be equivalent to $\neg a$, while $\neg a \wedge b$ is expected. The restriction imposed by Theorem 3 from [14] to bases subject to normalization is significant from the practical side since (1) normalizing the base through the introduction of new variables *holds_i* may easily lead to an exponential increase in the size of the *OBDD* compiled form, and (2) for many problems, the input base is already in normal form so the introduction of additional variables is useless (that is the case for the bases associated to consistency-based diagnosis problems, as shown previously in the paper). Contrastingly, our Proposition 5.1 can be directly applied to *DNNF*-normal weighted bases.

7.3. Applications

Now, from the application point of view, it is shown in [18,21] how the minimum-cardinality consistency-based diagnoses of a system can be enumerated in output polynomial time once the system description has been compiled into a *DNNF* sentence. Our approach extends the proposed technique by accounting for the probability component failure. This refinement is obtained “for free” from a computational point of view and it is important from the practical side since the set of most likely diagnoses can be exponentially smaller than the set of minimum-cardinality diagnoses. In particular, in a diagnosis approach where the search for diagnoses is interleaved with some additional measurements for discriminating among them, focusing on the most likely diagnoses may easily lead to significant time savings.

Finally, it appears that the cluster tree compilation technique described in [43,44] can be used to improve inference from propositional weighted bases. Such an approach has been evaluated on a specific (but challenging) application—the vehicle sales configuration for the automotive industry at Renault (one of the major companies in France)—and the corresponding configuration engine has exhibited very interesting performances. In particular, the ability to take weights (penalties) into account appeared as a major feature and it is not shared by many configuration engines. The cluster tree “compilation” technique is based on a divide-and-conquer principle: it exploits the fact that instantiating some variables is sufficient to make some propositional constraints logically independent from other constraints. Such a divide-and-conquer principle is the key idea of many propagation algorithms for probabilistic inference (or more generally in valuation algebras, see, e.g., [34]) and classical inference (see, e.g., [1,28]). It is also at the very core of the *DNNF* fragment (the other main idea at work is common subsentences sharing). In [43], the compiled form that is generated is a tree-structured set of sets of interpretations: the conjunction Σ_V of all input constraints that are built up from a given cluster V of variables is compiled into a *MODS* sentence (i.e., the set of models of Σ_V over V [25]). Such a set of *MODS* sentences is conjunctively interpreted. Interestingly, each Σ_V could be compiled into a *DNNF* sentence (instead of a *MODS* one) without questioning in depth the propagation algorithm. That way, more compact “compiled forms” could be derived (since *DNNF* is strictly more succinct than *MODS* [25]).

8. Conclusion

In this paper, we have studied how knowledge compilation can be used to improve model checking and inference from propositional weighted bases. We have first presented compilability results showing that computational benefits are hard to be expected in the worst case, as soon as preferences are subject to change. Then, we have presented a general notion of *C*-normal weighted base that is parametrized by any tractable class *C* for the clausal entailment problem. We have shown how every weighted base can be compiled into a query-equivalent *C*-normal base whenever *C* is a complete class for propositional logic. Both negative and positive results have been put forward. On the one hand, we have shown that the inference problem from a *C*-normal weighted base is as difficult as in the

general case, when prime implicates, Horn cover or renamable Horn cover target classes are considered. On the other hand, we have shown that this problem becomes tractable whenever *DNNF*-normal bases are used. Finally, we have sketched how our results can be used in model-based diagnosis in order to compute the most likely diagnoses of a system.

This work calls for several perspectives, both from the theoretical side and from the practical side. From the theoretical side, one of the issues would be to extend our compilation approach to other weighted logics, especially those for which the aggregation function at work is not additive. From the practical side, we plan to experiment our *DNNF*-compilation algorithms on the instance X64 of Renault, described in [43] (10813 clauses on 658 variables).

Acknowledgements

Many thanks to the anonymous reviewers for many interesting comments and suggestions. The first author has been partially supported by NSF grant IIS-9988543 and MURI grant N00014-00-1-0617. The second author has been partly supported by the IUT de Lens, the Université d'Artois, the Région Nord/Pas-de-Calais through the IRCICA Consortium, and by the European Community FEDER Program.

References

- [1] E. Amir, S. McIlraith, Partition-based logical reasoning, in: Proc. of the 7th International Conference on Knowledge Representation and Reasoning (KR-00), Breckenridge, CO, 2000, pp. 389–400.
- [2] S. Benferhat, C. Cayrol, D. Dubois, J. Lang, H. Prade, Inconsistency management and prioritized syntax-based entailment, in: Proc. IJCAI-93, Chambéry, France, 1993, pp. 640–645.
- [3] S. Benferhat, D. Dubois, J. Lang, H. Prade, A. Saffiotti, P. Smets, A general approach for inconsistency handling and merging information in prioritized knowledge bases, in: Proc. of the 6th International Conference on Knowledge Representation and Reasoning (KR-98), Trento, Italy, 1998, pp. 466–477.
- [4] S. Benferhat, D. Dubois, H. Prade, How to infer from inconsistent beliefs without revising, in: Proc. IJCAI'95, Montreal, Quebec, 1995, pp. 1449–1455.
- [5] S. Benferhat, S. Kaci, D. Le Berre, M.-A. Williams, Weakening conflicting information for iterated revision and knowledge integration, in: Proc. IJCAI'99, Seattle, WA, 2001, pp. 109–115.
- [6] B. Borchers, J. Furman, A two-phase exact algorithm for MAX-SAT and weighted MAX-SAT problems, *J. Combin. Optim.* 2 (4) (1999) 299–306.
- [7] Y. Bouffkhad, E. Grégoire, P. Marquis, B. Mazure, L. Saïs, Tractable cover compilations, in: Proc. IJCAI'97, Nagoya, Japan, 1997, pp. 122–127.
- [8] M. Cadoli, F.M. Donini, A survey on knowledge compilation, *AI Comm.* 10 (1997) 137–150 (printed in 1998).
- [9] M. Cadoli, F.M. Donini, P. Liberatore, M. Schaerf, The size of a revised knowledge base, *Artificial Intelligence* 115 (1) (1999) 25–64.
- [10] M. Cadoli, F.M. Donini, P. Liberatore, M. Schaerf, Space efficiency of propositional knowledge representation formalisms, *J. Artificial Intelligence Res.* 13 (2000) 1–31.
- [11] M. Cadoli, F.M. Donini, P. Liberatore, M. Schaerf, Preprocessing of intractable problems, *Inform. and Comput.* 176 (2) (2002) 89–120.
- [12] M. Cadoli, F.M. Donini, M. Schaerf, Is intractability of non-monotonic reasoning a real drawback?, *Artificial Intelligence* 88 (1996) 215–251.
- [13] M. Cadoli, F.M. Donini, M. Schaerf, R. Silvestri, On compact representations of propositional circumscription, *Theoret. Comput. Sci.* 182 (1997) 183–202.

- [14] C. Cayrol, M.-C. Lagasque-Schiex, Th. Schiex, Nonmonotonic reasoning: From complexity to algorithms, *Ann. Math. Artificial Intelligence* 22 (3–4) (1998) 207–236.
- [15] S. Coste-Marquis, P. Marquis, Compiling stratified belief bases, in: *Proc. of the 14th European Conference on Artificial Intelligence (ECAI-00)*, Berlin, 2000, pp. 23–27.
- [16] M. Dalal, Semantics of an anytime family of reasoners, in: *Proc. of the 12th European Conference on Artificial Intelligence (ECAI-96)*, Budapest, 1996, pp. 360–364.
- [17] A. Darwiche, A symbolic generalization of probability theory, Ph.D. Thesis, Stanford University, 1992.
- [18] A. Darwiche, Compiling devices: A structure-based approach, in: *Proc. of the 6th International Conference on Knowledge Representation and Reasoning (KR-98)*, Trento, Italy, 1998, pp. 156–166.
- [19] A. Darwiche, Compiling knowledge into decomposable negation normal form, in: *Proc. IJCAI-99*, Stockholm, Sweden, 1999, pp. 284–289.
- [20] A. Darwiche, Decomposable negation normal form, *J. ACM* 48 (4) (2001) 608–647.
- [21] A. Darwiche, On the tractability of counting theory models and its application to belief revision and truth maintenance, *J. Appl. Non-Classical Logics* 11 (1–2) (2001) 11–34.
- [22] A. Darwiche, A compiler for deterministic decomposable negation normal form, in: *Proc. AAAI-02*, Edmonton, AB, 2002, pp. 627–634.
- [23] A. Darwiche, M. Goldszmidt, On the relation between kappa calculus and probabilistic reasoning, in: *Proc. of the 10th Conference on Uncertainty in Artificial Intelligence (UAI-94)*, San Francisco, CA, 1994, pp. 145–153.
- [24] A. Darwiche, P. Marquis, A perspective on knowledge compilation, in: *Proc. IJCAI-99*, Seattle, WA, 2001, pp. 175–182.
- [25] A. Darwiche, P. Marquis, A knowledge compilation map, *J. Artificial Intelligence Res.* 17 (2002) 229–264.
- [26] F. Dupin de St Cyr, Gestion de l'évolutif et de l'incertain en logiques pondérées, Ph.D. Thesis, Université Paul Sabatier, 1996.
- [27] F. Dupin de St Cyr, J. Lang, Th. Schiex, Penalty logic and its link with Dempster–Shafer theory, in: *Proc. of the 10th Conference on Uncertainty in Artificial Intelligence (UAI-94)*, San Francisco, CA, 1994, pp. 204–211.
- [28] R. Dechter, J. Pearl, Tree clustering for constraint networks, *Artificial Intelligence* 38 (3) (1989) 353–366.
- [29] R. Dechter, I. Rish, Directional resolution: The Davis–Putnam procedure, revisited, in: *Proc. of the 4th International Conference on Knowledge Representation and Reasoning (KR-94)*, Bonn, Germany, 1994, pp. 134–145.
- [30] A. del Val, Tractable databases: how to make propositional unit resolution complete through compilation, in: *Proc. of the 4th International Conference on Knowledge Representation and Reasoning (KR-94)*, Bonn, Germany, 1994, pp. 551–561.
- [31] Th. Eiter, Th. Lukasiewicz, Default reasoning from conditional knowledge bases: Complexity and tractable cases, *Artificial Intelligence* 124 (2) (2000) 169–241.
- [32] M. Goldszmidt, Qualitative probabilities: A normative framework for commonsense reasoning, Technical Report R-190, University of California at Los Angeles, Ph.D. Thesis, 1992.
- [33] P. Hansen, B. Jaumard, Algorithms for the maximum satisfiability problem, *Computing* 44 (1990) 279–303.
- [34] J. Kohlas, P.P. Shenoy, Computation in valuation algebras, in: *Algorithms for Uncertain and Defeasible Reasoning*, in: *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, vol. 5, Kluwer Academic, Dordrecht, 2000, pp. 5–39.
- [35] C. Lafage, J. Lang, Logical representation of preferences for group decision making, in: *Proc. of the 7th International Conference on Knowledge Representation and Reasoning (KR-00)*, Breckenridge, CO, 2000, pp. 457–468.
- [36] J. Lang, P. Marquis, Complexity results for independence and definability in propositional logic, in: *Proc. of the 6th International Conference on Knowledge Representation and Reasoning (KR-98)*, Trento, Italy, 1998, pp. 356–367.
- [37] P. Liberatore, Monotonic reductions, representative equivalence, and compilation of intractable problems, *J. ACM* 48 (6) (2001) 1091–1125.
- [38] P. Liberatore, M. Schaerf, Relating belief revision to circumscription (and vice versa), *Artificial Intelligence* 93 (1–2) (1997) 261–296.
- [39] P. Marquis, Knowledge compilation using theory prime implicants, in: *Proc. IJCAI-95*, Montreal, Quebec, 1995, pp. 837–843.

- [40] Y. Moinard, Note about cardinality-based circumscription, *Artificial Intelligence* 119 (2000) 259–273.
- [41] B. Nebel, Belief revision, in: *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, vol. 3, Kluwer Academic, Dordrecht, 1998, pp. 77–145, Chapter: How hard is it to revise a belief base?.
- [42] Ch. Papadimitriou, *Computational complexity*, Addison-Wesley, Reading, MA, 1994.
- [43] B. Pargamin, Vehicle sales configuration: the cluster tree approach, in: *Proc. of the ECAI-02 Workshop on Configuration*, Lyon, 2002.
- [44] B. Pargamin, Extending cluster tree compilation with non-boolean variables in product configuration: A tractable approach to preference-based configuration, in: *Proc. of the IJCAI-03 Workshop on Configuration*, Acapulco, Mexico, 2003.
- [45] G. Pinkas, Propositional nonmonotonic reasoning and inconsistency in symmetric neural networks, in: *Proc. IJCAI-91*, Sydney, Australia, 1991, pp. 525–530.
- [46] G. Pinkas, Reasoning, nonmonotonicity and learning in connectionist networks that capture propositional knowledge, *Artificial Intelligence* 77 (1995) 203–247.
- [47] G. Pinkas, R.P. Loui, Reasoning from inconsistency: a taxonomy of principles for resolving conflict, in: *Proc. of the 3rd International Conference on Knowledge Representation and Reasoning (KR-92)*, Cambridge, MA, 1992, pp. 709–719.
- [48] R. Reiter, A theory of diagnosis from first principles, *Artificial Intelligence* 32 (1987) 57–95.
- [49] R. Reiter, J. de Kleer, Foundations of assumption-based truth maintenance systems: Preliminary report, in: *Proc. AAAI-87*, Seattle, WA, 1987, pp. 183–188.
- [50] R. Schrag, Compilation for critically constrained knowledge bases, in: *Proc. AAAI-96*, Portland, OR, 1996, pp. 510–515.
- [51] B. Selman, H.A. Kautz, Knowledge compilation and theory approximation, *J. ACM* 43 (1996) 193–224.
- [52] Y. Shang, B.W. Wah, Discrete Lagrangian-based search for solving MAX-SAT problems, in: *Proc. IJCAI-97*, Nagoya, Japan, 1997, pp. 377–383.
- [53] W. Spohn, Ordinal conditional functions: A dynamic theory of epistemic states, in: W.L. Harper, B. Skyrms (Eds.), *Causation in Decision, Belief Change, and Statistics*, vol. 2, 1987, pp. 105–134.
- [54] I. Wegener, *The Complexity of Boolean Functions*, Wiley–Teubner, Stuttgart, 1987.