B. BESSANT É. GRÉGOIRE P. MARQUIS L. SAÏS

ITERATED SYNTAX-BASED REVISION IN A NONMONOTONIC SETTING

B. BESSANT É. GRÉGOIRE P. MARQUIS L. SAÏS

ITERATED SYNTAX-BASED REVISION IN A NONMONOTONIC SETTING

ABSTRACT: In this paper, a new syntax-based approach to iterated belief base revision is presented. It is developed within a nonmonotonic framework that allows a two-steps handling of inconsistency to be adopted. First, a disciplined use of nonmonotonic ingredients is made available to the knowledge engineer to prevent many inconsistencies that would occur if a standard logical interpretation and representation of beliefs were conducted. Remaining inconsistencies are considered unexpected and revised by weakening the formulas occurring in any minimally inconsistent subbase, as if they were representing exceptional cases that do not actually occur. While the computation of revised knowledge bases remains intractable in the worst case, our approach benefits from an efficient local search-based heuristic technique that empirically proves often viable, even in the context of very large propositional applications.

1 INTRODUCTION

Various approaches to belief revision have been proposed this last decade. Most of them have been investigated from a knowledge-level theoretical perspective [32], mainly. Despite a few noticeable exceptions (e.g. [34] [29] [11] [20]), their computational counterparts have rarely been considered. Worse, due to worst case complexity results, they are often believed to be intractable when large-scale applications are addressed. However, recent impressive empirical progress in propositional reasoning and search [36] does open real computational perspectives for syntax-based approaches to belief revision with respect to large applications. Also, many efforts have been devoted to establish formal connections between nonmonotonic logics and belief revision (see e.g. [9] [21] [25] [19]). In this paper, a knowlege[1] engineering-motivated approach to belief revision is adopted, showing that belief revision and nonmonotonic logics are not only two sides of the same coin [14] but that they can play useful synergetic roles in a unified framework that proves empirically computationally viable. More precisely, a two-steps policy to syntax-based revision of inconsistent beliefs is proposed. On the one hand, nonmonotonic ingredients are provided within a belief representation language giving rise to a gain of expressiveness, allowing a careful knowledge engineer to avoid many future inconsistencies that would occur if a standard deductive representation and inference of beliefs were selected. Inconsistencies that are not forecast and prevented in this way are then the object of a syntax-based belief revision process. In this respect, we can envision a whole family of syntax-based revision approaches: from a *full-meet* cautious one that weakens each formula in each minimally inconsistent subbase to regain consistency, to a *maxichoice* change policy that

[1] In this paper, *knowledge* is used for *belief*.

just weakens one smallest subset of formulas to restore consistency. In this paper, an empirically viable computational technique for the first approach is proposed. It makes use of a specific heuristic in the trace of local search techniques to check (in)consistency. Interestingly enough, it is shown that the nonmonotonic ingredients do not significantly affect the computational complexity results with respect to the revision process, neither from a worst case point of view, nor from an empirical one.

The paper is organized as follows. After some formal preliminaries, a specific nonmonotonic representation setting is described. Then, a new syntax-based approach to iterated belief base revision is introduced and motivated in this framework, together with a preferred-models inference relation for the resulting logic. The complexity and rationality issues for this cautious, full-meet approach to base revision are analyzed. Local search techniques for SAT are then specialized to match our specific needs. Typical results from extensive tests are reported. Finally, the specific position adopted in this paper with respect to the various facets of belief revision is summarized, and perspectives for further research are briefly exposed. Basic elements of computational complexity can be found in an appendix.

2 SOME FORMAL PRELIMINARIES

Let PS denote a denumerable set of *propositional symbols* (also called *variables* or *atoms*). For every subset V of PS, $PROP_V$ denotes the *propositional language* built up from the symbols of PS, the boolean constants $true$ and $false$ and the connectives $\neg$, $\wedge$, $\vee$, $\Rightarrow$, $\Leftrightarrow$ in the standard way. The elements of $PROP_{PS}$ are called *formulas*.

Every propositional symbol of PS is also called a *positive literal* and a negated one a *negative literal*. The boolean constants are also viewed as positive literals. A *literal* is either a positive literal or a negative literal. Every finite disjunction of literals is called a *clause*. $Var(KB)$ denote the set of propositional variables occurring in the formula KB. For every propositional symbol v from PS, $KB_{v\leftarrow true}$ (resp. $KB_{v\leftarrow false}$) denotes the formula KB in which every occurrence of v is replaced by $true$ (resp. $false$).

An *interpretation* I (over $PROP_{PS}$) is a mapping that associates every propositional symbol to one of the two truth values of $BOOL = \{0, 1\}$ and $I(true) = 1$ and $I(false) = 0$. $[\![KB]\!](I)$ denotes the truth value taken by KB within I; it is defined in the usual compositional way. When I is s.t. $[\![KB]\!](I) = 1$, I is said to be a *model* of KB, noted $I \models KB$; otherwise, I is a *counter-model* of KB. The set of all models of KB is noted $\mathcal{M}(KB)$. When a formula has a model, it is said *consistent*. When a formula has no model (resp. no counter-model), it is said *inconsistent* (resp. *valid*). The binary relation $\models$ over $PROP_{PS}$ is defined by $KB \models f$ iff $\mathcal{M}(KB) \subseteq \mathcal{M}(f)$ holds. The *deductive closure* of KB is the set of all formulas that are logical consequences of KB. Whenever $KB \models f$ and $f \models KB$ both hold, KB and f are said *logically equivalent*, noted $KB \equiv f$.

Every set of formulas is interpreted conjunctively; especially, when it is finite, a set of formulas is also considered as the formula that is the conjunction of its elements. It is well-known that every formula admits a conjunction of clauses equivalent to it; such a conjunction of clauses is called a *conjunctive normal form* (CNF for short) of the formula.

3 A NONMONOTONIC REPRESENTATION FRAMEWORK

This paper is more concerned with the knowledge engineering branch of artificial intelligence than with its cognitive and philosophical sides; building effective practical systems is here the main goal. In this respect, it is generally agreed that some trade-off can be required between the expressiveness of the representation language and the inference efficiency. Similarly, a specific combination of belief revision and nonmonotonic representation of beliefs and inference can be beneficial, especially when software quality properties are required.

Accordingly, a two-steps handling of inconsistent beliefs is advocated in this paper. First, the knowledge engineer is provided with a nonmonotonic language allowing him to prevent many future inconsistencies that would occur if a purely standard deductive logic were used. This language should be expressive enough to represent (prioritized) rules of default reasoning and exceptions, preventing these last ones from leading to inconsistency when they actually occur. In this respect, it is claimed here that when real-world large-scale applications are undertaken, the knowledge engineer can be required a disciplined use of nonmonotonic ingredients that are restricted, in exchange for the hope for predictable system behavior and good computational performance in many cases. However, the knowledge engineer cannot be required to forecast all such exceptions in this way. This is exactly where belief revision has to play a role in our framework. Belief revision will thus here just concern a mere remainder of the inconsistencies that would occur if a purely deductive logic were selected. This point will allow specific revision policies to be justified.

In the following, any belief base KB will be represented as a finite set (interpreted conjunctively) of propositional formulas from $PROP_{PS}$. The set of propositional symbols PS is partitioned into three denumerable sets, P, AB_{KB} and $AB_{Revision}$. P is used to represent the core knowledge of KB, while the two disjoint sets AB_{KB} and $AB_{Revision}$ contain McCarthy's prioritized Abnormality propositions, noted respectively by Ab_i and Ab_{R_i} [24]. Elements from AB_{KB} allow prioritized rules of default reasoning to be expressed together with exceptions (they encode structural abnormalities) while $AB_{Revision}$ is actually a sequence and is devoted to the revision policy.

Typically, a default rule expressing that a bird can fly (unless exceptions) will be of the form $\neg Bird \vee Ab_{flying\_bird} \vee Fly$, representing the formula $Bird \wedge \neg Ab_{flying\_bird} \Rightarrow Fly$. Ab_i propositions are intended to allow for exceptions without the need of expressing them explicitly. Such a framework will be interpreted under model-preference semantics *à la* Shoham [39]: $KB \mid\!\sim_{\leq} f$ iff $[\![f]\!](I) = 1$ in all preferred models I of KB, where the preferred models of KB are those that are minimal w.r.t. a given pre-order $\leq$. Intuitively, models where Ab_i propositions are minimized (i.e. are interpreted as 0) will be preferred. Many minimization schemata (especially prioritizing ones) giving rise to preferred-models inference relations $\mid\!\sim_{\leq}$ can be adopted in this context. The approaches of revision that will be developed in this paper are compatible with all of them.

4 BASE REVISION BY WEAKENING: THE FULL-MEET APPROACH

As the last example illustrates it, the knowledge engineer may not forecast possible exceptions to all rules (assume that he has asserted that all birds fly and that now we are given a penguin). When inconsistency does occur, several possible approaches to restore consistency can be devised. As we adopt here a knowledge engineering approach to artificial intelligence, we claim that the syntax of the encoded beliefs is important and that this should appear in the revision process. Accordingly, in our approach, the epistemic state of an agent (i.e. the set of all its beliefs) is represented as a belief base, i.e. a finite set of propositional formulas: the pieces of knowledge that can be inferred from such a set are the agent's beliefs. While such an approach to belief revision is syntax-sensitive (at least to the extent that it cannot be guaranteed that two equivalent KBs will always be revised in the same way), it is representationally feasible; this contrasts with the belief set approach and the model-based approach to belief revision: due to space considerations, representing the epistemic state of an agent by a deductively closed set of formulas, or by the corresponding set of models, is typically out of reach.

Our cautious approach to belief revision is based on the logical concept of weakening. Indeed, we believe that throwing out some beliefs in order to restore consistency is unnecessary destructive. Accordingly, we prefer "hidding" these beliefs by logically weakening them, allowing interesting perspectives with respect to iterated belief revision.

Intuitively, the way according to which beliefs are weakened amounts to considering these beliefs as exceptional ones that do not hold in regard of the newly introduced formula f.

Assume KB to be a nonmonotonic propositional belief base to be interpreted under some preferred-models semantics, given by a pre-order $\leq$

among interpretations. Initially, KB does not contain any occurrence of a symbol from $AB_{Revision}$. Let f be the so-called *revision formula*, i.e. a new piece of knowledge to be added into KB. A *severe revision* has to occur whenever $KB \cup \{f\}$ is inconsistent. The sequence of Abnormality propositions $AB_{Revision} = \{Ab_{R_1}, \ldots, Ab_{R_j}, \ldots\}_{(j<\omega)}$ is used to remove inconsistencies. More precisely, each time a severe revision is to be conducted, a subset of formulas in KB are weakened using the next available Ab_{R_j} in the sequence $AB_{Revision}$ as an additional disjunct in each of these clauses. Note that the elements of $AB_{Revision}$ are ordered in such a way that AB_{R_j} is after AB_{R_i} whenever $i < j$, so as to ensure that the most recently introduced abnormality symbol Ab_{R_j} occurs after Ab_{R_i} in $AB_{Revision}$.

Now, several revision policies can be defined depending on the way this set of formulas to be weakened is selected. Indeed, in order to break all inconsistencies, it is sufficient to weaken one formula in every of the possibly many subbases of KB that are inconsistent with f. Accordingly, several hitting sets of such subbases can be elected for being weakened in the general case. For the case where no additional information (e.g. an epistemic relevance ordering [28]) enables chosing one such hitting set, a *cautious* policy is proposed: in this situation, all the formulas in the set-theoretic union of the smallest subsets of KB inconsistent with f are weakened. But let us formalize this.

DEFINITION 1 (minimally inconsistent subbase). A *minimally inconsistent* subbase of KB w.r.t. f is any subset S of KB s.t. $S \cup \{f\}$ is inconsistent and no proper subset of it satisfies the same property.

Such subbases are sometimes called *entailment sets* for $\neg f$ (indeed, $S \cup \{f\}$ is inconsistent iff $\neg f$ is entailed by S).

DEFINITION 2 (kernel). The *kernel* $Ker(KB, f)$ of KB w.r.t. f is the set-theoretic union of all minimally inconsistent subbases of KB w.r.t. f

$Ker(KB, f)$ can also be defined in a dual way, focusing on the maximally consistent subbases of KB w.r.t. f (also called *remainders* of KB by $\neg f$). Indeed, $Ker(KB, f)$ is the set-theoretic complement in KB of the intersection of all such subbases (see Theorem 7.1 from [31]).

We are now ready to define our full-meet base revision operation:

DEFINITION 3 (full-meet revision). Assume that KB is the result of i previous severe revisions. $KB \circ_{full-meet} f = (Ker(KB, f) \vee Ab_{R_{i+1}}) \cup (KB \setminus Ker(KB, f)) \cup \{f\}$

In this definition, $Ker(KB, f) \vee Ab_{R_{i+1}}$ denotes the set of formulas $g \vee Ab_{R_{i+1}}$, where g is a formula from $Ker(KB, f)$. Clearly enough, the rank $i + 1$ of the current revision can be determined in time linear in the size of KB (it is sufficient to filter out the greatest i s.t. AB_{R_i} occurs in KB, and to increment it).

EXAMPLE 4. Let us illustrate the previous definition through a very simple example. Let $KB = \{\neg Bird \vee Fly, Bird\}$. This corresponds to the case where the knowledge engineer does not envision any exceptional bird, i.e. a bird that does not fly. We have:

1. $KB \circ_{full-meet} \neg Fly = \{\neg Bird \vee Fly \vee Ab_{R_1}, Bird \vee Ab_{R_1}, \neg Fly\}$.
2. $(KB \circ_{full-meet} \neg Fly) \circ_{full-meet} Fly = \{\neg Bird \vee Fly \vee Ab_{R_1}, Bird \vee Ab_{R_1}, \neg Fly \vee Ab_{R_2}, Fly\}$.

In our approach, every formula from KB that belongs to the intersection of all maximally consistent subbases of KB w.r.t. f is kept in the revised base, while the remaining formulas are weakened. Thus, instead of focusing on the intersection of the *logical closures* of all (preferred[2]) maximally consistent subbases of KB w.r.t. f (as it is the case in many base revision approaches), we focus on the subbases themselves. This has two immediate, favourable consequences:

- Revised KBs can be efficiently represented in our approach (since the size of the revised KBs cannot exceed the size of the corresponding original KB plus the size of the revision formula plus the number of formulas occurring in the original KB).
- Revised KBs do not reduce to a single formula. Accordingly, iterated belief revision can be achieved without trivializing as soon as the first severe revision has been achieved.

These two valuable features are not shared by many syntax-based approaches to base revision. On the one hand, it is shown [5] that representations of revised KBs that are compact, i.e. of size polynomially bounded by the size of the original KB plus the size of the revision formula, are very unlikely to exist in the general case for various revision policies. Especially, while there can be exponentially many maximally consistent subbases of KB w.r.t. f, the size of their set-theoretic union is necessarily bounded by the size of KB. On the other hand, while the intersection of the logical closures of all maximally consistent subbases of KB w.r.t. f can be finitely encoded as the disjunction of all such subbases, the resulting revised base contains only one formula[3]; thus, the whole content of KB would be removed, would a second severe revision occur.

Because it is based on the computation of $Ker(KB, f)$, our approach is closely related to both WIDTIO base revision [41], and safe base revision [2]

[2]Several criteria based on subset maximality or cardinality maximality can be used to select some maximally consistent subbases in presence of an information preference, i.e. assuming that the KB is stratified.

[3]Or two, if the pair formed by such a disjunction and f is considered as the revised base.

(that coincide when no preference information is available [31], and share the two valuable features listed above). Indeed, in [41], a purely standard logic approach to revision (namely, WIDTIO, i.e. "When In Doubt Throw It Out") requires $Ker(KB, f)$ to be discarded in case of a severe revision: $KB \circ_{WIDTIO} f$ is defined as $KB \circ_{WIDTIO} f = (KB \setminus Ker(KB, f)) \cup \{f\}$.

Instead, we propose to weaken $Ker(KB, f)$. As we will see in the next section, this may have a great influence on the revised epistemic state in the context of iterated revisions.

5 INFERRING FROM A REVISED KNOWLEDGE BASE

Let $\leq$ be any pre-order among interpretations from which a preferred-models inference relation can be defined. Let us now define an inference relation $\mid\!\sim_{R,\leq}$ under which revised KBs will be interpreted. Actually, the inference relation will be nonmonotonic; it will interpret the Abnormality propositions from $AB_{Revision}$ and AB_{KB} separately, and successively. From a semantical point of view, the inference relation will use a concept of *preferred revision-preferred model*. Such models minimize abnormality according to two principles:

Principle 1.
Elements from $AB_{Revision}$ have a higher priority than elements from AB_{KB}. Elements of $AB_{Revision}$ encode unexpected abnormalities, corresponding to severe revisions. They must be interpreted in such a way that consistency is restored. This can easily constrain the set of possible truth values of the elements of AB_{KB}. Structural abnormalities from this latter set must also be minimized.

Principle 2.
The elements of $AB_{Revision}$ are prioritized, *à la* prioritized circumscription [24] but with one abnormality symbol per stratum, only. A higher priority is given to more recently introduced Abnormality propositions. Accordingly, every Ab_{R_i} is preferred $false$ (except the most recently introduced Abnormality proposition, which must be *true* to revise the last state of inconsistency).

Formally, let us assume without loss of generality that PS is totally ordered (using any order) and that interpretations I over $PROP_{PS}$ are represented as words over $\{0, 1\}$ s.t. the i^{th} bit of I is 1 iff the i^{th} symbol of PS is interpreted as 1 in I. For every subset E of PS, let $I[E]$ denote the restriction of I to the symbols of E. On this ground, we can define a total pre-order among the models of KB:

DEFINITION 5 (revision-preference). Let I and J be two models of KB. I

is said at least as *revision-preferred* as J, noted $I \leq_R J$, iff $I[AB_{Revision}] \leq_{lex} J[AB_{Revision}]$ holds, where $\leq_{lex}$ is the reverse lexicographic ordering on words over $\{0,1\}$, induced by $0 < 1$.

The revision-preferred models of KB are the minimal models of KB w.r.t. $\leq_R$.

DEFINITION 6 (revision-preferred model). The set of all revision-preferred models of KB is defined by $\mathcal{M}_R(KB) = min(\mathcal{M}(KB), \leq_R)$.

It is worth noting that all revision-preferred models I of KB coincide on $AB_{Revision}$. Such models can be characterized in the following constructive way. Let Ab_{R_z} be the lastly introduced symbol of $AB_{Revision}$ in KB:

Procedure Revision-preferred-models
Input: KB, $AB_{Revision}$, Ab_{R_z}
Output: `an interpretation` I `over` $PROP_{AB_{Revision}}$
```
begin
 for every Ab_{R_i} in AB_{Revision} from Ab_{R_z} downto Ab_{R_1} do
  if KB_{Ab_{R_i} <- false} is consistent
   then [[Ab_{R_i}]](I) = 0
   else [[Ab_{R_i}]](I) = 1
 return I
end
```

All the remaining symbols from $AB_{Revision}$ are set to 0 in the revision-preferred models I of KB[4]. Now, among the revision-preferred models, the preferred ones are those that minimize the structural abnormalities, according to any preset preferred-models semantics:

DEFINITION 7 (preferred revision-preferred model). Let $\leq$ be any pre-order among the models of KB. The set of all preferred revision-preferred models of KB is defined by $\mathcal{M}_{R,\leq}(KB) = min(\mathcal{M}_R(KB), \leq)$.

Finally, the corresponding preferred-models inference relation $\mid\!\sim_{R,\leq}$ can be defined as follows:

DEFINITION 8 (preferred revision-preferred inference). Let $\leq$ be any pre-order among the models of KB. $KB \mid\!\sim_{R,\leq} f$ iff f is satisfied in all preferred revision-preferred models of KB.

Clearly enough, in the situation where no severe revision has been performed, no symbol from $AB_{Revision}$ occurs in KB. Thus, every model is a revision-preferred one and the preferred revision-preferred models of KB are just the preferred models of KB, i.e. those that are minimal w.r.t. $\leq$. Contrastingly, in the situation where no structural abnormalities have been

[4] This is required by the definition but is not very significant since such remaining symbols do not occur in KB.

envisioned (i.e. $AB_{KB} = \emptyset$), the preferred revision-preferred models of KB are its revision-preferred models.

EXAMPLE 9 (continued). For simplicity, let us assume that $AB_{KB} = \emptyset$. Let $KB = \{\neg Bird \vee Fly, Bird\}$. We have:

1. $KB \mid\!\sim_{R,\leq} Fly \wedge Bird$. The (preferred) revision-preferred models of KB coincide with the models of KB.

2. $KB \circ_{full-meet} \neg Fly = \{\neg Bird \vee Fly \vee Ab_{R_1}, Bird \vee Ab_{R_1}, \neg Fly\} \mid\!\sim_{R,\leq} \neg Fly$. Once again, the (preferred) revision-preferred models I of $KB \circ_{full-meet} \neg Fly$ coincide with its models and satisfy $[\![Ab_{R_1}]\!](I) = 1$.

3. $(KB \circ_{full-meet} \neg Fly) \circ_{full-meet} Fly = \{\neg Bird \vee Fly \vee Ab_{R_1}, Bird \vee Ab_{R_1}, \neg Fly \vee Ab_{R_2}, Fly\} \mid\!\sim_{R,\leq} Fly \wedge Bird$. This time, the (preferred) revision-preferred models I of the revised base do not coincide with its models. Every I is s.t. $[\![Ab_{R_2}]\!] = 1$ and $[\![Ab_{R_1}]\!] = 0$.

As evoked before, our framework does not enable taking into account preference information about what subbases should be kept. Indeed, the two revision principles given above are to be applied to any KB, thus the way unexpected abnormalities from $AB_{Revision}$ are minimized does not depend on KB. Nevertheless, some preference information about structural abnormalities can be incorporated within our framework; it is supported by the given pre-order $\leq$, and different pre-orders can easily give rise to different inference relations $\mid\!\sim_{R,\leq}$ from revised KBs. Furthermore, some flexibility can be achieved in our framework by modifying the status of some occurrences of Abnormality propositions. Indeed, when an unexpected, revision-generated exception has been revealed, the knowledge engineer can easily decide to view it afterwards as a structural abnormality. This modification can easily change what can be inferred.

6 RATIONALITY AND COMPLEXITY ISSUES

It is time now to weight the pros and the cons of our approach, especially w.r.t. rationality and complexity issues.

From a formal point of view, it is not easy to check our non-standard approach against usual rationality postulates for belief revision. On the one hand, we use a nonmonotonic inference framework instead of the standard deductive one. On the other hand, the ontology is not constant since additional Abnormality propositions are introduced during the successive severe revision steps. As a consequence, our approach does not satisfy the inclusion postulate proposed by [16]: in the general case, $KB \circ_{full-meet} f$ is not a subset of $KB \cup \{f\}$. Also, standard AGM rationality postulates apply to *belief sets*, i.e. deductively closed sets of beliefs, whereas we consider *belief bases* (more precisely finite sets of formulas).

Adapting Nebel's technique [27] to our nonmonotonic framework, we can associate a belief set revision operator $*_{full-meet}$ to our belief base one. Let $Cn(KB) = \{h \text{ s.t. } KB \mid\!\sim_{R,\leq} h\}$ the nonmonotonic closure of KB given $\mid\!\sim_{R,\leq}$. A belief set, i.e. the set of beliefs of some agent, is the nonmonotonic closure of a belief base KB. The belief set operator $*_{full-meet}$ generated by $\circ_{full-meet}$ is given by $Cn(KB)^{*_{full-meet}} f = Cn(KB^{\circ_{full-meet}} f)$.

It is easy to show that $*_{full-meet}$ obeys the six basic rationality postulates by Alchourrón et al. [1]. However, the two supplementary postulates are not satisfied in the general case. This is not very surprising since WIDTIO base revision typically exhibits the same behaviour w.r.t. rationality postulates.

As evoked previously, our revision process does not entail an exponential increase of size for the revised KB as in many syntax-based approaches, especially [12], and it allows for iterated belief revision in a non-trivial way (unlike some techniques proposed so far, see e.g. [27]). Moreover, as it will be shown in the next section, it is far less destructive than WIDTIO in presence of iterated revision, hence adhering more closely to the principle of minimal change.

We believe that the way we hide pieces of information by weakening them can open new alternative research directions in the study of rationality postulates for iterated belief revision, which currently gives rise to an unsettled controversy (see e.g. [7] [18] [40] and [13]).

Let us now turn to computational aspects. Propositional belief base revision is intractable in the general case. This intractability concerns both the complexity of deciding whether a formula is implied by a revised belief base, and the complexity of representing a revised belief base. Thus, [31] analyzes in depth the first issue and shows that inference from a revised KB typically is at the first level or even at the second level of the polynomial hierarchy.

Like (full-meet) WIDTIO [41] and safe base revision [2], our approach suffers from a high worst case complexity. In order to simplify things, let us first assume that $AB_{KB} = \emptyset$ (in this situation, the chosen pre-order $\leq$ over models is not significant).

PROPOSITION 10. *Let KB, f be two formulas and $AB_{KB} = \emptyset$. Checking whether $KB \mid\!\sim_{R,\leq} f$ holds is Π_2^p-complete.*

Sketch of proof: In order to check that $KB \not\mid\!\sim_{R,\leq} f$, it is sufficient to guess a model I of KB, then check that it is a revision-preferred one, and that $I \not\models f$ holds. Clearly enough, checking that a given model I of KB is a revision-preferred one can be achieved in polynomial time on a Turing machine equipped with an NP oracle. Indeed, showing that I is not a revision-preferred model of KB can be achieved easily in nondeterministic polynomial time by guessing a model J of KB and checking that $J \leq_R I$

and $I \not\leq_R J$. Equivalently, I is a revision-preferred model of KB iff it coincides with all the revision-preferred models of KB on the symbols of $AB_{Revision}$, and the required truth values of these symbols can be computed thanks to a number of calls to SAT that is bounded by the size of KB (see the procedure `Revision-preferred-models` in the previous section). Accordingly, determining whether $KB \mid\sim_{R,\leq} f$ is in Π_2^p. Because inference from WIDTIO revised bases is a restriction of inference w.r.t. $\mid\sim_{R,\leq}$ (cf. next section), and inference from WIDTIO revised bases is Π_2^p-hard (see Lemma 6.2 from [11]), the Π_2^p-completeness of inference from $\mid\sim_{R,\leq}$ follows.

Now, in the case where AB_{KB} is not empty (so that the given preorder $\leq$ is significant), the complexity upper bound of the inference problem will be given by the complexity of checking whether a model I of KB is a preferred revision-preferred model of KB. If such a model checking problem belongs to coNP (which is the case for many preference relations $\leq$), checking $KB \mid\sim_{R,\leq} f$ still belongs to Π_2^p, hence it is Π_2^p-complete.

Focusing on the computation of revised KBs only, the main difficulty of both WIDTIO and our technique is the computation of $Ker(KB, f)$. Indeed, computing minimally inconsistent subbases of KB w.r.t. f can be hard; especially, we can easily prove that, given a finite set KB of propositional formulas and a revision formula f, checking whether a given subset of KB is a minimally inconsistent subbase of KB w.r.t. f is BH_2-complete. Moreover, from a worst case computational complexity point of view, checking the membership of a formula g from KB to $Ker(KB, f)$ is Σ_2^p-complete (this is a consequence of Theorem 8.2 from [11]). This makes the full-meet approach impractical in the worst case. Fortunately, from the practical side, things are not so bad.

7 APPROXIMATING KERNELS THANKS TO LOCAL SEARCH

A computationally valuable feature of our approach is the existence of an efficient local search-based heuristic for approximating kernels when KB and f are two CNF formulas (which is not so much an unrealistic assumption) [23].

Since this is just a heuristic technique, we cannot guarantee that the output of the proposed algorithm will actually be $Ker(KB, f)$ when the inputs are KB and f. Nevertheless, this does not question the practical utility of the technique. While local search techniques for SAT are not guaranteed to find out a model for every consistent KB, they enable proving the consistency of many of them, which cannot be achieved using a systematic method. In the same vein, experiments show that our heuristic technique can be used to derive or, at least, to focus the search on kernels in many situations.

Since our heuristic technique is based on local search, let us first recall the most representative local search algorithm, namely Selman et al.'s GSAT [38] [37]. This algorithm performs a greedy local search for a satisfying assignment of a set of propositional clauses. The algorithm starts with a randomly generated truth assignment. It then changes ("flips") the assignment of the variable that leads to the largest increase in the total number of satisfied clauses. Such flips are repeated until either a model is found or a preset maximum number of flips (MAX-FLIPS) is reached. This process is repeated as needed up to a maximum of MAX-TRIES times.

Procedure GSAT
Input: a set of clauses S, $MAX-FLIPS$, and $MAX-TRIES$
Output: a satisfying truth assignment of S, if found

```
begin
 for i := 1 to MAX-TRIES do
  I := a randomly generated truth assignment
  for j := 1 to MAX-FLIPS do
   if I is a model of S then return I
   x := a propositional variable s.t. a change in its truth
        assignment gives the largest increase (possibly negative)
        in the number of clauses of S that are satisfied by I
   I := I with the truth assignment of x reversed
 return ``no satisfying assignment found''
end
```

Let us stress that some variant (e.g. random) moves are inserted in order to escape from local minima. In the following, TSAT, a local search technique of our own, is used [22]. It makes use of a tabu list forbidding recurrent flips and appears competitive in most situations. One of the nice features of TSAT is that it drops many stochastic properties of other local search techniques and is purely deterministic as far as the initial interpretation is selected in a non-random way. However, the following experimental results can be obtained using most local search techniques, and are not TSAT specific.

Although such techniques prove efficient in showing the consistency of hard propositional CNF belief bases, they are clearly logically incomplete in the sense that they do not cover the whole search space and cannot thus directly prove that a formula is inconsistent. In this respect, when KB is inconsistent, GSAT-like algorithms provide us with a way to approximate $MAXSAT$, i.e. finding a cardinality maximal subbase of KB. Indeed, at each step, the unsatisfied clauses form a hitting set of the minimally inconsistent subbases of KB. Such a powerful computational approach to a maxichoice base revision policy is not investigated here (see [15]), as we focus on the full-meet one.

Some of us have discovered recently that the trace of local search algorithms can allow one to determine kernels very often when they fail to prove consistency within a preset amount of computing time [23]. More precisely, for each clause, taking each flip as a step of time, the number of times during which this clause is falsified is updated. A similar trace is also recorded for each literal occurring in the belief base, counting the number of times it has appeared in the falsified clauses. All the clauses from KB that are falsified t times more than the mean number of falsification per clause are delivered. t is an empirically fixed threshold. When the local search technique fails to prove consistency, it appears extremely often that the most often falsified clauses from KB form a superset of the set-theoretic union of the minimally inconsistent subbases of KB when KB is actually inconsistent. Likewise, literals that exhibit the highest scores take part in these subbases.

Accordingly, our heuristic is based on the following assumption:

Heuristic: The set S_{TSAT} of clauses from KB delivered by TSAT when it fails in finding a model of $KB \cup \{f\}$ is a superset of $Ker(KB, f)$.

Depending on the computational effort we are ready to spend, the validity of the heuristic can be tested by:

1. Checking that $KB \setminus S_{TSAT}$ is consistent. If this is not the case, the heuristic is wrong.

2. Checking that $S_{TSAT} \cup \{f\}$ is inconsistent. If this is not the case, the heuristic is wrong.

The consistency checks can be performed exactly (using a complete algorithm for SAT, e.g. a Davis-Putnam one [8]) or approximately using a local search algorithm (assuming that the input formula is inconsistent if no model has been found within a preset time). Note that the test phase can be achieved in practice quite efficiently.

In the case where the heuristic is found wrong, we can turn to a systematic, more computationally expensive technique, to compute $Ker(KB, f)$. Alternatively, we can modify the threshold or modify the parameters of T-SAT (the maximum numbers of tries and flips) and, in this latter case, run it again.

It is worth noticing that replacing $Ker(KB, f)$ by a superset S_{TSAT}, as it can be delivered by the first phase of our technique, does not question the consistency of the revised KB. It would just lead to an over-cautious revision policy that would weaken too many pieces of information. Such a policy could be adopted when limited computational resources justify a trade-off between the computational cost and a distance between the probable kernels and the real ones.

Let us stress that in the general case the heuristic provides us with an *approximation* of $Ker(KB, f)$ in the following sense: some formulas from

$Ker(KB, f)$ might not appear in S_{TSAT} and S_{TSAT} might contain formulas that do not belong to $Ker(KB, f)$. Clearly enough, if the heuristic is not found wrong, we cannot be sure that S_{TSAT} actually is a superset of $Ker(KB, f)$ (the test phase enables identifying *some* situations where the heuristic is wrong, but *not all of them*). However, under severe time constraints, such an approximation can prove useful as such.

Empirically, the validity of the heuristic depends on several aspects. First, in our nonmonotonic framework, inconsistency is the exception and should normally concern a few formulas, only. Moreover, in large real-world KBs, inconsistency is often due to a limited number of contradictory rules and data (unless total inconsistency of the modeled agent!), which allow a small $Ker(KB, f)$ to emerge very often. Such a phenomenon matches most philosophical and psychological models of beliefs. Also, the local search algorithm should "cover" the space of interpretations in the most uniform and systematic way. This can be achieved partly by a large number of tries, with initial interpretations that are sufficiently distant. Clearly, the accuracy of the heuristic depends also on the consumed computation time. We are currently working on an improved system that stops its computation when a stable (probable) $Ker(KB, f)$ has clearly emerged. Obviously enough, the accurracy of the heuristic also depends on the interdependence level between formulas from $Ker(KB, f)$ and the other ones: whenever a formula from $Ker(KB, f)$ is falsified, not too many formulas from $KB \backslash Ker(KB, f)$ should be falsified at the same time. This is related to a *knowledge locality* principle, which is often a valid assumption in actual KBs.

Finally, let us note that whenever the revision formula f does not contain any Abnormality proposition, no formula from KB among those that have been weakened so far can belong to $Ker(KB, f)$. Indeed, in this situation, every Ab_{R_i} literal occurs only positively in KB. Thus, it cannot be the case that a weakened formula belongs to a minimally inconsistent subbase of KB w.r.t. f (assuming that every Ab_{R_i} is *true* is always possible and sufficient to satisfy the weakened formulas). This remark is valuable from a computational point of view since it means that whenever sufficiently many severe revisions have been done, computing $Ker(KB, f)$ can be easier since the number of candidate formulas from KB will be smaller (all the formulas that have been weakened so far can be omitted).

The main remarkable feature of this approach lies in its effective tractability for very large applications. In spite of the fact that it does not require a subset of propositional clausal logic to be used (like the Horn one) as it is sometimes required and that it is nonmonotonic, it proves empirically efficient for very large KBs, at least when inconsistency can be related to a small subpart of KB.

Very comprehensive experimentations have been conducted to test the heuristic, using standard benchmarks for propositional reasoning and consistency checking that are taken from [10] or have been proposed in the

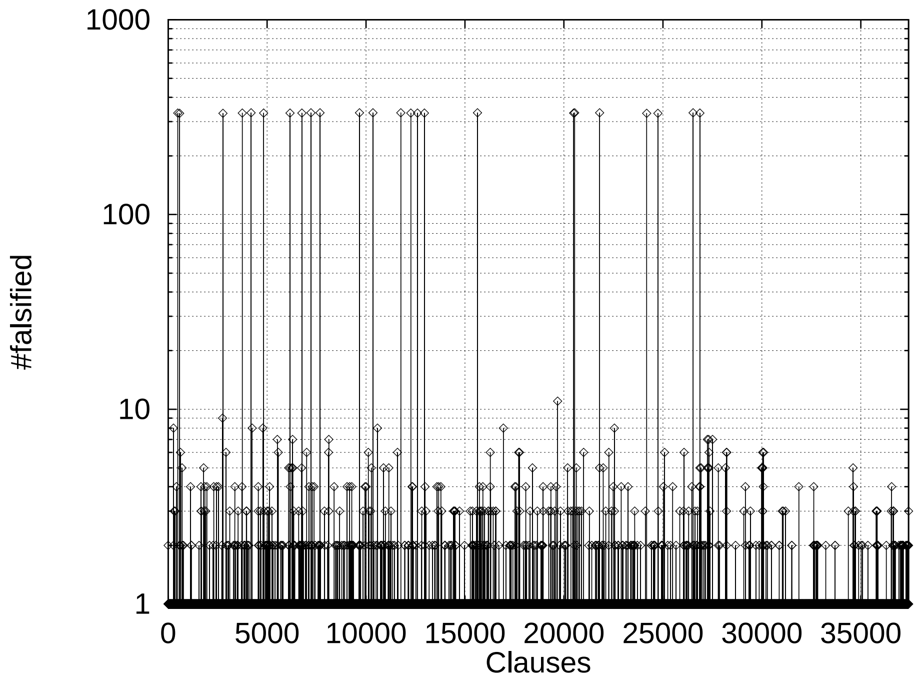

Figure 1. Monotonic framework.

meantime. All experiments have been conducted on Pentium 166 Mhz PCs. Let us just give a typical example. First, let us illustrate our approach when a purely monotonic knowledge is considered (i.e. $AB_{KB} = \emptyset$). We mixed a very large consistent propositional problem proposed by [17] representing a planning-related knowledge base with a globally inconsistent one proposed by [10]. These problems contain 37388 and 24 clauses (2883 and 9 variables), respectively. In our framework, we can interpret the first problem together with some clauses of the second one as representing a consistent KB, while the remaining part of the second one represents the clausal formula f to be inserted into KB. Clearly, $KB \wedge f$ is inconsistent.

TSAT (with a weight option [26] [37]) was run on $KB \wedge f$. Figure 1 shows (using a logarithmic scale) the trace (i.e. the score of the clauses) delivered after 2 seconds CPU computation time. The inconsistent subbase clearly emerges and is easily recognized. It consists of the 24 clauses with the highest scores. Weakening the subpart of this subbase that belongs to KB is now a straightforward task.

Let us now move to the nonmonotonic case. To our best knowledge, we are not aware of *large* propositional benchmarks in the nonmonotonic area representing real-world problems. Benchmarks in this domain have been

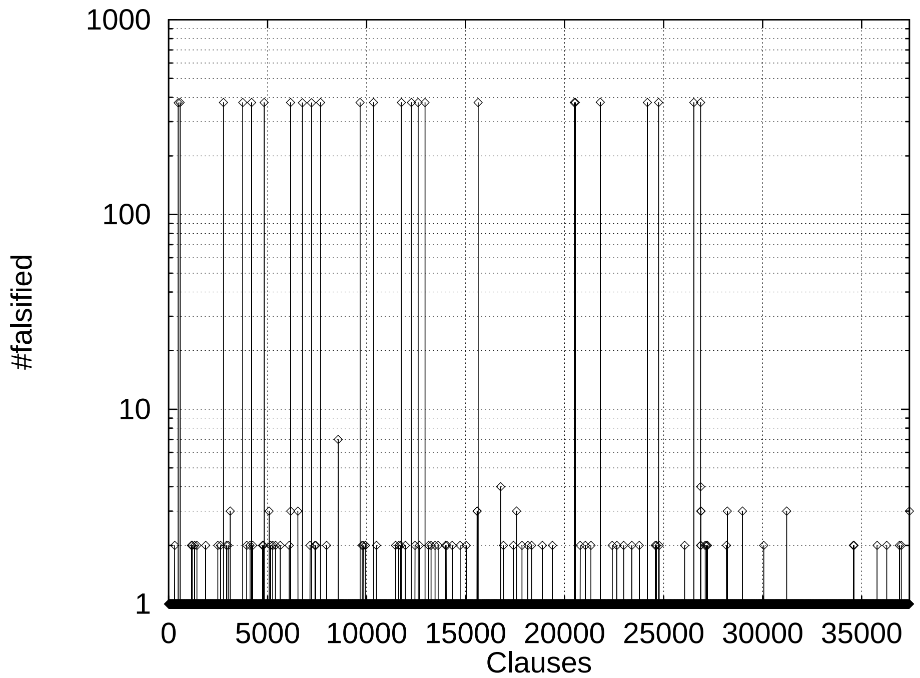

Figure 2. Nonmonotonic framework.

defined mainly in order to check the principles guiding the logics and not their computational properties. Therefore, we had to define our own benchmarks. For instance, we transformed monotonic ones so that they actually incorporate Abnormality propositions in some of their rules. Accordingly, the above knowledge base has been enriched in the following way. 10 % of clauses have been given their own additional literals Ab_i from AB_{KB}, representing possible future exceptions to the rules. TSAT was run again during 2 seconds. Figure 2 gives the trace of the computation. The introduction of Abnormality propositions does not modify the empirical results.

8 RELATED WORKS

As evoked previously, our approach to belief base revision is closely related to WIDTIO [41]. Especially, in the basic monotonic case and when only one-step of revision is considered, both our approach and WIDTIO coincide from an inferential point of view.

PROPOSITION 11. *Assume that KB, f and g do not contain any Abnormality proposition.* $KB \circ_{WIDTIO} f \models g$ *iff* $KB \circ_{full-meet} f \mid\!\sim_{R,\leq} g$

Sketch of proof: If $KB \cup \{f\}$ is consistent, then $Ker(KB, f) = \emptyset$, so $\mathcal{M}(KB \circ_{WIDTIO} f) = \mathcal{M}_{R,\leq}(KB \circ_{full-meet} f)$. Otherwise, every formula removed in the WIDTIO approach is weakened by Ab_{R_1} in our approach, and the converse also holds. Since Ab_{R_1} is necessarily *true* in the revised KB, the weakened formulas do not participate to inference. More formally, for every preferred revision-preferred model I of $KB \circ_{full-meet} f$, there exists a model J of $KB \circ_{WIDTIO} f$ s.t. for every symbol p of P, $[\![p]\!](I) = [\![p]\!](J)$, and *vice-versa*. The fact that g does not contain any Abnormality symbol is sufficient to end up the proof.

EXAMPLE 12 (continued). Let $KB = \{\neg Bird \vee Fly, Bird\}$. We have:

1. $KB \models Fly \wedge Bird$.
2. $KB \circ_{WIDTIO} \neg Fly = \{\neg Fly\} \models \neg Fly$.

However, our approach extends WIDTIO in at least two directions. First, WIDTIO is based on a purely standard logic framework. Second, our cautious approach is less "destructive" than WIDTIO in the sense that sources of inconsistency are "hidden" instead of being cancelled. This may lead to more "rationality", especially with respect to the principle of minimal change when iterated belief revision steps are processed. Indeed, WIDTIO is unnecessarily too much destructive.

EXAMPLE 13 (continued). $(KB \circ_{WIDTIO} \neg Fly) \circ_{WIDTIO} Fly = \{Fly\}$. Compared with our approach, let us note that $Bird$ is lost; actually when WIDTIO gives up a piece of information, it is forever (unless it is added again).

It is worth noting that the gain in expressiveness and "rationality" achieved by our approach (compared with WIDTIO) does neither necessarily lead to a degradation of the worst case complexity, with respect to inference from the revised KB. Indeed, inference from (full-meet) WIDTIO revised bases is Π_2^p-complete as well (see Theorem 5.21 from [31]). Moreover, once $Ker(KB, f)$ is derived, the revised base w.r.t. WIDTIO and our technique can be computed in polynomial time. Especially, removing inconsistency by weakening all its possible causes is not more expensive than deleting them.

Our approach can also be related to *linear base revision* from stratified bases [30]. In this approach, only one subbase of KB consistent with f has to be considered.

Let us associate to every (revised) belief base considered in our approach a stratified belief base s.t.:

- Originally (i.e. when no revision formula has been considered), all the formulas from KB are given the same priority, namely priority 0.

- Afterwards, when the i^{th} revision formula f is considered, every formula of $KB \setminus Ker(KB, f)$ is given priority i, f is also given priority i and all the remaining formulas keep their previous priorities.

Let KB_i denote the subset of KB consisting of all the formulas of priority i. Let z be the highest priority among the formulas in KB.

Procedure Inference-from-prioritized-KB
Input: `a stratified` KB
Output: `a subset` S_{max} `of` KB

```
begin
 S_max ← KB_z
 for every i from z − 1 downto 1 do
  if S_max ∪ KB_i is consistent
   then S_max ← S_max ∪ KB_i
 return S_max
end
```

A formula g is viewed as a consequence of such a stratified belief base KB iff it is a logical consequence of S_{max}. Interestingly, this occurs exactly when $KB \mid\!\sim_{R,\leq} g$, provided that $AB_{KB} = \emptyset$ and that the successive revision formulas f do not contain any Abnormality proposition. In other words, under these restrictive assumptions, inference in our approach and inference from such linearly revised stratified bases coincide. This can be easily explained by the fact that $AB_{Revision}$ is linearly ordered, and to each symbol Ab_{R_i} of $AB_{Revision}$ is associated a subset of the original KB, those formulas that are weakened by Ab_{R_i}, in so far as all the formulas from the subset can be considered as present in the revised KB when Ab_{R_i} is interpreted as $false$, and absent otherwise.

EXAMPLE 14 (continued).

1. We associate to the original $KB = \{\neg Bird \vee Fly, Bird\}$ the stratified base $KB^0 = \{\neg Bird \vee Fly(0), Bird(0)\}$. The corresponding set S^0_{max} is KB^0, from which $Bird$ and Fly are inferrable.

2. Now, KB is revised by $\neg Fly$. The corresponding revised base is associated to $KB^1 = \{\neg Bird \vee Fly(0), Bird(0), \neg Fly(1)\}$. The associated set S^1_{max} is $\{\neg Fly\}$, from which only $\neg Fly$ is inferrable.

3. Lastly, the resulting base is revised by Fly. The corresponding revised base is associated to $KB^2 = \{\neg Bird \vee Fly(2), Bird(2), \neg Fly(1), Fly(2)\}$. The associated set S^2_{max} is $\{\neg Bird \vee Fly, Bird, Fly\}$, from which both $Bird$ and Fly are inferrable.

While they give equivalent results in this restricted situation, our approach and linear base revision do not necessarily coincide in the general case, especially when AB_{KB} is not empty, or when the revision formulas may contain Abnormality symbols. Moreover, the fact that the ontology increases while severe revisions are performed enables *reasoning about the revision process itself* : the Ab_{R_i} symbols also play the role of temporal markers, allowing one to decide efficiently whether or not some initial piece of knowledge has been severely revised. For instance, the presence of the formula $\neg Bird \vee Fly \vee Ab_{R_1}$ in $(KB \circ_{full-meet} \neg Fly) \circ_{full-meet} Fly$ shows that the original piece of knowledge $\neg Bird \vee Fly$ has been subject to a severe revision.

9 CONCLUSIONS AND PERSPECTIVES

Most of the times, belief revision has been investigated from a theoretical point of view. As such, this domain has sometimes been criticized for its lack of computational theory-validating large-size applications. On the one hand, this paper is a modest contribution towards filling this missing facet of belief revision. On the other hand, non-conventional positions are adopted in this paper with respect to various facets of belief revision. As conclusive remarks, let us summarize and motivate them briefly again. We believe that each of them could open new fruitful paths of research.

- Belief revision should not be studied from the philosophical and cognitive sides of artificial intelligence only, but also from its (knowledge) engineering one. As such, specific new revision theories can be proposed that are guided by specific motivations: empirical computational efficiency, importance of the syntax and software quality properties, mainly.

- In this respect, we propose a combined use of nonmonotonic logics and belief revision. The nonmonotonic logic ingredients should be simple enough to allow an easy encoding of priorities, rules of default reasoning and exceptions. Inconsistencies should be prevented in advance as much as possible by an adequate encoding using this logic; severe revisions should concern inconsistencies that form a mere remainder of all inconsistencies that could occur if a standard representation and deductive logic were selected.

- Throwing out beliefs to restore consistency is unnecessary destructive. We propose to weaken beliefs instead. Quite remarkably, the weakening ingredients that we propose coincide with the nonmonotonic ones. Actually, beliefs that are inconsistent with respect to the newly introduced ones are interpreted as exceptional cases that do not actually occur. Weakening instead of throwing out opens new perspectives with respect to iterated belief revision considerations.

- Bad worst case complexity results for belief revision should not be misinterpreted. They do not prevent computational efficiency from being reached in many actual large-size applications.

- On the contrary, techniques based on local search techniques offer efficient heuristic tools to implement belief revisions techniques for very large propositional knowledge bases. The power of these techniques is currently unmatched, and offers interesting anytime properties that can be essential when time-critical belief revision processes are under consideration.

APPENDIX: BASIC ELEMENTS OF COMPUTATIONAL COMPLEXITY

One of the main purposes of complexity theory[5] is to classify problems according to their worst case requirements on computational resources depending on the size of the input. In this framework, a problem is a generic question, i.e., a set of specific instances. Specifically, a *decision problem* is one that has only "yes" and "no" as possible answers. Formally, a decision problem can be considered as a formal language consisting of the set of all its "yes" instances.

It is usually acknowledged that an *efficient algorithm* is one that runs in time polynomial in the size of the input (assuming a standard model of computation, e.g., a sequential deterministic Turing machine). Accordingly, a decision problem is considered *efficiently solvable* iff there exists an algorithm that can classify every instance of it in a number of computational steps that is polynomially bounded.

The class of all decision problems that are solvable in polynomial time is denoted by P. All the remaining ones require exponential time and are considered not efficiently solvable. In order to determine whether a problem is efficiently solvable or not, it is sufficient to point out an efficient algorithm to solve it, or to prove that such an algorithm cannot exist.

Unfortunately, for many problems, no polynomial time algorithms are known but it seems to be impossible to prove that super-polynomial time is actually required. For all these problems, the separation efficiently solvable vs. non-efficiently solvable is not fine-grained enough to classify them in a computationally valuable way: there is a need for more refined classes for problems that are not known in P.

[5]See e.g. [33] for details.

The notion of non-deterministic Turing machine is an important tool to achieve this goal. Thus, the class of all languages (encoding decision problems) that can be recognized in polynomial time by such a non-deterministic machine is denoted by NP. Because a deterministic Turing machine can be considered as a non-deterministic one, the inclusion $\mathsf{P} \subseteq \mathsf{NP}$ is established; however, the converse is the famous open problem: $\mathsf{P} \stackrel{?}{=} \mathsf{NP}$ (that is conjectured false). Among all the problems in NP, the hardest ones are those from which every problem in NP can be *polynomially many-one reduced*: such problems are referred to as NP-*complete*. If any of them has a polynomial algorithm, then $\mathsf{P} = \mathsf{NP}$ holds. Accordingly, it is believed that it is impossible to solve NP-complete problems in polynomial time. SAT, the problem of determining whether a propositional formula in conjunctive normal form is satisfiable, is the prototypical NP-complete problem [6]. Its complementary problem $UNSAT$ (consisting of determining whether a propositional formula in conjunctive normal form is unsatisfiable) is not necessarily in NP (in contrast to P, NP is not known to be closed under complementation). It is assigned to the class coNP that contains the complementary problems to problems of NP. It is conjectured that $\mathsf{NP} \neq \mathsf{coNP}$. In the following, let us note $\bar{L}$ the complementary problem of L.

To go further into the classification of non-efficiently solvable problems, another important tool is the notion of Turing machine (deterministic or non-deterministic) *with oracle*. Let X be a class of decision problems. P^X (resp. NP^X) is the class of all decision problems that can be solved in polynomial time using a deterministic (resp. non-deterministic) Turing machine that can use an oracle for deciding a problem $Q \in X$ for "free" (i.e., within a constant, unit time). On this basis, the classes Δ_k^p, Σ_k^p and Π_k^p are defined by:

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

Thus, $\Sigma_1^p = \mathsf{NP}$ and $\Pi_1^p = \mathsf{coNP}$. The *polynomial hierarchy* PH is the union of all Σ_k^p (for k non-negative integer). While it is easy to check that both $\Delta_k^p \subseteq \Sigma_{k+1}^p$, $\Delta_k^p \subseteq \Pi_{k+1}^p$, $\Sigma_k^p \subseteq \Delta_{k+1}^p$, and $\Pi_k^p \subseteq \Delta_{k+1}^p$ hold for every k, it is unknown whether the inclusions are proper or not (but it is conjectured that it is the case). Thus, it is strongly believed that the polynomial hierarchy does not collapse, i.e., is a truly infinite hierarchy.

In order to discriminate further among the problems from Δ_2^p, one can focus on the number of calls to an NP-oracle that are used. Thus, the classes BH_k and coBH_k are defined in the following way:

- $\mathsf{BH}_0 = \mathsf{P}$,
- $\mathsf{BH}_1 = \mathsf{NP}$,
- $\mathsf{BH}_2 = \{L_1 \cap \bar{L}_2 | L_1 \in \mathsf{NP}, L_2 \in \mathsf{NP}\}$,
- $\mathsf{BH}_{2k+1} = \{L_1 \cup L_2 | L_1 \in \mathsf{BH}_{2k}, L_2 \in \mathsf{NP}\}$ with $k \geq 1$,
- $\mathsf{BH}_{2k+2} = \{L_1 \cap \bar{L}_2 | L_1 \in \mathsf{BH}_{2k+1}, L_2 \in \mathsf{NP}\}$, with $k \geq 1$,
- $\mathsf{coBH}_k = \{L | \bar{L} \in \mathsf{BH}_k\}$.

The *boolean hierarchy* BH (over the complexity class NP) is the union of all BH_k (for k non-negative integer). It turns out that BH is equivalent to the class of problems that can be solved in deterministic polynomial time using a constant number of calls to an NP-oracle.

Acknowledgments
This paper has been supported in part by a *"Contrat d'Objectifs de la Région Nord/Pas-de-Calais"* and by the IUT de Lens. We are indebted to Salem Benferhat for so many interesting comments about a preliminary version of this paper and to Bertrand Mazure for his help with respect to the implementation side.

REFERENCES

[1] C.E. Alchourrón, P. Gärdenfors, and D. Makinson. On the logic of theory change: partial meet functions for contraction and revision. *Journal of Symbolic Logic*, 50:510–530, 1985.

[2] C.E. Alchourrón and D. Makinson. On the logic of theory change: safe contractions. *Studia logica*, 44:405–422, 1985.

[3] S. Benferhat, D. Dubois, and H. Prade. How to infer from inconsistent beliefs without revising. In *IJCAI'95*, pages 1449–1455, 1995.

[4] B. Bessant, É. Grégoire, P. Marquis, and L. Saïs. Combining nonmonotonic reasoning and belief revision: a practical approach. In *AIMSA'98*, volume 1480 of *Lecture Notes in Artificial Intelligence*, pages 115–128. Springer-Verlag, 1998.

[5] M. Cadoli, F. Donini, P. Liberatore, and M. Schaerf. The size of a revised knowledge base. In *PODS'95*, pages 151–162, 1995.

[6] S.A. Cook. The complexity of theorem-proving procedures. In 3^{rd} *Annual ACM Symposium on Theory of Computing*, pages 151–158, 1971.

[7] A. Darwiche and J. Pearl. On the logic of iterated belief revision. *Artificial Intelligence*, 89:1–30, 1997.

[8] M. Davis, G. Logemann, and D. Loveland. A machine program for theorem proving. *Journal of the Association for Computing Machinery*, 5:394–397, 1962.

[9] A. del Val. On the relation between the coherence and foundations theories of belief revision. In *AAAI'94*, pages 909–914, 1984.

[10] DIMACS, 1993. Second SAT challenge organized by the Center for Discrete Mathematics and Computer Science of Rutgers University, `http://dimacs.rutgers.edu/Challenges`.

[11] T. Eiter and G. Gottlob. On the complexity of propositional knowledge base revision, updates, and counterfactual. *Artificial Intelligence*, 57:227–270, 1992.

[12] R. Fagin, J.D. Ullman, and M.Y. Vardi. On the semantics of updates in databases. In *PODS'83*, pages 352–365, 1983.

[13] N. Friedman and J.Y. Halpern. Belief revision: a critique. In *KR'96*, pages 429–431, 1996.

[14] P. Gärdenfors. Belief revision and nonmonotonic logic: Two sides of the same coin? In *ECAI'92*, pages 768–773, 1992.

[15] É. Grégoire. Handling inconsistency efficiently in the incremental construction of stratified belief bases. Technical report, CRIL, 1999.

[16] S.O. Hansson. Reversing the Levi identity. *Journal of Phiolosophical Logic*, 22:637–669, 1993.

[17] H. Kautz and B. Selman. Pushing the envelope: planning, propositional logic, and stochastic search. In *AAAI'96*, pages 1194–1201, 1996.

[18] D. Lehman. Belief revision revisited. In *IJCAI'95*, pages 1534–1540, 1995.

[19] P. Liberatore and M. Schaerf. Relating belief revision and circumscription. In *IJCAI'95*, pages 1557–1553, 1995.

[20] P. Liberatore and M. Schaerf. The complexity of model checking for belief revision and update. In *AAAI'96*, pages 556–561, 1996.

[21] D. Makinson and P. Gärdenfors. Relations between the logic of theory change and nonmonotonic logic. In Fuhrmann and Moreau, editors, *Workshop on the logic of theory change*, pages 185–205. LNCS 465, Springer, 1991.

[22] B. Mazure, L. Saïs, and É. Grégoire. Tabu search for sat. In *AAAI'97*, pages 281–285, 1997.

[23] B. Mazure, L. Saïs, and É. Grégoire. Boosting complete techniques thanks to local search. *Annals of Mathematics and Artificial Intelligence*, 22:319–322, 1998.

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

[25] Y. Moinard. Revision and nonmonotonicity. *International Journal of Intelligent Systems*, 9, 1994.

[26] P. Morris. The breakout method for escaping from local minima. In *AAAI'93*, pages 40–45, 1993.

[27] B. Nebel. A knowledge level analysis of belief revision. In *KR'89*, pages 301–311, 1989.

[28] B. Nebel. *Reasoning and revision in hybrid representation systems*, volume 422 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, 1990.

[29] B. Nebel. *Belief Revision*, chapter Syntax-based approaches to belief revision, pages 53–88. Cambridge University Press, 1992.

[30] B. Nebel. Base revision operations and schemes: semantics, representation and complexity. In *ECAI'94*, pages 341–345, 1994.

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

[32] A. Newell. The knowledge level. *Artificial Intelligence*, 18:87–127, 1982.

[33] C.H. Papadimitriou. *Computational complexity*. Addison-Wesley, 1994.

[34] O. Papini. Revision in propositional calculus. In *ECSQAU'91*, volume 548 of *Lecture Notes in Artificial Intelligence*, pages 272–276. Springer-Verlag, 1991.

[35] B. Selman and H. Kautz. An empirical study of greedy local search for satisfiability testing. In *AAAI'93*, 1993.

[36] B. Selman, H. Kautz, and D. McAllester. Computational challenges in propositional reasoning and search. In *IJCAI'97*, pages 50–54, 1997.

[37] B. Selman, H.A. Kautz, and B. Cohen. Local search strategies for satisfiability testing. In *DIMACS Workshop on Maximum Clique, Graph Coloring, and Satisfiability*, 1993.

[38] B. Selman, H. Levesque, and D. Mitchell. A new method for solving hard satisfiability problems. In *AAAI'92*, pages 440–446, 1992.

[39] Y. Shoham. *Readings in Non-Monotonic Reasoning*, chapter A semantical approach to non-monotonic logics. Morgan Kaufmann, 1987.

[40] M.A. Williams. Iterated theory base change: a computational model. In *IJCAI'95*, pages 1541–1550, 1995.

[41] M. Winslett. *Updating Logical Databases*. Cambridge University Press, 1990.