Baltzer Journals March 12, 1996

Nonmonotonic Reasoning: from Complexity to Algorithms

C. Cayrol[1], M.-C. Lagasquie-Schiex[1], T. Schiex[2]

[1] *IRIT, 118 route de Narbonne*
31062 Toulouse Cedex, France
E-mail: {testemal, lagasq}@irit.fr

[2] *INRA, Chemin de Borde Rouge BP 27*
31326 Castanet Tolosan Cedex, France
E-mail: Thomas.Schiex@toulouse.inra.fr

The purpose of this paper is to outline various results regarding the computational complexity and the algorithms of nonmonotonic entailment in different coherence-based approaches. Starting from a (non necessarily consistent) belief base E and a pre-order on E, we first remind different mechanisms for selecting preferred consistent subsets. Then we present different entailment principles in order to manage these multiple subsets. The crossing point of each generation mechanism m and each entailment principle p defines an entailment relation $(E, \leq) \mid\!\sim^{p,m} \Phi$ which we study from the computational complexity point of view. The results are not very encouraging since the complexity of all these nonmonotonic entailment relations is, in most restricted languages, larger than the complexity of monotonic entailment. Therefore, we decided to extend Binary Decision Diagrams techniques, which are well suited to the task of solving NP-hard logic-based problems. Both theoretical and experimental results are described along this line in the last sections.

Topic area: Nonmonotonic reasoning, computational complexity, algorithms and binary decision diagram.

1 Introduction

Formalizing "common sense" reasoning is one of the most important research topics in artificial intelligence. When the available knowledge may be incomplete, uncertain or inconsistent, the classical logic is no more relevant (for example, anything can be classically inferred from inconsistent knowledge bases). Nonmonotonic reasoning is needed. Many researchers have proposed new logics (called nonmonotonic logics) in order to formalize nonmonotonic reasoning, for instance, Reiter's default logic [30]. Others proposed to keep the classical logic in integration with numerical or symbolic structures for ordering the beliefs. In the latter context, we focus on the so-called coherence-based approaches. These approaches handle syntactical belief bases, as in [28]: each belief is a distinct piece of information and only beliefs which are explicitly present in the base are taken into account. It departs from the logical point of view where a base is identified with the set of its models. Due to the belief status of its elements, the belief base is not assumed consistent. Moreover, we assume that the belief base is equipped with a total pre-order (a priority relation) which, contrarily to [16], is not related to semantical entailment. It is equivalent to consider that the base is stratified in a collection of subbases of different priority levels.

In this paper, we are concerned with the deductive aspect of reasoning (cf. [4, 28, 9, 1] for works in the same framework). Following Pinkas and Loui's analysis [29], it is convenient to see coherence-based nonmonotonic entailment as a two-steps procedure which first restores the coherence by generating and selecting preferred belief states (*generation mechanism*) and then manages these multiple states in order to conclude using classical logic (*entailment principle*). For instance, the following kind of inference is considered in [1]: "*E infers Φ iff Φ is classically inferred by all the preferred consistent subsets of E*".

A taxonomy of conflict resolution principles, from credulous to skeptical ones, can be found in [29]. The selection of preferred subsets relies upon the definition of aggregation modes which enable to extend the priority order defined on the initial belief base into a preference relation between subsets (see [1, 9]).

In the framework described above, we propose a comparative study of various coherence-based entailment relations from the point of view of the computational complexity[1]. This topic is essential for practical applications. Indeed, as far as we know, only few papers have been devoted to computational complexity issues for nonmonotonic reasoning. Nebel has thoroughly considered the computational complexity of syntax-based *revision procedures* [28]. Eiter and Gottlob [19, 14] have also considered the case of default logic and abductive procedures.

The paper is organized as follows. First, we present the coherence-based entailment problems under consideration. Starting from a belief base E and a pre-order on E, we present three mechanisms for selecting preferred consistent subsets of E, each one being more selective than the previous one. Then we present three entailment principles in order to manage these multiple subsets: the skeptical principle, the argumentative principle and the credulous principle. The crossing point of each generation mechanism m and each entailment principle p defines an entailment relation $(E, \leq) \mid\!\sim^{p,m} \Phi$. Secondly, after an informal and simplified presentation of the main concepts of the complexity theory, we provide comparative results in the general propositional case. Results are also provided in the three restricted cases of strictly ordered belief bases, of Horn bases (sets of conjunctions of Horn clauses) and of strictly ordered Horn bases. Even in

[1]Cautiousness and validity of deduction rules have been considered in [8].

restricted cases such as strictly ordered belief bases or Horn bases, the results are quite disappointing since the complexity in the worst case remains greater than the complexity of classical entailment unless both restrictions apply simultaneously, an unrealistic restriction. These results inclined us to look for an adapted tool for solving decision problems above NP. In the last part of the paper, we show that Binary Decision Diagrams [6] can be extended for solving some classes of coherence-based entailment problems.

2 Coherence-Based Nonmonotonic Entailment

Throughout the paper, E denotes a non-empty finite set of propositional formulae referred to as the "belief base". E is not assumed to be consistent.

Coherence-based nonmonotonic entailment from a stratified belief base can be described as a two-steps procedure which first restores the coherence by selecting preferred consistent subbases, and then applies classical entailment on some of these preferred subbases according to a so-called entailment principle.

2.1 Selecting Preferred Belief States

The most usual idea for handling inconsistency is to work with maximal (w.r.t. set-inclusion) consistent subsets of E, called *theses* of E in the following.

Definition 1 (Thesis)
A subset X of E is a thesis of E iff X is consistent and there is no consistent subset of E which strictly contains X.

Unfortunately, in the worst case, this approach is not selective enough: too many theses must be taken into account. Now, we assume that E is equipped with a total pre-order $\leq$ (a priority relation). It is equivalent to consider that E is stratified in a collection $(E_1, \ldots, E_n)$ of belief bases, where E_1 contains the formulae of highest priority (or relevance) and E_n those of lowest priority. The pair $(E, \leq)$ is called a *prioritized* (or equivalently *stratified*) belief base. Each E_i is called a *stratum* of E. Different approaches have been proposed to use the priority relation in order to select "preferred" subsets (see [7] for a survey). For the purpose of this paper, we concentrate on the approaches which refine the set-inclusion and lead to select preferred subsets among the theses of E. Indeed, the priority relation on E induces a *preference* relation on the set of subsets of E. Let us first briefly remind the "inclusion-based" preference, which is the most frequently encountered, despite different presentations.

Definition 2 ***(Inclusion-based preference)***
Let $E = (E_1, \ldots, E_n)$ a stratified belief base. Z being a subset of E, Z_i denotes $Z \cap E_i$. The "inclusion-based" preference is the strict order defined on the power set of E by: $X \ll^{\text{INCL}} Y$ iff there exists i such that Y_i strictly contains X_i and for any $j < i$, $X_j = Y_j$.

Note that $\ll^{\text{INCL}}$-preferred theses are also called preferred sub-theories in [4], democratic preferred theses in [9], and exactly correspond to strongly maximal-consistent subbases in [12].

Another way of selecting preferred subsets is to use consistent subsets of maximum cardinality (see [1, 26]).

Definition 3 ***(Cardinality-maximal-consistent subset)***
A subset X of E is a cardinality-maximal-consistent subset of E iff X is consistent and for each consistent subset Y of E, $|Y| \leq |X|$ ($|Y|$: cardinality of Y).

Taking into account the stratification of E leads to the definition of the so-called "lexicographic" preference (see [1, 26]):

Definition 4 ***(Lexicographic preference)***
Let $E = (E_1, \ldots, E_n)$ a stratified belief base. The "lexicographic" preference is the strict order defined on the power set of E by: $X \ll^{\text{LEX}} Y$ iff there exists i such that $|X_i| < |Y_i|$ and for any $j < i$, $|X_j| = |Y_j|$.

It can be shown that the lexicographic preference refines the inclusion-based preference: any $\ll^{\text{LEX}}$-preferred consistent subset of E is an $\ll^{\text{INCL}}$-preferred thesis, but the converse is false as illustrated at the end of this section. Moreover, the associated lexicographic pre-order is total.

Example Consider the following propositional variables:

Variable	Meaning
r	bump when reversing
b	bump at the back of the car
nl	I am not liable for the damage
np	I won't pay the repairs for the car
x	I have got a collision damage waiver
ci	insurance cost will increase

Consider the stratified belief base with the following five strata: $E_1 = \{\rightarrow r; \rightarrow x\}$, $E_2 = \{r \rightarrow b\}$, $E_3 = \{b \rightarrow nl; r, nl \rightarrow\}$, $E_4 = \{nl \rightarrow np; np \rightarrow nl; x \rightarrow np\}$, $E_5 = \{\rightarrow nl, ci\}$. There are three inclusion-based preferred theses:

- $Y_1 = \{\rightarrow r; \rightarrow x; r \rightarrow b; r, nl \rightarrow; nl \rightarrow np; x \rightarrow np; \rightarrow nl, ci\}$,

- $Y_2 = \{\to r; \to x; r \to b; r, nl \to; nl \to np; np \to nl; \to nl, ci\}$,
- $Y_3 = \{\to r; \to x; r \to b; b \to nl; nl \to np; np \to nl; x \to np; \to nl, ci\}$.

However, Y_3 is the only one lexicographic preferred thesis (indeed, $Y_1 \ll^{\text{LEX}} Y_3$ and $Y_2 \ll^{\text{LEX}} Y_3$).

2.2 Three Entailment Principles

In the previous section, we have presented three mechanisms for producing a set of consistent belief states from the initial prioritized belief base $(E, \leq)$. In the following, we call T the mechanism which produces the set of theses of E (maximal-consistent subsets), INCL the mechanism which produces the inclusion-based preferred theses of E and LEX the refinement which produces the set of preferred theses for the lexicographic order. A taxonomy of numerous entailment principles has been established by Pinkas and Loui [29] according to their cautiousness. Here, we are interested in three of them which we now briefly present. We start from a set of consistent subsets of E denoted by $m(E)$ in the following (for instance, m is one of the generation mechanisms T, INCL, LEX). Let Φ be a propositional formula.

Definition 5 (Skeptical entailment principle)
Φ is inferred from $m(E)$ according to the skeptical entailment principle iff Φ can be classically inferred from each element of $m(E)$. This entailment principle, often called strong entailment or universal entailment in the literature, will be denoted by $\forall$ and referred to as the UNI *principle in the following.*

Definition 6 (Credulous entailment principle)
Φ is inferred from $m(E)$ according to the credulous entailment principle iff Φ can be classically inferred from at least one element of $m(E)$. This entailment principle, often called weak entailment or existential entailment in the literature, will be denoted by $\exists$ and referred to as the EXI *principle in the following.*

These two entailment principles are the most commonly activated in presence of multiple conflicting belief states. Obviously, the UNI principle is more cautious than the EXI principle, since each conclusion obtained from $m(E)$ by UNI inference is also obtained by EXI inference. Since the EXI principle leads to unsafe consequence relations (*i.e.* pairwise contradictory conclusions may be produced), an intermediary principle has been considered, which consists in keeping only the weak consequences whose negation cannot be inferred (see [2] for a discussion on the so-called argumentative inference).

Definition 7 ***(Argumentative entailment principle)***
Φ is inferred from $m(E)$ according to the argumentative entailment principle iff Φ is classically inferred from at least one element of $m(E)$ and no element of $m(E)$ classically entails $\neg\Phi$. This entailment principle will be denoted by A and referred to as the ARG *principle in the following.*

We are now ready to give a precise definition of the entailment relations and the associated problems which we will consider from the computational complexity point of view. Each one appears at the crossing point of a belief state generation mechanism m and an entailment principle p. Let $(E, \leq)$ be the initial belief base and Φ a propositional formula.

Definition 8 ***(*T*-based problems)***
The problem UNI-T *(resp.* EXI-T*,* ARG-T*) is defined by "verify that Φ is a strong (resp. weak, argumentative) consequence of E using the theses of E". The* T *generation mechanism is used.*
Notation: $E \mid\!\sim^{\forall(resp.\ \exists,A),\mathrm{T}} \Phi$ *for* UNI-T *(resp.* EXI-T*,* ARG-T*).*

In the above notation, it is sufficient to mention E instead of $(E, \leq)$ since producing the theses makes no use of the pre-order on E.

Definition 9 ***(*Incl*-based problems)***
The problem UNI-INCL *(resp.* EXI-INCL*,* ARG-INCL*) is defined by "verify that Φ is a strong (resp. weak, argumentative) consequence of E using the inclusion-based preferred theses of E". The* INCL *generation mechanism is used.*
Notation: $(E, \leq) \mid\!\sim^{\forall(resp.\ \exists,A),\mathrm{INCL}} \Phi$ *for* UNI-INCL *(resp.* EXI-INCL*,* ARG-INCL*).*

Definition 10 ***(*Lex*-based problems)***
The problem UNI-LEX *(resp.* EXI-LEX*,* ARG-LEX*) is defined by "verify that Φ is a strong (resp. weak, argumentative) consequence of E using the lexicographic preferred theses of E". The* LEX *generation mechanism is used.*
Notation: $(E, \leq) \mid\!\sim^{\forall(resp.\ \exists,A),\mathrm{LEX}} \Phi$ *for* UNI-LEX *(resp.* EXI-LEX*,* ARG-LEX*).*

The inclusion-based preference and the lexicographic preference are induced by the pre-order $\leq$ (see respectively Def. 2 and Def. 4).

Example Applying the above principles on the example of the previous section produces:

- $(E, \leq) \mid\!\sim^{\forall,\mathrm{INCL}} b$, $(E, \leq) \mid\!\sim^{\exists,\mathrm{INCL}} nl$ and $(E, \leq) \mid\!\sim^{A,\mathrm{INCL}} ci$,
- $(E, \leq) \mid\!\sim^{\forall,\mathrm{LEX}} np$ which is equivalent to $(E, \leq) \mid\!\sim^{\exists,\mathrm{LEX}} np$ and which is equivalent to $(E, \leq) \mid\!\sim^{A,\mathrm{LEX}} np$ since there is only one $\ll^{\mathrm{LEX}}$-preferred subset.

3 Computational Complexity

We consider that the readers have some knowledge about computational complexity (else, see [17]). In this paper, we are interested only by decision problems (each instance of a decision problem has either a "yes" or a "no" answer), and we note the fact that a polynomial transformation exists from Q' to Q by $Q' \propto Q$ (this informally means that Q is at least as hard as Q'). In this work, we use classes from the polynomial hierarchy (called PH), each of them containing supposedly harder and harder problems. This PH is defined inductively using the notion of oracle. An oracle of complexity X may be viewed as a subroutine which solves any problem of complexity X. Each call to an oracle is counted as one time unit. So, there are polynomial problems using an oracle of complexity X and non-deterministic polynomial problems using an oracle of complexity X. They define respectively the P^X and NP^X classes. PH is defined by the set of classes $\{\Delta_k^p, \Sigma_k^p, \Pi_k^p \text{ for } k \geq 0\}$:

- $\Sigma_0^p = \Delta_0^p = \Pi_0^p = \mathsf{P}$
- $\Delta_{k+1}^p = \mathsf{P}^{\Sigma_k^p}$
- $\Sigma_{k+1}^p = \mathsf{NP}^{\Sigma_k^p}$
- $\Pi_{k+1}^p = \text{co-}\Sigma_{k+1}^p$

In each of these classes, we also have the notion of completeness (a Σ_k^p-complete problem being harder than any Σ_k^p problem). The conjecture $\mathsf{NP} \neq \mathsf{P}$ is generalized to the PH with the following stronger conjectures: $\mathsf{NP} \neq \text{co-}\mathsf{NP}$ and $\forall k, \Sigma_k^p \neq \Pi_k^p$. Note that $\mathsf{NP} = \mathsf{P}$ implies that the PH collapses into P. The problem stated below, called 2-QBF also denoted $\exists a \forall b H(a,b)$, is an example of a Σ_2^p-complete problem (see [22, 28]).

> **Instance**: A propositional formula $H(a,b)$ where a and b denote sets of propositional variables: $a = \{a_1, \ldots, a_n\}$ and $b = \{b_1, \ldots, b_m\}$.
> **Question**: Is there a truth assignment of the variables in a such that $H(a,b)$ is true for any truth assignment of the variables in b?

3.1 Complexity of General Entailment Relations

We consider entailment relations of the form $(E, \leq) \mid\!\sim^{p,m} \Phi$ where E, $\leq$, p and m have been defined in the previous sections, and where Φ is a single propositional formula. The complexity results for the general propositional case are given in Table 1. For lack of space, we just give sketches of proof. The detailed proofs are given in [7, 24]. For each problem Q, the complexity proof is done in two steps:

- first, we exhibit an algorithm which solves Q and whose complexity class is X (class membership proof which gives an upper bound for the complexity);
- then, we prove that Q is X-complete by giving a polynomial transformation from an X-complete problem to Q (or else give any other lower bound for the complexity).

p	m	Complexity class
UNI	T	Π_2^p-complete
EXI	T	Σ_2^p-complete
ARG	T	$\Delta_3^p - (\Sigma_2^p \cup \Pi_2^p)$ if $\Sigma_2^p \neq \Pi_2^p$
UNI	INCL	Π_2^p-complete
EXI	INCL	Σ_2^p-complete
ARG	INCL	$\Delta_3^p - (\Sigma_2^p \cup \Pi_2^p)$ if $\Sigma_2^p \neq \Pi_2^p$
UNI	LEX	Δ_2^p-complete
EXI	LEX	Σ_2^p-complete
ARG	LEX	Δ_3^p, Σ_2^p-hard

Table 1: Complexities in the general propositional case

Proofs for strong relations:

For UNI-T and UNI-INCL, we use the results of Nebel in [28], because these entailment relations correspond to the SBR and PBR revision procedures for which Nebel has proved the Π_2^p-completeness. So, UNI-T and UNI-INCL are Π_2^p-complete.

For UNI-LEX, we prove the membership to Δ_2^p using the following idea: if we want to check if Φ is classically entailed by all the lexicographic preferred theses of E, we may insert $\neg\Phi$ alone in a new least prioritary stratum. This defines a new base E'. Since all lexicographic preferred theses have always the same cardinality at each stratum, Φ will be entailed by all the lexicographic preferred theses of E iff any lexicographic preferred thesis of E' has a zero cardinality in the last stratum. The algorithm 1 sophisticates this idea by introducing $\Phi \rightarrow \ell$ (where ℓ is new variable) in the most prioritary stratum and $\neg\ell$ in a new least prioritary stratum to avoid a possible interference with an already existing occurrence of $\neg\Phi$ in E.

Algorithm 1: UNI-LEX $((E, \leq), \Phi)$
begin
  $E' \leftarrow \{\Phi \rightarrow \ell\} \cup E \cup \{\neg\ell\}$
  $k \leftarrow < 0, 0, \ldots, 0 >$ *; k is a n'-vector (n'=number of strata in E')*
  for $n_s = 1$ to n' **do**
    $n_f \leftarrow$ number of formulae in the stratum E'_{n_s}
    End? $\leftarrow$ false
    while $(n_f \geq 0)$ and (not End?) **do**
      $k[n_s] \leftarrow n_f$
      if MAX-GSAT-ARRAY(E', k) **then** End? $\leftarrow$ true
      else $n_f \leftarrow n_f - 1$
  Verify that $k[n'] \neq 1$
end

In this algorithm, we use an oracle MAX-GSAT-ARRAY defined by:

> **Instance**: A pre-ordered set $(Y, \leq)$ of propositional formulae, an array k of dimension n with n=number of strata in Y.
> **Question**: Is there a truth assignment which satisfies at least $k[i]$ formulae for each stratum i of Y?

This problem is NP-complete (NP class membership is obvious, completeness is proved by restriction to SAT). Therefore the previous algorithm (and its dichotomic version too) is deterministic polynomial and uses a non-deterministic polynomial oracle. So, UNI-LEX belongs to the class Δ_2^p.

The Δ_2^p-completeness for UNI-LEX is proved using a Δ_2^p-complete problem defined in [15] and referred to as ALM in the following:

> **Instance**: Let $C = \{C_1, \ldots, C_m\}$ be a satisfiable set of clauses, let the set of propositional variables of C denoted by $X = \{x_1, \ldots, x_n\}$, let a prioritization of X denoted by $O(X) =< x_1, \ldots, x_n >$.
> **Question**: Let V_M be the truth assignment lexicographically maximal with respect to $O(X)$ satisfying C, does V_M fulfill $V_M(x_n)$ = true?

Proofs for weak relations:

We consider the algorithm 2 for the EXI-m problems ($\forall m \in \{\mathrm{T}, \mathrm{INCL}, \mathrm{LEX}\}$).

Algorithm 2: EXI-m $((E, \leq), \Phi)$
begin
  Guess a subset Y of $(E, \leq)$
  Verify that Y is:
    - a thesis (for EXI-T)
    - an inclusion based preferred thesis (for EXI-INCL)
    - a lexicographic preferred thesis (for EXI-LEX)
  Verify that Y classically entails Φ
end

First of all, note that "*verify that Y classically entails* Φ" is co-NP-complete. Then, all the oracles "*verify that Y is ...*" are non-deterministic polynomial time oracles. Note that the oracle used for the lexicographic case solves the following problem (called MAX-GSAT-STRICT) which is NP-complete:

> **Instance**: A set Y of propositional formulae, an integer $k \leq |Y|$.
> **Question**: Is there a consistent subset Y' of Y such that $|Y'| > k$?

Therefore, EXI-m belongs to the class $NP^{NP} = \Sigma_2^p$. Moreover, the completeness proofs are the following:

- For Exi-T, we consider the following polynomial transformation from 2-Qbf to Exi-T: let "$\exists a \forall b H(a,b)$" be an instance of 2-Qbf, the instance of Exi-T is defined by $E = \{a_1, \ldots, a_n, \neg a_1, \ldots, \neg a_n\}$ and $\Phi = H(a,b)^2$.
- For Exi-Incl, the completeness is obvious, since Exi-T is a restriction of Exi-Incl.
- For Exi-Lex, we may use the previous proof for Exi-T since any thesis of E, when E is of the form $\{a_1, \ldots, a_n, \neg a_1, \ldots, \neg a_n\}$ is also a lexicographic preferred thesis of E.

Proof for argumentative relations:

$\forall m \in \{\mathrm{T}, \mathrm{Incl}, \mathrm{Lex}\}$, the Arg-$m$ problems can be solved by the algorithm 3.

Algorithm 3: Arg-m $((E, \leq), \Phi)$
begin
  Verify that $(E, \leq) \not\mid\!\sim^{\exists,m} \neg\Phi$
  Verify that $(E, \leq) \mid\!\sim^{\exists,m} \Phi$
end

This algorithm is deterministic polynomial and uses a Σ_2^p oracle solving Exi-m. Therefore, we conclude that $\forall m$, Arg-m belongs to the class $P^{\Sigma_2^p} = \Delta_3^p$.

We cannot prove Δ_3^p-completeness for any of these problems, but we refine the class membership, as in [28]. Indeed, we prove that most of the Arg-m problems are in $\Delta_3^p - (\Sigma_2^p \cup \Pi_2^p)$. So, we have the following lower bounds:

- For Arg-T, we prove that there is a polynomial transformation from Exi-T to Arg-T and there is a polynomial transformation from co-Exi-T to Arg-T. Therefore, both Exi-T and co-Exi-T can be polynomially transformed to Arg-T. Since Exi-T is Σ_2^p-complete and co-Exi-T is Π_2^p-complete, assuming that Arg-T $\in (\Sigma_2^p \cup \Pi_2^p)$ would lead to $\Sigma_2^p = \Pi_2^p$.
- For Arg-Incl, we still rely on the fact that Arg-T is a restriction of Arg-Incl: Arg-T $\propto$ Arg-Incl. Since Exi-T $\propto$ Arg-T and co-Exi-T $\propto$ Arg-T, we obtain the same conclusion as for Arg-T.
- For Arg-Lex, similarly as for Arg-T, it is possible to prove that Exi-Lex $\propto$ Arg-Lex. However, we haven't found a polynomial transformation from co-Exi-Lex (or any other Π_2^p-complete problem) to Arg-Lex. We simply conclude that Arg-Lex is Σ_2^p-hard.

3.2 Complexity for Strictly Ordered Bases

[2] This result is not surprising. In [15], Eiter and Gottlob define an abductive problem which is Σ_2^p-complete and which may be polynomially transformed to Exi-T.

p	m	Strictly stratified bases	Horn bases	Strictly stratified Horn bases
Uni	T	not concerned	co-NP-complete	not concerned
Exi	T	not concerned	NP-complete	not concerned
Arg	T	not concerned	$\Delta_2^p - (NP \cup \text{co-NP})$ if $NP \neq$ co-NP	not concerned
Uni	Incl	Δ_2^p-complete	co-NP-complete	P
Exi	Incl	Δ_2^p-complete	NP-complete	P
Arg	Incl	Δ_2^p-complete	$\Delta_2^p - (NP \cup \text{co-NP})$ if $NP \neq$ co-NP	P
Uni	Lex	Δ_2^p-complete	Δ_2^p-complete	P
Exi	Lex	Δ_2^p-complete	Δ_2^p	P
Arg	Lex	Δ_2^p-complete	Δ_2^p	P

Table 2: Complexities in three restricted cases

We first consider the case of a strictly ordered belief base (see results in Table 2). In this case E is stratified with exactly one formula per stratum. The complexity of the problems p-T (for p in {Uni, Exi, Arg}) is not affected by this restriction since the pre-order on the belief base is not taken into account by the generation mechanism T. We will show that all the other problems become equivalent to a single problem.

Theorem 1
Let $<$ be a strict total order on E. There is only one $\ll^{\text{Incl}}$-preferred thesis, which is also the only one $\ll^{\text{Lex}}$-preferred thesis (proof in [7]).

Corollary 1
The problems Uni-Incl *(resp.* Lex*),* Exi-Incl *(resp.* Lex*),* Arg-Incl *(resp.* Lex*) are equivalent to a single problem called* 1/Stratum.

The class membership is proved by the algorithm 4. This algorithm (and its

Algorithm 4: 1/Stratum $((E, \leq), \Phi)$
begin
  $X \leftarrow \varnothing$
  $n_s \leftarrow 1$ (current stratum)
1 **if** $X \cup E_{n_s}$ is consistent **then** $X \leftarrow X \cup E_{n_s}$
  $n_s \leftarrow n_s + 1$
  if $n_s =$ (total number of strata in E) **then** verify that $X \vdash \Phi$
  else go to step 1
end

dichotomic version too) is deterministic polynomial and relies on an NP-complete oracle. Therefore, 1/STRATUM is in Δ_2^p. Then, using the same transformation as in the proof of UNI-LEX, we prove that 1/STRATUM is Δ_2^p-complete, since the belief base E considered in that transformation is a strictly ordered base.

3.3 Complexity for Horn Bases

In this section, we assume that the belief base is a finite set of conjunctions of propositional Horn clauses[3] and the formula Φ is also a conjunction of Horn clauses (see results in Table 2). In this case, we remind the reader that both SAT and the entailment problem in classical propositional logic become polynomial. Once again, the UNI-LEX problem is quite specific: its complexity seems unchanged in case of Horn bases while most other problems shift down by one level in PH.

Proofs for m = T: We may still use the previously stated algorithms. Using the fact that the complexity of the entailment problem is reduced, we conclude that UNI-T-HORN is in co-NP, EXI-T-HORN is in NP and ARG-T-HORN is in Δ_2^p. Then, the completeness proofs are the following:

- For UNI-T-HORN, we use an idea previously proposed in [14]: SAT can be polynomially transformed to co-UNI-T-HORN (SAT is the satisfiability problem for any set of clauses). So, UNI-T-HORN is co-NP-complete.
- For EXI-T-HORN, the proof is the same than the one for UNI-T-HORN.
- For ARG-T-HORN, we cannot keep the ARG-T proof, because our polynomial transformations from EXI-T and from co-EXI-T to ARG-T do not preserve Horn clauses. So, we consider a new problem EXI-T-HORN-POS:

  Instance: E a Horn base, ℓ a positive literal.
  Question: Is it true that $E \mid\!\sim^{\exists,T} \ell$?

  It is clear that this problem is NP-complete (see the EXI-T-HORN proof). Therefore, we may use the polynomial transformations defined for ARG-T on EXI-T-HORN-POS and co-EXI-T-HORN-POS. We conclude that ARG-T-HORN is in $\Delta_2^p - (NP \cup \text{co-NP})$ if $NP \neq \text{co-NP}$.

Proof for m = INCL: We may use the algorithms previously considered in the unrestricted case. All the polynomial transformations we used preserve Horn clauses and we conclude that UNI-INCL-HORN is co-NP-complete, EXI-INCL-HORN is NP-complete and ARG-INCL-HORN is in $\Delta_2^p - (NP \cup \text{co-NP})$ if $NP \neq \text{co-NP}$.

Proof for m = LEX: Obviously, UNI-LEX is still a member of Δ_2^p. For EXI-LEX we prove the membership to Δ_2^p by using the following idea: we first compute

[3]When E and Φ are CNF formulae (conjunctive normal form), the complexity results remain unchanged w.r.t. the general case.

the n-vector k which contains the cardinalities per stratum of a lexicographic preferred thesis (this uses a polynomial number of calls to the oracle MAX-GSAT-ARRAY[4] , see algorithm 1). Then, we may guess a subbase, check that it is consistent, lexicographic preferred (using k) and that it entails Φ. All these tests are polynomial and this corresponds to one call to an NP oracle: this algorithm proves Δ_2^p membership. ARG-LEX can simply be solved by an EXI-LEX and a co-EXI-LEX call and is therefore also in Δ_2^p.

Then, the completeness proofs are the following:

- For UNI-LEX-HORN, we prove Δ_2^p-completeness using a Δ_2^p-complete problem defined in [15] and referred to as ACM in the following:

  Instance: $C = \{C_1, \dots, C_m\}$ a set of clauses, $X = \{x_1, \dots, x_n\}$ the variables of C, $k \in \{1, \dots, m\}$ an integer.
  Question: Let V a truth assignment cardinality-maximal of X, does V fulfill $V(C_k) = \text{true}$?

- For EXI-LEX-HORN and for ARG-LEX-HORN, we have neither proved completeness nor refined the class membership result.

3.4 Complexity for Strictly Ordered Horn Bases

We may use Theorem 1. So, in the case of a strictly ordered Horn base, the problems UNI-INCL (resp. LEX), EXI-INCL (resp. LEX), ARG-INCL (resp. LEX) are equivalent to a single problem called 1/STRATUM-HORN, and we may use the algorithm previously defined for 1/STRATUM, which used an oracle for SAT. In the case of a Horn base, this oracle becomes deterministic polynomial. Therefore, 1/STRATUM-HORN is in P (see Table 2).

3.5 Conclusion on Complexity

All the previous results are presented in Tables 1 and 2. These results are very discouraging in the sense that the few polynomial classes are incredibly restrictive and of poor practical interest while most other problems are located between Δ_2^p and Δ_3^p. One appealing relation is the UNI-LEX relation, which is "only" Δ_2^p-complete in the general case, but its complexity is mostly unaffected by the restrictions we considered.

Therefore, rather than focusing on unrealistic polynomial classes, we have chosen to directly tackle problems in the PH using adapted algorithms. Local search algorithms have recently shown promising results on large hard random instances of SAT, but all these algorithms focus on the search of a polynomial

[4]Note that MAX-GSAT-ARRAY remains NP-complete when restricted to Horn clauses: NP-membership is obvious, NP-completeness is proved using a polynomial transformation from MAX-2SAT (see [17]).

length certificate and seem therefore useless for tackling problems which are above NP, for which no polynomial length certificate exists (unless P=NP). In the next section, we show how Binary Decision Diagrams (which are routinely used in the field of digital-system design and testing, for solving problems above NP) may be applied to decide some of the previous problems in the general case.

4 Binary Decision Diagrams for Nonmonotonic Logics

We first rapidly introduce the main principles of *Binary Decision Diagrams* (or BDD). Detailed presentations of BDD can be found in [5, 6]. Given a formula ϕ on a set of variables V, a BDD [25, 6] represents the formula ϕ using a labeled directed acyclic graph (DAG). The graph has one source and two sink vertices labeled 0 and 1 representing the boolean constants 0 and 1 respectively. Each non-sink vertex is labeled with a boolean variable $v \in V$ and has two out-edges labeled *then* and *else*. The *then* child corresponds to the case where $v = 1$ and the *else* child to the case where $v = 0$ and a path from the source to a sink therefore defines a truth assignment. The idea, which extends the coding principle of decision trees, is that paths from the source to sink 1 (resp. 0) represent truth assignments that satisfy ϕ (resp. violate ϕ).

Given an order on the set of the variables V that occur in ϕ, an ordered BDD is a BDD such that all paths from the source to a sink visit the variables in an ascending order. Finally, a reduced ordered BDD (or ROBDD for short) may be defined as a compressed decision tree for the formula. The decision tree may be transformed into the ROBDD by iteratively applying two reduction rules until quiescence (each rule application lowers the number of vertices by one – see Figure 1, taken from [6]):

- redundant vertices, such that the two out-edges point to the same vertex, are simply bypassed and deleted;
- pairs of vertices that denote the same function *i.e.*, with the same label and the same *then* and *else* children (if any), are merged.

Under a given variable order, the final ROBDD R_ϕ obtained is unique and canonical in the sense that two equivalent formulae define two identical ROBDDs. For example, an unsatisfiable formula will always reduce to the sink 0 while a tautology will always reduce to the sink 1. There is no restriction on the formulae represented (CNF or not).

Property 1
Let R_Φ be the ROBDD that represents Φ under a given order. Then:

x1	x2	x3	ϕ
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

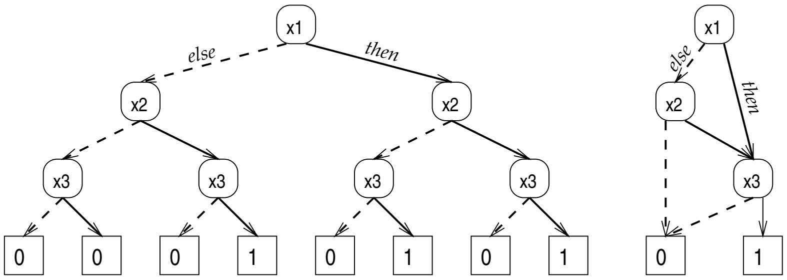

Figure 1: From left to right: the truth table of the formula ϕ, the associated decision tree and the reduced ordered BDD. A dotted edge corresponds to an assignment to 0, a solid edge to 1

- *any path P from the source of the ROBDD to the sink* 1 *defines a partial truth assignment ω_P such that all the complete truth assignments that contain ω_P satisfy Φ;*
- *for any complete truth assignment ω that satisfies Φ there exists one and only one path P from the source of the ROBDD to the sink* 1 *such that ω_P is included in ω.*

In Figure 1, there are only two paths from the source to sink 1: $x_1 = 0, x_2 = 1, x_3 = 1$ and $x_1 = 0, x_3 = 1$. The first path directly defines one model of ϕ while the second implicitly defines two models, depending on x_2 assignment.

The main advantage of ROBDDs w.r.t. decision trees is the compression which is achieved by the two previous reduction rules. Actually, the size of the ROBDD of a formula is not necessarily exponential in the number of the variables (whereas the number of vertices in a decision tree is always equal to $2^{|V|+1} - 1$). In practice, the actual size of a ROBDD largely depends on the order of the variables used and though the theoretical worst case space complexity remains exponential, "there has been ample empirical evidence that many functions encountered in real applications can be represented efficiently as ROBDDs" [5, 6].

Most operations on ROBDDs rely on a single essential operator: the so-called *if-then-else* or ITE operator. This operator applies to a boolean variable x and to two ROBDDs R_f and R_g representing the formulae f and g respectively. $\text{ITE}(x, R_f, R_g)$ returns the ROBDD representing the formula $(x \wedge f) \vee (\neg x \wedge g)$ in constant time [3]. The implementation of ITE garantees that no duplicate vertex (same label and children) will ever be created and therefore that a ROBDD will effectively be built. In practice, instead of reducing a huge decision tree, this ITE operation is used repeatedly to build ROBDDs incrementally: given the ROBDD representations R_ϕ and R_ψ of the two formulae ϕ and ψ, the ROBDD representation for the formula $(\phi\langle op\rangle\psi)$, where $\langle op\rangle$ is any binary boolean operator, can be computed in time $O(S_\phi \times S_\psi)$, where S_f denotes the number of vertices in the BDD R_f representing f. This again emphasizes the importance of the size of a ROBDD and therefore of the problem of finding "good" variable orders.

4.1 Tackling the UNI-LEX *Problem*

In the following, we show how ROBDDs may be used to solve the UNI-LEX decision problem. Consider a stratified belief base $E = \{\phi_i\}_i$ and V the set of variables appearing in the formulae ϕ_i. Since E is supposedly inconsistent, the ROBDD which represents E will simply reduce to the sink 0, which is not very interesting. To bypass the inconsistency, we introduce one new "assumption" variable ℓ_{ϕ_i} per formula in E and we consider a new belief base where each formula ϕ_i from E is replaced by the formula $\ell_{\phi_i} \rightarrow \phi_i$. This process, which is used in the ATMS of De Kleer (see [23]) and which has also been suggested to solve "Dynamic Constraint Satisfaction Problems" in [11, 20, 21], yields a belief base which is obviously consistent.

Let A be the set of all the assumption variables introduced and $E_A = \{\ell_{\phi_i} \rightarrow \phi_i \mid \phi_i \in E\}$. For a given truth assignment ω on $(V \cup A)$, we note ω_V (resp. ω_A) the restriction of ω to V (resp. A). Any truth assignment ω_A of A defines one subbase of E, namely the subbase which contains all the formulae ϕ_i of E such that $\omega_A(\ell_{\phi_i}) = 1$. This subbase will be denoted $Base(\omega_A)$. Obviously, a truth assignment ω of $(V \cup A)$ is a model of E_A iff ω_V is a model of the subbase $Base(\omega_A)$ and therefore each model of E_A corresponds to a consistent subbase of E plus one of its model, and vice-versa. Therefore, to identify $\ll^{\text{LEX}}$-preferred subbases among all the consistent subbases, it will be sufficient to identify "preferred" models of E_A. To enable us to later use shortest path algorithms in the BDD, we first build a weighting function that encodes the LEX preference.

Definition 11 (Weight of a subbase)
We associate a weight (a positive integer) to each formula ϕ_i in E. The weight $w(B)$ of a subbase B of E is defined as the sum of the weights of all the formulae in $E \setminus B$.

To encode the LEX preference, the weight associated to a formula ϕ_i will be related to the stratum in which ϕ_i appears in E (see in [13]):

- for the least prioritary stratum n, this weight $w(n)$ is equal to 1;
- for other strata, it is inductively defined by:

$$w(i) = 1 + \sum_{i<j\leq n} [w(j) \times |E_j|] = w(i+1) \times (|E_{i+1}| + 1)$$

which guarantees that the weight of a formula in stratum i is strictly larger than the sum of all the weights of all the formulae which appear in less important strata. This inductive definition reduces to:

$$w(i) = \prod_{j=n-1}^{i} \left[1 + |E_{j+1}|\right]$$

Theorem 2
A subbase B is $\ll^{\mathrm{LEX}}$-preferred to a subbase C iff the weight of B is lower than the weight of C.

The proof is given in the appendix. We now consider the ROBDD R_{E_A} that represents the conjunction of the formulae in E_A. For each vertex which is labeled by an assumption ℓ_{ϕ_i}, we weight the *else* edge of the vertex with the weight associated to the formula ϕ_i. As usual, the length of a path is defined as the sum of all the weights of the edges in the path (non weighted edges count for nothing in that sum).

Theorem 3
A formula Φ is UNI-LEX *entailed by a stratified base E iff all the models ω_V represented in a shortest path of the ROBDD representing E satisfy Φ.*

The proof is given in the appendix. This approach is related to [20, 21], which relates shortest paths in a ROBDD to least cost assignments of constraint satisfaction problems. As indicated in [20, 21], Bellman's linear time algorithm (see [10], section 25.4) can then be applied to a ROBDD in order to compute the length of the shortest path from each vertex in the ROBDD to the sink 1.

We have enhanced this algorithm in order to simultaneously build a new ROBDD R'_{E_A} that contains only the shortest paths from the source to 1 (all non shortest paths are simply redirected to sink 0). The ROBDD obtained is called the "fat-free" version of the initial ROBDD. The modified algorithm remains linear and consists in applying a single procedure called REMOVE-FAT to each vertex of the ROBDD, from the sinks to the source, using a topological order[5]. The procedure applied is described in algorithm 5. Beyond all the usual data-structures used in ROBDDs and the ITE function which is the core of all ROBDD packages (see [3]), we use two simple data-structures:

- we associate an integer variable $\mathsf{Length}(v)$ to each vertex v in the ROBDD; this variable represents the length of the shortest path from the vertex v to the sink 1; initially, the Length associated to the sink 0 and 1 are set to $+\infty$ and 0 respectively;

[5]In practice, the algorithm implemented uses a depth-first post-order search algorithm, intermediate results being cached at each node in Length and Fat-Free to keep a linear time complexity.

- then, we also associate a pointer variable Fat-Free(v) to each vertex v; this variable points to the "fat-free" version of the ROBDD rooted at vertex v; initially, the Fat-Free variables of sink 0 and 1 point to sink 0 and 1 respectively.

Algorithm 5: REMOVE-FAT(R)

; *In the algorithm, when a ROBDD is used as an argument instead of a vertex,*
; *one should understand that the root of the ROBDD is the actual argument.*

begin
  $t \leftarrow$ Length(*then* child of R)
  $e \leftarrow$ Length(*else* child of R)
  if label of the root of R is an assumption variable ℓ **then** $r \leftarrow$ weight of ℓ
  else $r \leftarrow 0$
  $e \leftarrow (e + r)$
  Length(R) $\leftarrow \min(t, e)$
  if $t >$ Length(R) **then** $n_t \leftarrow$ sink vertex 0
  else $n_t \leftarrow$ Fat-Free(*then* child of R)
  if $e >$ Length(R) **then** $n_e \leftarrow$ sink vertex 0
  else $n_e \leftarrow$ Fat-Free(*else* child of R)
  Fat-Free(R) $\leftarrow$ ITE (the label of the root of R, n_t, n_e)
end

The ROBDD R'_{E_A} seems useful to solve all LEX based problems. In the specific case of UNI-LEX, our Theorem 3 shows that we can reduce the problem of UNI-LEX entailment of a formula Φ to a problem of classical entailment of Φ by this ROBDD. Actually, we can still improve things by noting that all the informations on assumption variables conserved in R'_{E_A} are now useless. Therefore, we build from R'_{E_A}, a third ROBDD, denoted R''_{E_A}, which is simply obtained by existential quantification of all assumption variables: the paths from the source to sink 1 in R''_{E_A} represent all truth assignments ω_V on V such that there *exists* a truth assignment ω_A of the assumptions that can extend ω_V to a model represented in R'_{E_A} *i.e.*, all the models of $\ll^{\text{LEX}}$-preferred subbases. Therefore, $(E, \leq) \mathrel{|\!\sim}^{\forall,\text{LEX}} \Phi$ iff the ROBDD R''_{E_A} classically entails Φ.

4.2 "Good" Variable Orders

Given any order on V, [20, 21] shows that inserting all the assumption variables after V gives a space complexity for R_{E_A} which is guaranteed to be lower than $[2^n \times (m+1) - 1]$ non terminal vertices. This yields a worst case space complexity in $O(2^{|V|} \times |A|)$, a much better result than the obvious $O(2^{|V|+|A|})$. We propose instead, given any initial order on V, to insert the variable $\ell_{\phi_i} \in A$ just after all the variables that appear in ϕ_i. This order is used in our simple example, at the end of the section. Let n_ℓ be the number of non-assumption variables before assumption ℓ in the resulting order, we prove the following result:

Theorem 4
The number of non terminal vertices of the BDD R_{E_A} using this order is less than $(2^n - 1) + \sum_{\ell \in A} 2^{n_\ell}$

The proof is given in the appendix. This new bound is always lower than the bound given in [20, 21]. The important thing is that this theoretical improvement is accompanied by large improvements in the actual size of the ROBDD R_{E_A} in practice (see Section 5).

The use of this order and the ROBDDs R_{E_A}, R'_{E_A} and R''_{E_A} are illustrated on a simple version of the penguin problem defined by the belief base E with 2 strata and 4 formulae: $\phi_1 = p, \phi_2 = p \rightarrow b, \phi_3 = p \rightarrow \neg f, \phi_4 = b \rightarrow f$ with ϕ_1 and ϕ_2 are in the most important stratum. The initial order on V is defined by $p \prec b \prec f$. The four assumption variables ℓ_{ϕ_1} to ℓ_{ϕ_4} are inserted using our order process, yielding the final order $p \prec \ell_{\phi_1} \prec b \prec \ell_{\phi_2} \prec f \prec \ell_{\phi_3} \prec \ell_{\phi_4}$.

Applying our weighting process, we get $w(2) = 1$ and $w(1) = 3$. The Figure 2 successively presents the ROBDD R_{E_A}, with bold numbers indicating weight on dashed *else* edges and italic number indicating the length of the shortest path from each vertex to sink 1. The ROBDD R'_{E_A}, obtained from the previous BDD by applying the algorithm 5, which redirects all non shortest path edges to the sink 0. Finally, the ROBDD R''_{E_A} is obtained after existential quantification on the ℓ_{ϕ_i} variables. When our variable order or [21]'s order is used, an assumption is always ordered after all the variables of the associated formula. In this case, we can prove the Property 2 (see proof of Theorem 4 in the appendix).

Property 2
The then *children of all the assumption vertices in R'_{E_A} are the sink vertex 0.*

This property makes the existential quantification operation very easy: the ROBDD algorithms usually perform existential quantification on a variable ℓ by replacing any vertex labeled by ℓ by a ROBDD representing the disjunction of the two formulae represented by the *else* and *then* out-edges of the vertex. Here, since the *then* child is always the sink 0, it is sufficient to replace the vertex by its *else* child[6]. This can be performed during the application of the modified Bellman's algorithm 5, without destroying its linear time complexity.

R'_{E_A} shows that the belief base has two $\ll^{\text{LEX}}$-preferred subbases that respectively reject ϕ_3 and ϕ_4. R''_{E_A} implicitly represents the two models of these two subbases. Using the UNI principle, we can entail p and b but not f: UNI-LEX is

[6]This is done using the ITE operator to avoid a possible duplication of vertices with identical label and children.

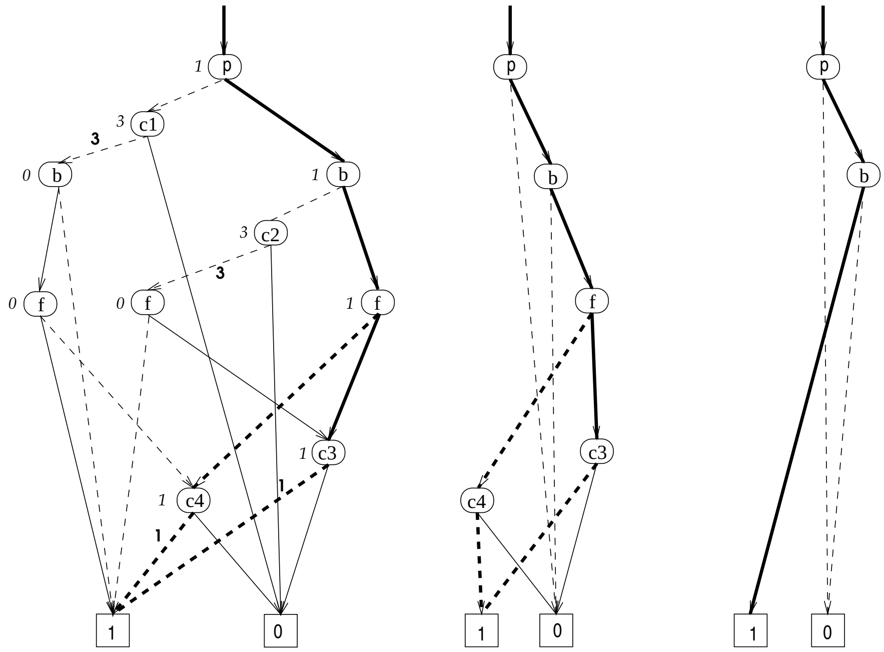

Figure 2: The ROBDD R_{E_A}, R'_{E_A} and R''_{E_A} on the penguin problem

still quite cautious (note that another stratification where ϕ_4 is made less important than ϕ_3 enables the entailment of $\neg f$, since the only $\ll^{\text{Lex}}$-preferred subbase remaining rejects the less specific formula ϕ_4).

5 Experimentations

We have extended the ROBDD package distributed by Bryant and Brace [3] with the ability of weighting the *else* edge of assumption labeled vertices and with the previously described algorithm that simultaneously computes shortest paths, redirects non shortest paths to terminal vertex 0 and quantifies existentially on assumptions. This allows us to build the ROBDD R''_{E_A} which can then be used for checking UNI-LEX entailment of any formula. All the tests presented here have been performed on a SPARCSERVER 1000 using a 50Mhz processor, a machine slower than a standard SPARCSTATION 10.

5.1 Comparing the Variable Orders

We have first applied our algorithm to three simple stratified belief bases: a complete version of the previous (in)famous penguin problem and two formalizations of a small real common-sense reasoning problem that respectively involve 31 and 77 formulae, distributed in respectively 7 and 9 strata (the test problems and the code can be asked by e-mail at lagasq@irit.fr). The aim of the test was mainly to compare the practical efficiency of [21]'s order and our new order. Table 3 successively gives:

- in number of non-terminal vertices, the size of the BDD R_{E_A} using [21]'s order and our order; in both cases, the same order on V was used;
- the CPU time needed to compute R_{E_A} for each order;
- the size, in number of non-terminal vertices, of the BDD R''_{E_A}; since the assumptions do not appear anymore in the BDD, the size is identical for the two orders;
- the CPU time needed to compute R''_{E_A} for each order.

Test	Sizes		CPU (R_E)		Size	CPU (R''_E)	
	[21]'s order	our order	[21]'s order	our order	R''_E	[21]'s order	our order
Peng.	12	10	≈ 0	≈ 0	4	≈ 0	≈ 0
31 f.	252 932	62 028	1'07"	4.7"	17	12.8"	3.1"
77 f.	748 461	83 040	4'25"	17.3"	17	38"	4.2"

Table 3: Comparing the orders

A first conclusion is that the order we propose yields much smaller ROBDDs. Better results could possibly be obtained by optimizing the initial order on V in order to minimize the term $\sum_{\ell \in A}(2^{n_\ell})$ which appears in our lower bound on the BDD size. For the problems considered, the size of the BDD R_{E_A} is very reasonable and yields a final BDD R''_{E_A} which is here very small (only one $\ll^{\text{LEX}}$-preferred subbase for the last 2 tests). Obviously, larger problems could be tackled but the field of nonmonotonic reasoning lacks benchmarks. Therefore, we decided to use random 3-SAT problems as the basis of the next experiments.

5.2 Tests Using Random 3-SAT *Formulae*

The tests have been performed on sets of random 3-clauses (involving three literals), generated using the procedure described in [27]. Two parameters are used: the number n of variables and the number l of 3-clauses. Each 3-clause is built by randomly choosing three variables among the n ones et by randomly changing the sign of each variable with probability 0.5. This model has been intensively studied in the literature and it is known that a so-called phase transition occurs at a ratio of $\frac{l}{n} = 4.25$. Instances generated using a lower (resp. higher) ratio will be consistent (inconsistent) with high probability. Instances generated using the above ratio of 4.25 are also known to define difficult instances for the SAT problem.

The aim of the tests is to see how ROBDDs can cope with problems of various

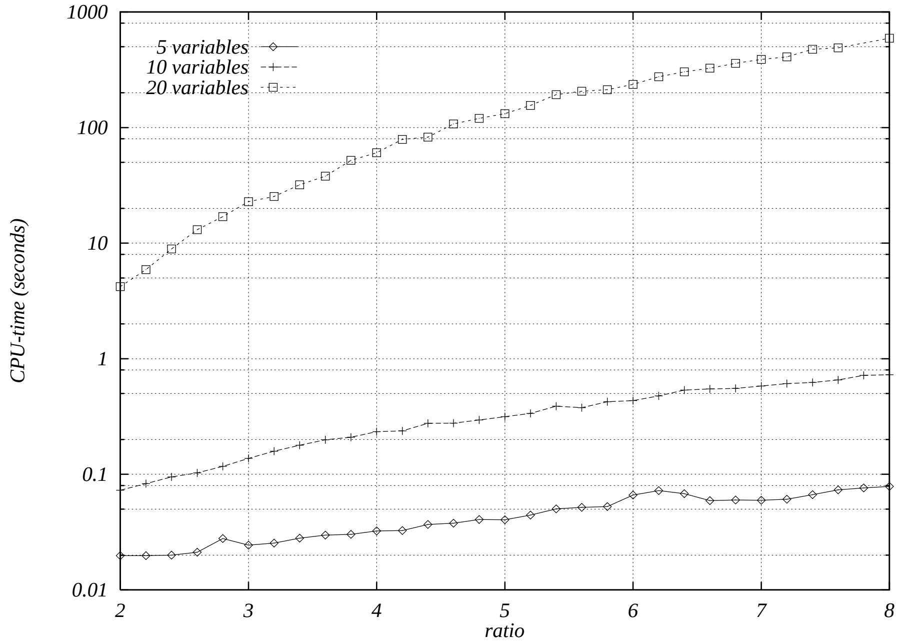

Figure 3: Total cpu-time needed for building R_{E_A}, R''_{E_A} and verifying the entailment

sizes, to compare the sizes of the ROBDDs R_{E_A} and R''_{E_A}, to evaluate the impact of the ratio $\frac{l}{n}$ on the efficiency of the approach and to check that the "knowledge compilation" is efficient *i.e.*, that once R''_{E_A} is built, we get an efficient procedure for checking UNI-LEX entailment. A test consists in generating a random base using a given value of n and l. The base is then pre-ordered by splitting it into 5 strata, simply by randomly assigning each formula to one stratum. Then the ROBDD R_{E_A} is built using our order, starting from an initial random order on V. Then the ROBDD R''_{E_A} is computed. Finally an extra random 3-clause is generated and we check if it is entailed by R''_{E_A}.

These tests have been performed with three different numbers of variables (5, 10 and 20) with a ratio $\frac{l}{n}$ going from 2 to 8 by 0.2 step[7]. For each value of n and l, fifty different bases have been generated and the numbers reported are the average of the results on each of these fifty bases.

The Figure 3 shows the total cpu-time needed to build R_{E_A}, to compute R''_{E_A} and to check for the entailment of one random 3-clause. The horizontal axis indicates the ratio used, the vertical axis gives the cpu-time in seconds using a logarithmic scale. First, we notice that there is apparently no "phase transition"

[7]We remind the reader that all bases generated with a ratio larger than 4.25 are inconsistent with high probability.

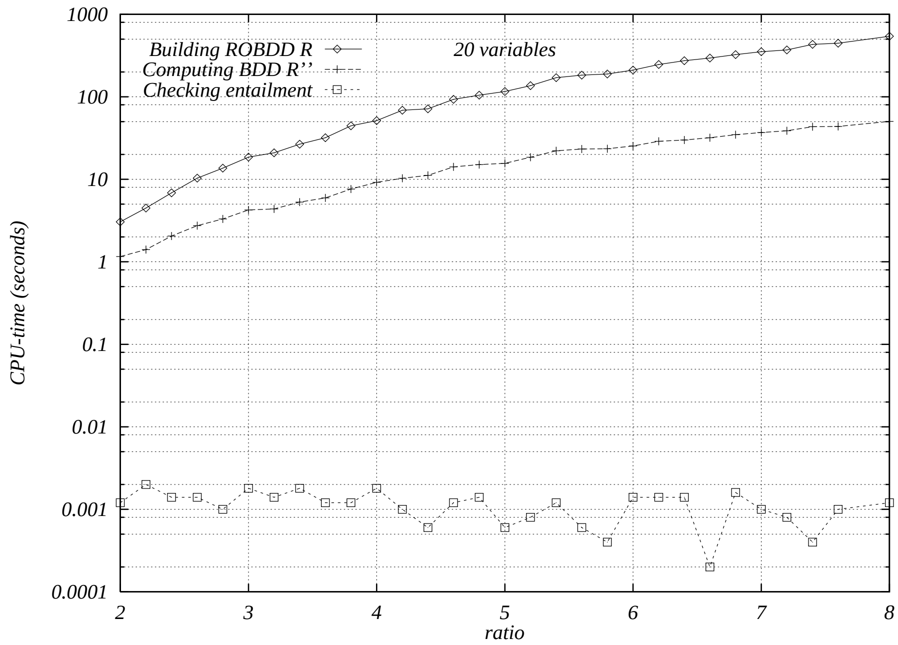

Figure 4: Separated cpu-time needed for building R_{E_A}, R''_{E_A} and verifying the entailment

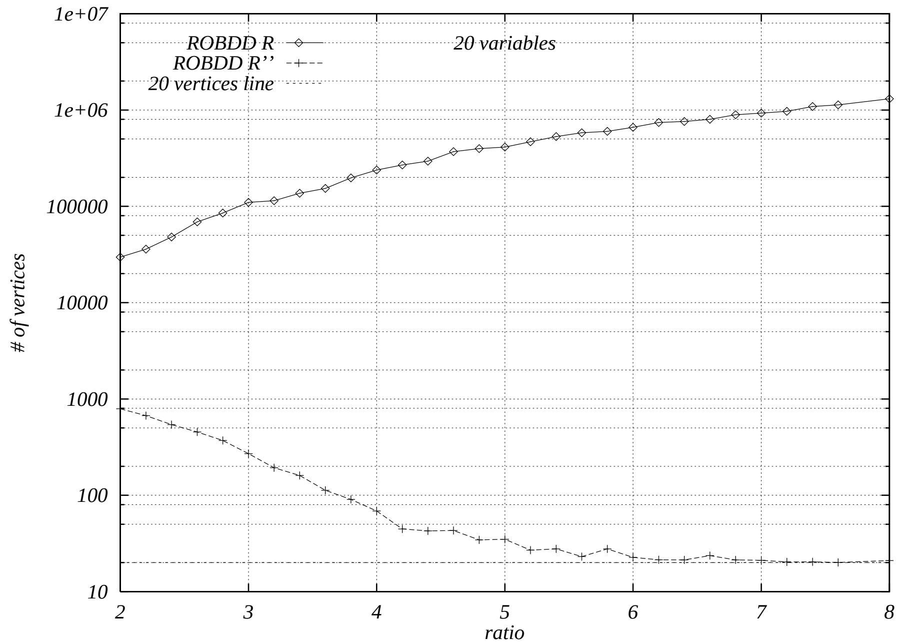

Figure 5: Sizes of the ROBDD R_{E_A} and R''_{E_A}

here: the cpu-time seems to increase steadily as the ratio increases. One thing that is not visible on the figure is the very low variance of the measures: the amount of time needed is highly predictable and stable. Then, one can see that problems with more than 150 clauses are entirely solved in less than $15'$. This amount of time is better decomposed, in the case of bases with 20 variables, in the Figure 4. Here, we can see that almost all the cpu-time is spent building the first ROBDD R_{E_A}. Then some extra time is spent computing R''_{E_A}. But once this ROBDD is built, checking the entailment actually takes a negligible time, around $\frac{1}{1000}th$ of second, even on the largest instances. This shows clearly that the approach can be considered as a "knowledge compilation" approach: some large preprocessing finally yields a very efficient entailment procedure (see [31] for works on the "knowledge compilation" approaches).

Moreover, if the belief base is modified, the "recompilation" is partly incremental, if the ROBDD R_{E_A} has been saved (see [18] for works on this notion of incremental "recompilation"): if a new formula ϕ needs to be inserted in the base, it is sufficient to introduce a new assumption ℓ_ϕ, to compute the conjunction of R_{E_A} and $\ell_\phi \to \phi$ and to apply the procedure Remove-fat once again. To delete a formula ϕ, one can simply compute the conjunction of R_{E_A} with the formula $\neg\ell_\phi$ and then apply the procedure Remove-fat once again.

The efficiency of the final entailment check is the result of the small size of the ROBDD R''_{E_A} compared to the size of R_{E_A}. The sizes, in number of vertices, of the two ROBDDs R_{E_A} and R''_{E_A} are given in the Figure 5. It appears that if the compilation becomes more and more difficult as the ratio increases, this is because the size of the first ROBDD R_{E_A} increases too, but the size of the ROBDD R''_{E_A} reduces as the ratio increases, making entailment more and more efficient. This can be explained by the fact that the $\ll^{\text{Lex}}$ order is extremely selective: for most bases with a ratio above 6, not only does the $\ll^{\text{Lex}}$ preference select one single preferred subbase, but this subbase has usually only one model. This explains the size of the ROBDD R''_{E_A} on highly inconsistent bases: the ROBDD contains only one model and uses therefore 20 vertices (the number of variables).

One could think, from these results, that the $\ll^{\text{Lex}}$ preference is actually too selective to have any practical significance for highly inconsistent bases. But it is difficult to conclude from bases entirely composed of 3-sat formulae.

6 Conclusion

We have studied the computational complexity of various coherence-based entailment relations which can be defined as: $(E, \leq) \mid\!\sim^{p,m} \Phi$. E denotes a set of beliefs, $\leq$ a priority relation on E, Φ a propositional formula, and p, m enable to combine the classical entailment and the selection of preferred consistent subsets. The results reported in this paper show that most of the nonmonotonic entailment problems have very likely exponential time complexity with respect to the problem size. Although the complexities observed are limited by the third level of the PH, they are prohibitive and applications may likely wait for an answer for hours, days or centuries!

We have considered three restrictions (strictly ordered belief bases, Horn bases, strictly ordered Horn bases), but only the last one leads to a polynomial problem, and it is a very restrictive case.

A more complete analysis permits to distinguish the UNI-LEX entailment, whose complexity never exceeds the Δ_2^p class. Note that the computational complexity is not related to cautiousness: though ARG-m is more cautious than EXI-m and less cautious than UNI-m, ARG-m is more complex than EXI-m and UNI-m (see [8] for a study on this point of view).

Considering the strength of the restrictions needed to reach polynomial complexity, we decided to try to tackle one specific entailment relation using an algorithmic tool which is dedicated to the resolution of propositional logic based NP-hard problems: Binary Decision Diagrams.

On the specific UNI-LEX relation considered, our BDD-based approach offers some interesting features:

- efficiency via knowledge compilation: after a first expensive computation, the binary decision diagram R''_{E_A} can be used to efficiently check the entailment of any formula;
- a "good" variable order for BDD, with both better theoretical guarantees and better practical results than [21]'s order;
- the recompilation can be incremental as long as the first BDD R_{E_A} is kept; even if this BDD may be huge, it may simply be saved to disk.

We feel that this work can still be extended in several directions:

- For a given order on the set V, is our order on $V \cup A$ optimal? Since our theoretical bound depends on the initial order on the variables in V, is it worth considering the optimization of this order in order to minimize the bound? Is this optimization problem computationally tractable? Finally, does this optimization lead to better practical results?
- Obviously, the BDD approach can be extended to other preference relations than the lexicographic order. Naturally, this is immediate for cardinality based preferences, a special case of the lexicographic order, but one could also consider the *Best-Out* order, related to possibilistic logic (see definitions in [12], and complexity results in [7]), and for which specific optimization should apply.

- Finally, one should try to use the BDD R'_{E_A}, which represents all preferred consistent subbases and their models, in order to tackle consequence relations based on other entailment principles than the UNI principle. This is especially interesting because of the higher complexity of the problems defined by the EXI or ARG principles.

References

[1] Salem Benferhat, Claudette Cayrol, Didier Dubois, Jérôme Lang, and Henri Prade. Inconsistency management and prioritized syntax-based entailment. In *Proc. of the 13^{th} IJCAI*, pages 640–645, Chambéry, France, 1993.

[2] Salem Benferhat, Didier Dubois, and Henri Prade. Argumentative inference in uncertain and inconsistent knowledge bases. In *Proc. of the 9^{th} UAI*, pages 411–419, Washington, DC, 1993.

[3] Karl S. Brace, Richard R. Rudell, and Randal E. Bryant. Efficient implementation of a BDD package. In *27^{th} ACM/IEEE Design Automation Conference*, pages 40–45, 1990.

[4] Gerhard Brewka. Preferred subtheories : An extended logical framework for default reasoning. In *Proc. of the 11^{th} IJCAI*, pages 1043–1048, Detroit, MI, 1989.

[5] Randal E. Bryant. Graph-based algorithms for boolean function manipulation. *IEEE Transactions on Computers*, C-35(8):677–691, 1986.

[6] Randal E. Bryant. Symbolic boolean manipulation with ordered binary-decision diagrams. *ACM Computing Surveys*, 24(3):293–318, September 1992.

[7] Claudette Cayrol and Marie-Christine Lagasquie-Schiex. Comparaison de relations d'inférence non-monotone : étude de complexité. Rapport de recherche 93-23R, Institut de Recherche en Informatique de Toulouse (I.R.I.T.), France, September 1993.

[8] Claudette Cayrol and Marie-Christine Lagasquie-Schiex. Non-monotonic syntax-based entailment: A classification of consequence relations. In *Lecture Notes in Artificial Intelligence 946 (Proc. of ECSQARU-95)*, pages 107–114, Fribourg, Switzerland, 1995.

[9] Claudette Cayrol, Véronique Royer, and Claire Saurel. Management of preferences in assumption-based reasoning. In *Advanced methods in AI. Lecture notes in computer science 682*, pages 13–22. 1992.

[10] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to algorithms.* MIT Press, 1990.

[11] Rina Dechter and Avi Dechter. Belief maintenance in dynamic constraint networks. In *Proc. of AAAI-88*, pages 37–42, St. Paul, MN, 1988.

[12] Didier Dubois, Jérôme Lang, and Henri Prade. Inconsistency in possibilistic knowledge bases — to live or not to live with it. In *Fuzzy logic for the Management of Uncertainty*, pages 335–351. 1991.

[13] Florence Dupin de Saint Cyr, Jérôme Lang, and Thomas Schiex. Gestion de l'inconsistance dans les bases de connaissances : une approche syntaxique basée sur la logique des pénalités. In *Actes de RFIA'94*, pages 507–518, France, 1994.

[14] Thomas Eiter and Georg Gottlob. On the complexity of propositional knowledge base revision, updates, and counterfactuals. *Artificial Intelligence*, 57:227–270, 1992.

[15] Thomas Eiter and Georg Gottlob. The complexity of logic-based abduction. In *Proc. of the 10^{th} Symposium on Theoretical Aspects of Computing STACS*, pages 70–79, Würzburg, Germany, 1993.

[16] Peter Gärdenfors and David Makinson. Nonmonotonic inference based on expectations. *Artificial Intelligence*, 65:197–245, 1994.

[17] Michael R. Garey and David S. Johnson. *Computers and Intractability : A Guide to the Theory of NP-completeness.* W.H. Freeman and Company, New York, 1979.

[18] Goran Gogic, Christos H. Papadimitriou, and Martha Sideri. Incremental recompilation of knowledge. In *Proc. of AAAI-94*, pages 922–927, Seattle, WA, 1994.

[19] Georg Gottlob. Complexity results for nonmonotonic logics. *Journal of Logic and Computation*, 2(3):397–425, 1992.

[20] Philippe Jégou. A logical approach to solve dynamic CSPs: Preliminary report. In *Proceedings ECAI'94 Workshop on Constraint Satisfaction Issues raised by Practical Applications*, pages 87–94, Amsterdam, The Netherlands, August 1994.

[21] Philippe Jégou and Fabrice Bouquet. Solving over-constrained CSP using weighted OBDD. In *Proceedings of the CP'95 Workshop on Over-Constrained Systems*, Cassis, France, September 1995.

[22] David S. Johnson. A catalog of complexity classes. In Jan van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume A : Algorithms and Complexity, chapter 2, pages 67–161. Elsevier, 1990.

[23] Johan De Kleer. An assumption-based TMS. *Artificial Intelligence*, 28:127–162, 1986.

[24] Marie-Christine Lagasquie-Schiex. *Contribution à l'étude des relations d'inférence non-monotone combinant inférence classique et préférences.* Thèse, Université Paul Sabatier, IRIT, 1995.

[25] C. Y. Lee. Representation of switching circuits by binary-decision programs. *Bell System Tech. J.*, 38(4):985–999, July 1959.

[26] Daniel Lehmann. Another perspective on default reasoning. Rapport de recherche 92-12, Leibniz Center for Research in Computer Science. Hebrew University of Jerusalem, Israel, July 1992.

[27] Hector Levesque, David Mitchell, and Bart Selman. Hard and easy distributions of SAT problems. In *Proc. of AAAI-92*, pages 459–465, San Jose, CA, 1992.

[28] Bernhard Nebel. Belief revision and default reasoning: Syntax-based approaches. In *Proc. of the 2^{nd} KR*, pages 417–428, Cambridge, MA, 1991.

[29] Gadi Pinkas and Ronald P. Loui. Reasoning from inconsistency : A taxonomy of principles for resolving conflict. In *Proc. of the 3^{rd} KR*, pages 709–719, Cambridge, MA, 1992.

[30] Raymond Reiter. A logic for default reasoning. *Artificial Intelligence*, 13(1-2):81–132, 1980.

[31] Bart Selman and Henry Kautz. Knowledge compilation using horn approximations. In *Proc. of AAAI-91*, pages 904–909, Anaheim, CA, 1991.

A Proofs

Proof of Theorem 2

We first suppose that B is $\ll^{\text{LEX}}$-preferred to C. Let $\bar{B}$ be the complement of B in E and $B_j = B \cap E_j$. The weight of B is defined as $\sum_{j=1}^{n} |\bar{B}_j| \times w(j)$. Since B is $\ll^{\text{LEX}}$-preferred to C, by definition there exists a stratum i such that $|B_i| > |C_i|$ and therefore $|\bar{B}_i| < |\bar{C}_i|$. Furthermore, for all $j < i$, $|B_i| = |C_i|$ and therefore $|\bar{B}_i| = |\bar{C}_i|$. Finally, we know that $\forall k > i, |\bar{B}_k| - |\bar{C}_k| \geq -|E_k|$. The difference $w(C) - w(B)$ between the weight of C and the weight of B is then equal to:

$$w(C) - w(B) \quad = \quad \sum_{j=1}^{n} ([|\bar{C}_j| - |\bar{B}_j|] \times w(j))$$

$$= \sum_{j=i}^{n}([|\bar{C}_j| - |\bar{B}_j|] \times w(j)) \quad \geq \quad w(i) - \sum_{j=i+1}^{n}(|E_j| \times w(j))$$

If we introduce the definition of $w(j) = \prod_{k=n-1}^{j}[1 + |E_{k+1}|]$ for $j > 1$, we get:

$$\begin{aligned} w(C) - w(B) \quad \geq \quad & \prod_{k=n-1}^{i}[1 + |E_{k+1}|] - \\ & |E_{i+1}| \times \prod_{k=n-1}^{i+1}[1 + |E_{k+1}|] - \\ & \vdots \\ & |E_{n-1}| \times \prod_{k=n-1}^{n-1}[1 + |E_{k+1}|] - |E_n| \times 1 \quad \geq \quad 1 \end{aligned}$$

And therefore, the weight C is strictly larger than the weight of B.

The converse follows from the fact that $\ll^{\text{LEX}}$-preference and weight order are total: if B is not $\ll^{\text{LEX}}$-preferred to C, C is $\ll^{\text{LEX}}$-preferred to B and therefore the weight of C is lower then the weight of B or equivalently, the weight of B is not lower than the weight of C. □

Proof of Theorem 3

This proof relies on two lemmas:

Lemma 1
Each model ω_V of a minimum weight consistent subbase B defines a model of E_A which is represented in the ROBDD R_{E_A} by a path P from the source to the sink 1 whose length is lower than the minimum weight of a consistent subbase.

Proof: Consider any model ω_V of a minimum weight consistent subbase B. Let ω be the model of E_A defined by ω_V and this base and P the path representing ω in the ROBDD. The partial truth assignment ω_P defined by the path P being included in ω, the set of assumption variables which are assigned to 0 in the path P is included in the set of assumptions which are assigned to 0 in the truth assignment ω. Therefore, the length of the path is necessarily lower than the weight of the base B. □

Lemma 2
Any model ω_V defined by a model ω represented by a shortest path P from the source to the sink 1 is a model of a consistent subbase of weight equal to the length of the path.

Proof: Consider a shortest path P from the source to sink 1 representing a partial assignment ω_P. All the models ω that contain ω_P are models of E_A which define a consistent subbase $Base(\omega_A)$ and one of its model ω_V. Consider any such ω_V, and an assignment ω'_A of A such that $\omega'_A(\ell_{\phi_i}) = 0$ iff $\omega_P(\ell_{\phi_i}) = 0$. Then, by construction, the truth assignment ω' of $V \cup A$ defined by ω_V and ω'_A is a model of E_A since it is represented by the path P (since it contains ω_P) and therefore ω_V is the model of a subbase $Base(\omega'_A)$ whose weight is equal to the length of the shortest path. □

Proof of Theorem 3

$(E, \leq) \mid\sim^{\forall,\text{LEX}} \Phi$ means that all $\ll^{\text{LEX}}$-preferred subbases of E classically entail Φ, or equivalently that each model of each $\ll^{\text{LEX}}$-preferred consistent subbase satisfies Φ. By Theorem 2, $\ll^{\text{LEX}}$-preferred consistent subbases are also minimum weight consistent subbases and therefore it suffices to show that the set of all the models ω_V represented in shortest paths of the ROBDD is equal to the set of all the models of minimum weight consistent subbases. According to Lemma 2, the length of a shortest path (and therefore of any path) cannot be strictly lower than the weight of a minimum weight base. Since, according to Lemma 1, the length of a path representing a model of a minimum weight subbase is lower than this weight, we can conclude that the length of a shortest path is equal to the weight of a minimum weight base. Then, Lemma 1 simply says that all the models of a minimum weight consistent subbase are represented in a shortest path, while Lemma 2 says that all the models of E represented in shortest path are models of a minimum weight base. This concludes the proof. □

Proof of Theorem 4

We state a more precise form of Theorem 4:

Theorem 5
Let $E = \{\phi_i\}_i$ be a belief base, and V the set of variables of E ($|V| = n$). We associate to each formula ϕ_i of E a new variable ℓ_{ϕ_i}. Let A be the set of these new variables, called assumption variables ($|A| = |E| = m$). We define $E_A = \{\ell_{\phi_i} \rightarrow \phi_i, i = 1 \dots m\}$. Given an order $\prec$ on V, we consider an order $\prec'$ on $V \cup A$ such that:

- *it extends the order $\prec$ ($\forall v, v' \in V, (v \prec' v') \Leftrightarrow (v \prec v')$);*
- *an assumption ℓ_ϕ is ordered after all the variable of ϕ ($\forall v \in V$ that appears in ϕ, $v \prec' \ell_\phi$);*
- *and before any variable located after all the variables of ϕ ($\forall w \in V$ s.t. $\forall v \in V$ that appears in ϕ, $v \prec w$, then $\ell_\phi \prec' w$).*

For each assumption variable ℓ, n_ℓ denotes the number of variables $v \in V$ s.t. $v \prec' \ell$. In the corresponding reduced ordered BDD of the base E_A, the number of non terminal vertices is less than $(2^n - 1) + \sum_{\ell \in A}(2^{n_\ell})$.

Proof: Let us first note that an order such as $\prec'$ always exists and is simply obtained by inserting each assumption just after all the variables of the associated formula. The proof uses the fact that a reduced ordered BDD is *defined* as the closure by two reduction rules of the ordered binary decision tree (cf. Figure 1). Each rule application decreases the number of vertices of the BDD by one and the reduced BDD is obtained by application of these two rules until quiescence.

We will show that a limited application of these rules on the ordered binary decision tree of E_A permits to obtain an ordered and partially reduced BDD whose size is less than $(2^n - 1) + \sum_{\ell \in A}(2^{n_\ell})$. The size of the completely reduced BDD being necessarily less than the size of the partially reduced BDD, the proof will follow.

We consider the ordered binary decision tree of the base E_A. Let $\ell_{\phi_i} \in A$, the assumption variable associated to the formula ϕ_i. Let $s_{\ell_{\phi_i}}$ one of the vertices corresponding to this assumption variable in the decision tree. By definition, ℓ_{ϕ_i} is greater than all the variables of ϕ_i according to $\prec'$. So, we have two cases:

- The assignment of the variables of ϕ_i satisfies ϕ_i: in this case, the formula $\ell_{\phi_i} \rightarrow \phi_i$ is satisfied for all the values of ℓ_{ϕ_i}. As ℓ_{ϕ_i} does not appear in other formulae of E_A, the two subtrees (left and right) of $s_{\ell_{\phi_i}}$ are the same and we may remove the vertex $s_{\ell_{\phi_i}}$ (and one of the subtrees) by application of the reduction rules.

- The assignment of the variables of ϕ_i does not satisfy ϕ_i: in this case, if ℓ_{ϕ_i} is supposed to be true, the formula $\ell_{\phi_i} \rightarrow \phi_i$ is not satisfied, whatever the values of the variables which "follow" ℓ_{ϕ_i} in the order. So, all the sink vertices of the subtree of $s_{\ell_{\phi_i}}$, corresponding to the assignment at true of ℓ_{ϕ_i}, are the sink vertex 0. Therefore, by several applications of the two reduction rules, we may replace this subtree by the sink vertex 0.

The proof follows by induction. If $m = 0$, then the number of non terminal vertices is obviously less than $(2^n - 1)$. The induction hypothesis will be that Theorem 5 applies on a BDD representing a set of formulae $\{\ell_\phi \rightarrow \phi\}$, with an order corresponding to the definition, and in which there are strictly less than m assumption variables.

Consider the binary decision tree associated to the base E_A, with $2^{n+m} - 1$ non terminal vertices. Let α be the lowest assumption variable according to $\prec'$. Using the previous remarks, the application of the two reduction rules implies on each vertex s_α either the removal of the vertex, or the fact that one of its children becomes the sink vertex 0. Therefore, the number of non terminal vertices in the partially reduced BDD obtained by this procedure is decomposed into:

- $(2^{n_\alpha} - 1)$ vertices for the n_α variables which are not assumption variables and which are lower then α according to $\prec'$;
- a maximum of 2^{n_α} vertices corresponding to the assumption variable α;
- 2^{n_α} subtrees whose roots correspond to the variable which is the successor of α in $\prec'$.

These 2^{n_α} subtrees are ordered binary decision trees. Let ω the partial assignment defined by the path between the root of the tree and the root of one of these subtrees. Whatever subtree among the 2^{n_α} subtrees whose root corresponds to a successor of α in $\prec'$, it represents the binary decision tree of the formula E_{A_ω} obtained by the substitution of the variables assigned by ω in E_A by their truth values in ω. Because of the order $\prec'$, E_{A_ω} contains formulae 0 or 1 and formulae of the form $\ell_\phi \rightarrow \phi$. The formulae 1 don't affect the truth value of E_{A_ω}, because they are always satisfied. So:

- either E_{A_ω} contains a formula 0 and its truth value is always equal to 0: the binary decision subtree of E_{A_ω} is reduced to the sink vertex 0 by the use of reduction rules;
- or E_{A_ω} doesn't contain any formula 0 and furthermore satisfies all the conditions of the induction hypothesis: there are $m-1$ assumption variables and the condition on the order is satisfied (an assumption variable is located "after" the variables of the associated formula in the order). Therefore, we may reduce the subtree to an ordered and partially reduced BDD whose number of non terminal vertices is less than $(2^{n-n_\alpha} - 1) + \sum_{\ell \in A-\{\alpha\}} (2^{n_\ell - n_\alpha})$.

In all the cases, we may reduce each subtree to an ordered partially reduced BDD whose number of non terminal vertices is less than $(2^{n-n_\alpha} - 1) + \sum_{\ell \in A-\{\alpha\}} (2^{n_\ell - n_\alpha})$.

The set of reductions permits to obtain an ordered and partially reduced BDD whose number of non terminal vertices is less than:

$$\begin{aligned} S &= \overbrace{[2^{n_\alpha} - 1]}^{\text{variables } \prec' \alpha} + \overbrace{[2^{n_\alpha}]}^{\alpha} + \overbrace{[2^{n_\alpha}.((2^{n-n_\alpha} - 1) + \sum_{\ell \in A-\{\alpha\}} (2^{n_\ell - n_\alpha}))]}^{\text{subtrees rooted } \succ' \alpha} \\ &= [2^{n_\alpha} - 1] + [2^{n_\alpha}] + [2^n - 2^{n_\alpha} + \sum_{\ell \in A-\{\alpha\}} (2^{n_\ell})] \\ &= 2^{n_\alpha} - 1 + 2^n + \sum_{\ell \in A-\{\alpha\}} (2^{n_\ell}) \;=\; (2^n - 1) + \sum_{\ell \in A} (2^{n_\ell}) \end{aligned}$$

□