The Complexity of Iterated Belief Revision

Paolo Liberatore

Università di Roma "La Sapienza",
Dipartimento di Informatica e Sistemistica
Via Salaria 113, 00198 Rome, Italy
email: liberato@dis.uniroma1.it
http://www.dis.uniroma1.it/∼liberato

Abstract. In this paper we analyze the complexity of revising a knowledge base when an iteration of this process is necessary. The analysis concerns both the classical problems of belief revision (inference, model checking, computation of the new base) and new issues, related to the problem of "committing" the changes.

1 Introduction

Belief revision is an active area in Databases, AI and philosophy. It has to do with the problem of accommodating new information into an older theory, and is therefore a central topic in the study of knowledge representation.

Suppose to have a knowledge base, represented with a propositional base K. When new information a arrives, the old base must be modified. If the new information is in contradiction with the old one, we must resolve the conflict somehow. The principle of minimal change is often assumed: the new knowledge base, that we denote with $K * a$, must be as similar as possible to the old one K. In other terms, the k.b. must be modified as little as possible. Many researchers give both general properties [AGM85,KM91] and specific methods (for example, [Dal88]) to resolve this conflict.

More recently, the studies have been focused on the iteration of this process. When, after a first revision, another piece of information arrives, the system of revision should take into account the first change. Roughly speaking, the program that makes this change on the knowledge base should not consider only facts which are currently true (the objective database), but also the revisions which have been done up to now.

In this paper we analyze the complexity of these frameworks. An analysis of the complexity of single-step (non-iterated) revision can be found in [EG92,LS96,Neb91,Neb94]. The key problem is of course to extract information from the revised k.b., or

> given a knowledge base K, a revision a, and a formula q, decide whether q is derivable from $K * a$, the revised knowledge base.

that is, given an old base and a revision, decide how much does it cost to extract information from the base obtained by revising the old one.

Iterating the process of revision, new problems arise. As will be clear in section 2, the operators defined in the literature need some extra information that depend on the history of the previous revisions. The size of this information becomes quickly exponential, thus we need a criterion to decide when the history becomes irrelevant, that is, when the changes can be "committed". At least, we need to know how many previous revisions the process must take into account.

2 Definitions

In this section we present the background and the terminology needed to understand the results of the rest of the paper. Throughout this paper, we restrict our analysis to a finite propositional language.

The *alphabet* of a propositional formula is the set of all propositional atoms occurring in it. Formulae are built over a finite alphabet of propositional letters using the usual connectives $\neg$ (not), $\vee$ (or) and $\wedge$ (and). Additional connectives are used as shorthands, $a \rightarrow b$ denotes $\neg a \vee b$, $a \equiv b$ is a shorthand for $(a \wedge b) \vee (\neg a \wedge \neg b)$ and $a \not\equiv b$ denotes $\neg(a \equiv b)$.

An *interpretation* of a formula is a truth assignment to the atoms of its alphabet. A *model* M of a formula f is an interpretation that satisfies f (written $M \models f$). Interpretations and models of propositional formulae will be denoted as sets of atoms (those which are mapped into true). For example, the interpretation that maps the atoms a and c into true, and all the others into false is denoted $\{a, c\}$. We use $\mathcal{W}$ to denote the set of all the interpretations of the considered alphabet. A *theory* K is a set of formulae. An interpretation is a model of a theory if it is a model of every formula of the theory. Given a theory K and a formula f we say that K *entails* f, written $K \models f$, if f is true in every model of K. The set of the *logical consequence* of a theory $Cn(K)$ is the set of all the formulas implied by it. Given a propositional formula or a theory K, we denote with $Mod(K)$ the set of its models. We say that a knowledge base K *supports* a model M if $M \in Mod(K)$, or equivalently $M \models K$. A formula f is satisfiable if $Mod(f)$ is non-empty. Let $Form$ be the operator inverse to Mod, that is, given a set of models A, $Form(A)$ denotes one of the equivalent formulas that have A as set of its models.

2.1 General Properties of Revision

Revision attempts to describe how a rational agent incorporates new information. This process should obey the principle of minimal change: the agent should make as little changes as possible on the old knowledge base.

As a result, when a new piece of information is consistent with the old one, the revision process should simply add it to the agent's beliefs. The most interesting situation is when they are inconsistent. The postulates stated by Alchourron, Gärdenfors and Makinson (AGM postulates for now on) provide base principles for this process.

Given a knowledge base K (represented with a deductively closed set of formulas) and a formula a representing a new information to be incorporated in it, they denote with $K * a$ the result of this process. The AGM postulates attempt to formalize the principle of minimal change.

Katsuno and Mendelzon in [KM91] give a reformulation of AGM's postulates in terms of formulas, instead of complete theories. For our purposes, this representation is more suitable. They proved that the AGM postulates are equivalent to the following ones (notice that now $*$ is an operator that takes two propositional formulas k and a, and the result $k * a$ is a propositional formula).

KM1 $k * a$ implies a
KM2 If $k \wedge a$ is satisfiable, then $k * a = k \wedge a$
KM3 If a is satisfiable, then $k * a$ is also satisfiable
KM4 If $k_1 \equiv k_2$ and $a_1 \equiv a_2$ then $k_1 * a_1 \equiv k_2 * a_2$
KM5 $(k * p_1) \wedge p_2$ implies $k * (p_1 \wedge p_2)$
KM6 If $(k * p_1) \wedge p_2$ is satisfiable, then $k * (p_1 \wedge p_2)$ implies $(k * p_1) \wedge p_2$

In the same paper they give an elegant representation theorem of the AGM revisions. Let $\mathcal{W}$ be the set of all the interpretations. A linear preorder over $\mathcal{W}$ is a reflexive, transitive relation $\leq$ over $\mathcal{W}$, with the additional property that for all $I, I' \in \mathcal{W}$ either $I \leq I'$ or $I' \leq I$.

In the KM formalization, each revision operator $*$ is associated to a family of linear preorders $O = \{\leq_f | f \text{ is a formula}\}$ that have the so called property of faithfulness.

Definition 1. A family $O = \{\leq_f | f \text{ is a formula}\}$ of linear preorders is said to be faithful if and only if for each formula f the following conditions hold (we use $I <_f J$ to denote that $I \leq_f J$ holds but $J \leq_f I$ does not).

1. If $I, J \in Mod(f)$ then $I <_f J$ does not hold.
2. If $I \in Mod(f)$ and $J \notin Mod(f)$ then $I <_f J$ holds.
3. If $f_1 \equiv f_2$ then $\leq_{f_1} = \leq_{f_2}$

Note that the above definition does not impose constraints over a relation $\leq_f$ when $I, J \notin Mod(f)$. The following theorem shows how an AGM revision can be represented with a faithful family of preorderings.

Theorem 2 ([KM91]). *A revision operator $*$ satisfies postulates* KM1-KM6 *if and only if there exists a faithful family $O = \{\leq_f \mid f \text{ is a formula}\}$ such that for all pairs of propositional formulas k and a, it holds*

$$Mod(k * a) = \min(Mod(a), \leq_k)$$

We can view $\leq_k$ as the order of plausibility induced by k, where $I <_k J$ means that I is considered more plausible than J to an agent believing k. The above formula means that the process of revision "chooses" the models of a that are considered more plausible.

2.2 Full Meet Revision

In this section we present a specific revision operator that has been proposed. The AGM's and KM's postulates imposes that when k and a are consistent then $k * a = k \wedge a$. Following Lehmann [Leh95], we call *mild* a revision that is consistent with the old information, *severe* otherwise.

The full meet revision [AGM85] is a "drastic" form of revision, that is, it preserves no information of the old k.b. if the revision is severe.

Definition 3. The full meet revision is defined as:

$$k *_{FM} \ a = \begin{cases} k \wedge a & \text{if consistent} \\ a & \text{otherwise} \end{cases}$$

As noticed by Alchourron, Gärdenfors and Makinson, this revision satisfies all the eight AGM's postulates and thus all the six KM's postulates. As a result, it can be expressed with a faithful family of linear preorderings. This family is:

$$FM = \{ \ \leq_k^{FM} | \ k \text{ is a formula, and } \leq_k^{FM} \text{ is defined as} \\ I \leq_k^{FM} J \text{ iff } I \in Mod(k) \text{ or } J \notin Mod(k) \ \}$$

Example 4. Let us consider the sequence $[p_1, p_2, p_3]$, where $p_1 = Form(\{I, L\})$, $p_2 = Form(\{L, M\})$ and $p_3 = Form(\{I, J, M\})$, and

$$\begin{array}{ll} I = \{a\} & J = \{b\} \\ L = \{a, b\} & M = \{a, b, c, d\} \end{array}$$

After revising p_1 with p_2 the current knowledge is $p_1 * p_2 = p_1 \wedge p_2$, since p_1 and p_2 are consistent. If a new information p_3 arrives, the result of the second revision is $(p_1 \wedge p_2) * p_3 = p_3$, since $p_1 \wedge p_2$ and p_3 are inconsistent. Note that all the information given by p_1 and p_2 has been lost.

2.3 Iterated Revision Operators

In this section we summarize some iterated revisions proposed. A sequence of formulas is denoted with $[p_1, \ldots, p_m]$. The null sequence (the sequence with no elements) is denoted with $[\,]$. If P and Q are two sequences, $P \cdot Q$ denotes their concatenation. In the previous section we have introduced the notion of single-step revision, denoted with $k * a$. In a similar manner, if we have a sequence of revisions w.r.t. the formulas $p_1, \ldots, p_m$, this will be denoted as $k * [p_1, \ldots, p_m]$. We assume, without loss of generality, that the initial formula k contains no information, and denote $() * [p_1, \ldots, p_m]$ simply with $*[p_1, \ldots, p_m]$.

Natural Revision. Boutilier [Bou93] defines a revision model as a triple $\mathcal{M} = \langle \mathcal{W}, R, \phi \rangle$, where $\mathcal{W}$ is a set of worlds with a valuation function ϕ, and R is a linear preorder relation over $\mathcal{W}$. The relation R represents the plausibility of worlds, that is, if wRw' then the world w is considered more plausible than the world w'. For our purposes, $\mathcal{W}$ can be viewed as the set of all the models, and $\phi(M)$ as the function of evaluation of the atoms associated with the model M. The natural revision is defined as follows.

Let $\mathcal{M}_{[p_1,\ldots,p_{m-1}]} = \langle \mathcal{W}, R, \phi \rangle$ be the revision model associated to a sequence $[p_1, \ldots, p_{m-1}]$. The natural revision maps this model into a new model $\mathcal{M}_{[p_1,\ldots,p_m]} = \langle \mathcal{W}, R', \phi \rangle$ such that

1. If $I \in \min(Mod(p_m), R)$ then $\langle I, J \rangle \in R'$ for each J, and $\langle J, I \rangle \in R'$ only if $J \in \min(Mod(p_m), R)$.
2. If $I, J \notin \min(Mod(p_m), R)$, then $\langle I, J \rangle \in R'$ if and only if $\langle I, J \rangle \in R$.

We assume that the null sequence [ ] is associated to the model $\mathcal{M}_{[\ ]} = \langle W, R_0, \phi \rangle$, where $R_0 = \mathcal{W} \times \mathcal{W}$. Once determined $\mathcal{M}_{[p_1,\ldots,p_m]} = \langle \mathcal{W}, R, \phi \rangle$, the natural revision is defined as

$$*_{NR}[p_1, \ldots, p_m] = Form(\min(\mathcal{W}, R))$$

Example 5. Consider the sequence $[p_1, p_2, p_3]$, where p_1, p_2 and p_3 are as in the previous example. We have $\mathcal{W}_{[\ ]} = \langle W, R_0, \phi \rangle$ where $R_0 = W \times W$ by definition. For the sequence with only the first element the above rules gives $W_{[p_1]} = \langle W, R_1, \phi \rangle$ where R_1 is $\{\langle I, L \rangle, \langle L, I \rangle, \langle I, J \rangle, \langle I, M \rangle, \langle L, J \rangle, \langle L, M \rangle\}$ (for the sake of simplicity we write only the pairs that contain two elements in $\{I, J, L, M\}$)

The sequence with two formulas has model $\mathcal{W}_{[p_1,p_2]} = \langle W, R_2, \phi \rangle$ where R_2 contains $\{\langle L, I \rangle, \langle I, M \rangle, \langle I, J \rangle\}$ and the other pairs implied by transitivity.

The complete sequence has model $\mathcal{W}_{[p_1,p_2,p_3]} = \langle W, R_3, \phi \rangle$, where R_3 contains $\{\langle I, L \rangle, \langle L, J \rangle, \langle L, M \rangle\}$. Thus, the result of the revision is $Form(min(W, R_3)) = Form(\{I\}) = p_1 \wedge p_3$. Note that p_2, although is consistent with p_3 and is considered more plausible than p_1, is not in the result.

Prioritized Iteration of Revision. Suppose to have a faithful family of orderings O, for example the family associated to the full meet revision. Consider a formula p_1 and its associated ordering $\leq^{FM}_{p_1}$. It holds $I \leq^{FM}_{p_1} J$ if, when we believe p_1, we consider the interpretation I more plausible than J.

Now, another formula p_2 arrives. What should be the plausibility ordering after revising p_1 with p_2? In other terms, what should be $\leq^{FM}_{[p_1,p_2]}$? Consider that p_2 is associated to the ordering $\leq^{FM}_{p_2}$. When both $I \leq^{FM}_{p_1} J$ and $I \leq^{FM}_{p_2} J$ hold (when these orderings agree), it is reasonable to assume $I \leq^{FM}_{[p_1,p_2]} J$.

Suppose instead that $I \not\leq^{FM}_{p_1} J$ but $I \leq^{FM}_{p_2} J$. Our guiding principle is that the new information is more reliable than the older one. As a result, $\leq^{FM}_{p_2}$ should be considered more reliable, and thus $I \leq^{FM}_{[p_1,p_2]} J$.

In general, we can state the following definition. Let $P = [p_1, \ldots, p_m]$ be a sequence. Its full meet prioritized associated ordering is defined inductively as follows ($I \cong J$ means that both $I \leq J$ and $J \leq I$ hold).

$$I \leq^{FM}_{[p_1,\ldots,p_m]} J \quad \text{if and only if} \quad \begin{cases} I <^{FM}_{p_m} J \ \text{ or} \\ I \cong^{FM}_{p_m} J \text{ and } I \leq^{FM}_{[p_1,\ldots,p_{m-1}J]} \end{cases}$$

The prioritized iteration of full meet revision is defined as

$$*_{PR(FM)}[p_1,\ldots,p_m] = Form(\min(\mathcal{W}, \leq^{FM}_{[p_1,\ldots,p_m]}))$$

Example 6. Consider again the sequence $[p_1, p_2, p_3]$ defined in the example 1. By definition $\leq^{FM}_{[p_1]}$ gives

$$I \cong^{FM}_{[p_1]} L <^{FM}_{[p_1]} J \cong^{FM}_{[p_1]} M$$

Applying the definition, $\leq^{FM}_{[p_1,p_2]}$ is

$$L <^{FM}_{[p_1,p_2]} M <^{FM}_{[p_1,p_2]} I <^{FM}_{[p_1,p_2]} J$$

and finally for $\leq^{FM}_{[p_1,p_2,p_3]}$ we obtain

$$M <^{FM}_{[p_1,p_2,p_3]} I <^{FM}_{[p_1,p_2,p_3]} J <^{FM}_{[p_1,p_2,p_3]} L$$

The result of the revision is $Form(min(W, <^{FM}_{[p_1,p_2,p_3]})) = Form(\{M\}) = p_2 \wedge p_3$.

Ranked Revisions. Lehmann in [Leh95] defines a (widening) ranking model as a function $\mathcal{R} : \mathcal{N} \to 2^{\mathcal{W}} - \emptyset$ (where $\mathcal{N}$ is the set of the non-negative integers), such that, for any $i, j \in \mathcal{N}$, if $i < j$ then $\mathcal{R}(i) \subseteq \mathcal{R}(j)$, and for any $I \in \mathcal{W}$ there exists a number n such that $I \in \mathcal{R}(n)$. The meaning of $\mathcal{R}$ is the following: if $I \in \mathcal{R}(i)/\mathcal{R}(j)$ and $J \in R(j)$ with $i > j$ then J is considered more plausible than I.

Each ranking model induces an iterated revision operator. Let $\mathcal{M}$ be a ranking model. Any sequence of revisions $P = [p_1, \ldots, p_m]$ defines a rank $r(P)$ that is a number, and a set of models $p(P)$.

The definition of revision is by induction on the length of the sequence. If $P = [\ ]$ then $p(P) = \mathcal{R}(0)$ and $r(P) = 0$, and $*_{RR(\mathcal{R})}P = Form(p(P))$. If $P = [p_1, \ldots, p_m]$ then, if there is in $p([p_1, \ldots, p_{m-1}])$ a model that satisfies p_m then $r(P) = r([p_1, \ldots, p_{m-1}])$ and $p(P) = p([p_1, \ldots, p_{m-1}]) \cap Mod(p_m)$. Furthermore, $*_{RR(\mathcal{R})}[p_1, \ldots, p_m] = Form(p(P))$.

If no interpretation in $p([p_1, \ldots, p_{m-1}])$ satisfies p_m, then $r(P)$ is the smallest number greater than $r([p_1, \ldots, p_{m-1}])$ such that there is a model in $\mathcal{R}(r(P))$ that satisfies p_m. Furthermore, $p(P) = \mathcal{R}(r(P)) \cap Mod(p_m)$ and $*_{RR(\mathcal{R})}[p_1, \ldots, p_m] = Form(p(P))$.

Notice that $\mathcal{R}$ is not changed by the introduction of new formulas in the sequence. The only effect of adding a formula in a sequence is to modify the set of current models $p(P)$, and the rank of the sequence $r(P)$. In all the other revision operators, instead, the plausibility ordering (however it is represented) is changed each time a formula is introduced in a sequence.

Example 7. In order to evaluate the revision of a sequence, we must specify the ranked model considered. Let $\mathcal{R}(M)$ be the number of the atoms in the interpretation M (or, the number of the atom that are mapped to true in the interpretation M). For the models I, J, L, and M defined in the example 1, we have $\mathcal{R}(I) = \mathcal{R}(J) = 1$, $\mathcal{R}(L) = 2$ and $\mathcal{R}(M) = 4$.

The sequence [ ] has by definition $r([\,]) = 0$ and $p([\,]) = \mathcal{R}(0)$. For $[p_1]$, consider that $p([\,])$ and $Mod(p_1)$ share no model, thus the definition gives $r([p_1]) = 1$ and thus $p([p_1]) = \mathcal{R}(1) \cap Mod(p_1) = \{I\}$.

The formula p_2 does not support the model I, thus we have to increase the rank, that becomes $r([p_1, p_2]) = 2$, and hence $p([p_1, p_2]) = \{L\}$.

As a result, the final revision with p_3 is severe, thus the rank is increased once again: $r([p_1, p_2, p_3]) = 3$. The result is thus

$$Form(p([p_1, p_2, p_3])) = Form(\mathcal{R}(3) \cap Mod(p_3)) = Form(\{I, J\}) = \neg p_2 \wedge p_1$$

Transmutations. Williams in [Wil94] defines a class of iterated revision that operates on a generic logic with some constraints. We are interested only in the propositional version of her work.

An ordinal conditional function (OCF for now on), is a function C from the set of models to non-negative integers, such that there are some models assigned to the number 0. Extend C to sets of models: $C(A) = \min(\{C(I) | I \in A\}, \leq)$ where $\leq$ is the classical ordering between numbers. Transmutations work on sequences of pairs $(formula, number)$. The number associated to a formula is the degree of acceptance of the formula. Thus, $(a, 1)$ means that we have a low degree of confidence on the fact that a is true, while $(b, 100)$ means that we are almost sure that b is valid.

Two specific transmutations have been defined.

Conditionalization. Given the OCF C assigned to a sequence of pairs $P = [(p_1, i_1), \ldots, (p_{m-1}, i_{m-1})]$, the new OCF C' that corresponds to the sequence $P' = [(p_1, i_1), \ldots, (p_m, i_m)]$ is defined as the function C' such that $C'(I) = C(I) - C(Mod(p_m))$ if $I \models p_m$, and $C'(I) = C(I) - C(Mod(\neg p_m)) + i_m$ otherwise. Furthermore, the OCF assigned to the null sequence [ ] is the function that associates 0 to each model. The conditionalization of a sequence $[(p_1, i_1), \ldots, (p_m, i_m)]$ is defined as

$$*_{CT}[(p_1, i_1), \ldots, (p_m, i_m)] = Form(\{I \in \mathcal{W} | C(I) = 0\})$$

where C is the OCF assigned to P.

Adjustment. Given the OCF C assigned to $P = [(p_1, i_1), \ldots, (p_{m-1}, i_{m-1})]$, the new OCF C' assigned to $P' = [(p_1, i_1), \ldots, (p_m, i_m)]$ is defined as

1. If $i_m = 0$ then if $C(I) = C(Mod(\neg p_m))$ and $I \models \neg p_m$ then $C'(I) = 0$, otherwise $C'(I) = C(I)$.
2. If $i_m \geq C(Mod(\neg p_m))$ then a) if $I \in Mod(p_m)$ and $C(I) = C(Mod(p_m))$ then $C'(I) = 0$; b) if $I \in Mod(\neg p_m))$ and $C(I) < i_m$ then $C'(I) = i_m$; c) otherwise $C'(I) = C(I)$.
3. If $0 < i_m < C(Mod(\neg p_m))$ then a) if $I \in mod(p_m)$ and $C(I) = C(Mod(p_m))$ then $C'(I) = 0$; b) if $I \in mod(\neg p_m)$ and $C(I) = C(Mod(\neg p_m))$ then $C'(I) = i_m$; c) otherwise $C'(I) = C(I)$.

The adjustment of a sequence $[(p_1, i_1), \ldots, (p_m, i_m)]$ is defined as:

$$*_{AT}[(p_1, i_1), \ldots, (p_m, i_m)] = Form(\{I \in \mathcal{W} | C(I) = 0\})$$

Note that if $i_m = 0$ then the revision with (p_m, i_m) is actually a contraction, that is, the formula p_m is not true in the revised knowledge base.

Example 8. Consider the sequence $[(p_1, 1), (p_2, 1), (p_3, 1)]$. The conditionalization associates to the sequence $[(p_1, 1)]$ the OCF C_1 such that $C_1(I) = C_1(L) = 0$ and $C_1(J) = C_1(M) = 1$.

Introducing $(p_2, 1)$ in the sequence, the OCF C_1 is transformed into a new OCF C_2 defined as $C_2(L) = 0$, $C_2(M) = C_2(I) = 1$ and $C_2(J) = 2$.

The OCF C_3 associated to the whole sequence is defined as $C_3(I) = C_3(M) = 0$, $C_3(L) = C_3(J) = 1$. As a result,

$$*_{CT}[(p_1, 1), (p_2, 1), (p_3, 1)] = Form(\{M | C_3(M) = 0\}) = Form(\{I, M\})$$

It is also easy to see that $*_{AT}[(p_1, 1), (p_2, 1), (p_3, 1)] = p_3$.

2.4 Computational Complexity

We assume that the reader is familiar with the basic concepts of computational complexity. We use the standard notation of complexity classes that can be found in [Joh90]. Namely, the class P denotes the set of problems whose solution can be found in polynomial time by a *deterministic* Turing machine, while NP denotes the class of problems that can be resolved in polynomial time by a *non-deterministic* Turing machine. The class coNP denotes the set of decision problems whose complement is in NP. We call NP-hard a problem G if any instance of a generic problem NP can can reduced to an instance of G by means of a polynomial-time (many-one) transformation (the same for coNP hard). The class LOGSPACE is defined as the class of problems that can be resolved by a deterministic Turing machine using only a logarithmic amount of space.

Clearly, P $\subseteq$ NP and P $\subseteq$ coNP. We assume, in line with the prevailing assumptions of computational complexity, that these containments are strict, that is P $\neq$ NP and P $\neq$ coNP. Therefore, we call a problem that is in P *tractable*, and a problem that is NP-hard or coNP-hard *intractable* (in the sense that any algorithm resolving it would require a super polynomial amount of time in the worst case).

We also use higher complexity classes defined using oracles. In particular P^A (NP^A) corresponds to the class of decision problems that are solved in polynomial time by deterministic (nondeterministic) Turing machines using an oracle for A in polynomial time (for a much more detailed presentation we refer the reader to [Joh90]). All the problems we analyze reside in the *polynomial hierarchy*, that is the analog of the Kleene arithmetic hierarchy. The classes Σ_k^p, Π_k^p and Δ_k^p of the polynomial hierarchy are defined by

$$\Sigma_0^p = \Pi_0^p = \Delta_0^p = \mathrm{P}$$

and for $k \geq 0$,

$$\Sigma_{k+1}^p = \mathrm{NP}^{\Sigma_k^p}, \quad \Pi_{k+1}^p = \mathrm{co}\Sigma_{k+1}^p, \quad \Delta_{k+1}^p = \mathrm{P}^{\Sigma_k^p}.$$

Notice that $\Delta_1^p = \mathrm{P}$, $\Sigma_1^p = \mathrm{NP}$ and $\Pi_1^p = \mathrm{coNP}$. The class $\Delta_2^p[\log n]$, often mentioned in the paper, is the class of problems solvable in polynomial time using a logarithmic number of calls to an NP oracle. The definitions of hardness and completeness for all these classes are similar to those of NP-hardness and completeness. The prototypical Δ_2^p-complete problem is the following [Kre88]:

Definition 9 (MAXLEXMOD). Given a propositional formula f using the letters of the alphabet $\mathcal{L} = \{x_1, \ldots, x_n\}$, decide if the maximal lexicographic model of f contains x_n.

Given a propositional formula t, its cardinality-based circumscription NCIRC(t) is the formula whose models are the minimal (w.r.t. number of atoms) models of t. The following problem [LS95] is complete for the class $\Delta_2^p[\log n]$.

Definition 10 (INFNCIRC). Given two propositional formulas t and q, decide if NCIRC$(t) \models q$.

We use also the class D^p, that is defined as the set of problems that can be expressed as deciding if a string x belongs to a language L_1/L_2, where deciding whether $x \in L_1$ is an NP problem and $x \in L_2$ is a coNP one (for more details see [Joh90]). A D^p complete problem is the following.

Definition 11 (CRITSAT). Given a set of clauses Π decide if it is unsatisfiable but any of its proper subsets is satisfiable.

The complexity of deciding $k * a \models q$ was studied by Eiter and Gottlob in [EG92]. Very briefly, for many revision operators it is Π_2^p-complete. The complexity of deciding whether $M \models k * a$ has been studied in [LS96].

3 Overview and Discussion of the Results

We have performed a thorough analysis of several computational aspects of iterated belief revision. In this section we explain which problems have been considered, and give an insight of the results. The results are presented in Table 1 and Table 2. Next section contains some proof sketches.

In the first table we report the complexity of the classical problems of belief revision, that is, deciding whether a formula is implied or not by a sequence of revisions (inference), whether a model is supported by the result of a revision (model checking), and the complexity of actually computing the result of revision, that is, to find the formula $*[p_1, \ldots, p_m]$.

More formally, the problem of inference is: given a sequence of formulas $p_1, \ldots, p_m$ and a formula q, decide if $*[p_1, \ldots, p_m] \models q$. The problem of model checking is to decide, given a sequence and a model M, if $M \models *[p_1, \ldots, p_m]$. The other problem, the computation, is to find a formula that is equivalent to $*[p_1, \ldots, p_m]$.

The first thing to notice is that for most operators the complexity of inference and model checking is Δ_2^p complete (and for all of them is *in* Δ_2^p). We remind

	inference	model checking	computation
Full Meet Iterated	$\Delta_2^p[\log n]$-hard in $\mathrm{LOGSPACE}^{\mathrm{NP}}$	D^p-hard in $\mathrm{LOGSPACE}^{\mathrm{NP}}$	NP equivalent
Natural Revision	Δ_2^p complete	Δ_2^p complete	NP equivalent
Adjustment	Δ_2^p complete	Δ_2^p complete	NP equivalent
Condizionalization	Δ_2^p complete	Δ_2^p complete	NP equivalent
Prioritized Full Meet Revision	Δ_2^p complete	coNP complete	NP equivalent
Ranked (general)	$\Delta_2^p[\log n]$-hard in Δ_2^p	D^p-hard in Δ_2^p	NP equivalent
Ranked (upper bound)	Δ_2^p complete	Δ_2^p complete	NP equivalent

Table 1. Complexity of inference, model checking and computation.

	equivalence to one formula	equivalence between sequences	minimality	minimal length
Full Meet Iterated	always	equal to inference	O(1)	1
Natural Revision	coNP complete	Δ_2^p complete	Δ_2^p complete	2^n
Adjustment	Δ_2^p complete	Δ_2^p complete	Δ_2^p complete	2^n
Condizionalization	D^p-hard in Δ_2^p	Δ_2^p complete	Π_2^p-hard in PSPACE	n
Prioritized Full Meet Revision	coNP complete	coNP complete	Π_2^p-hard in PSPACE	n
Ranked (general)	Δ_2^p	$\Delta_2^p[\log n]$-hard in Δ_2^p	Δ_2^p	4 if $el_{\mathcal{R}} = 0$
Ranked (upper bound)	Δ_2^p complete	Δ_2^p complete	Δ_2^p complete	4

Table 2. Complexity of equivalence and minimality.

that the class Δ_2^p is the class of the problems that can be computed with a polynomial number of calls to a procedure that solves an NP problem.

This happens because the revisions can be computed using plausibility orderings (represented by OCF, or revision models, or ranks), and the plausibility ordering induced by a sequence $[p_1, \ldots, p_i]$ can always be determined from the ordering associated to $[p_1, \ldots, p_{i-1}]$ by solving a polynomial number of resolution of NP problems.

Some interesting observations can be made on the third column of Table 1. The first is that the revision of a sequence has always size polynomial in the total size of the sequence. This is in contrast with the results about many forms of one-step revision [CDLS95], for which it is not always possible to find a polynomial size formula that represents the result of the revision. The second observation is that all the operators have the same complexity (namely, NP-equivalent). The reason is that the class of the NP-equivalent problems contains all the problems that are at least NP-hard, and for which a polynomial number of calls to an NP oracle suffices to determine the result. Both these facts hold for the iterated revisions introduced.

Note also that there are two rows for the ranked revision. This happens because the ranked revisions are actually a class of revisions or, using the terminology of Nebel [Neb94], a scheme of revision. Let $\mathcal{R}$ be a ranking model. We denote $*_{RR(\mathcal{R})}$ the corresponding revision. In order to compute the complexity of this operator, some hypothesis are required.

1. The rank r such that $\mathcal{R}(r) = \mathcal{W}$ can always be represented with $O(n)$ bits, where n is the number of atoms in the considered alphabet.
2. Deciding whether $I \in \mathcal{R}(r)$ is polynomial for any r.

These hypothesis restrict to a specific class of revisions. However, in this class there are different revision, that have different computational properties.

There are new aspects that must be considered in the iterated process of revision. Consider a sequence of revisions $[p_1, \ldots, p_m]$. The resulting k.b. $*[p_1, \ldots, p_m]$ in general does not suffice alone to evaluate a further revision if another formula p_{m+1} arrives. That is, if we store a $f \equiv *[p_1, \ldots, p_m]$ and forget the formulas $p_1, \ldots, p_m$, then we have not enough information to evaluate $*[p_1, \ldots, p_m, p_{m+1}]$. At any time, the revision process must take into account the history of the past revisions, or a suitable data structure representing it.

An important problem is decide when the "commitment" of the changes is possible, that is, when it is possible to replace a sequence $[p_1, \ldots, p_m]$ with a formula representing its revision $f \equiv *[p_1, \ldots, p_m]$.

More formally, we state the following definition.

Definition 12. Let $P = [p_1, \ldots, p_m]$ be a sequence and f be a formula. We say that P is equivalent to f w.r.t. a revision $*$ (written $P \equiv_* f$) if and only if for any sequence Q it holds $*P \cdot Q \equiv *[f] \cdot Q$.

More generally, it is useful to decide if a sequence is equivalent to another (hopefully shorter) one.

Definition 13. Let P, S be two sequences. We say that P is equivalent to S, w.r.t. a specific revision $*$, written $P \equiv_* S$, if and only if for any sequence Q it holds $*P \cdot Q \equiv *S \cdot Q$

The key problem is clearly to decide if a sequence is minimal, or is equivalent to a shorter one. Actually, for all the revisions it is possible to show that any sequence is equivalent to a sequence of a certain length independent to the length of the original sequence. For example, for the prioritized iteration of the full meet revision, any sequence is equivalent to a sequence of length n (the number of atoms in the formulas of the sequence). The complexity of resolving these problems is reported in Table 2.

The result of the last column for the ranked revision holds only if $el_{\mathcal{R}} = 0$, where $el_{\mathcal{R}}$ is the maximum number of consecutive empty levels of $\mathcal{R}$ (that is, it holds only if $\mathcal{R}$ has no empty level).

As one would expect, deciding the equivalence between sequences is more difficult than the problem of equivalence with one formula. Note however that this is not directly implied by the definitions: the first problem is to decide, given *two* sequences, if they are equivalent, while the second one is to decide, given *one* sequence if there exists a formula f equivalent to it. It could be possible, (for revisions not yet introduced) that this task of finding f makes the one formula equivalence problem more hard.

Deciding if a sequence is minimal turns out to be the most difficult problem, for each revision defined (exception made for the full meet revision, for which it is trivial). As a result, finding a minimal sequence equivalent to a given one is also an hard problem. However, a sequence does not need in general to be represented by the minimal one equivalent to it: a sub-optimal representation may meet the memory requirements.

The last column shows an interesting property of all the operators introduced: any sequence of length m is equivalent to a sequence whose length is independent to m. Namely, the length of these minimal sequences depends only on the number n of atoms in the considered alphabet. For long sequences ($m \gg n$) this can be useful.

4 Proof Sketches

In this section, we give proof sketches for some of the results of Table 1 and Table 2.

Theorem 14. *Inference for the natural revision is Δ_2^p complete.*

Proof (sketch). The membership in Δ_2^p follows from definition: the revision of a sequence is defined in terms of revision models, and the revision model of a sequence $[p_1, \ldots, p_m]$ can always be determined with a polynomial number of calls to a procedure that resolve an NP problem, if the revision model associated with $[p_1, \ldots, p_{m-1}]$ is known. However, representing a revision model in polynomial space is not a so trivial issue.

In order to prove hardness, we give a polynomial reduction from the problem MAXLEXMOD (see Section 2) to the problem of inference for the natural revision. Given a formula t on the alphabet $\{x_1, \ldots, x_n\}$, the natural revision applied to the sequence

$$P = [t, y_1, x_1 \equiv y_1, \ldots, y_n, x_n \equiv y_n]$$

has only one model I, and $I \cap \{x_1, \ldots, x_n\}$ is the maximal lexicographic model of t.

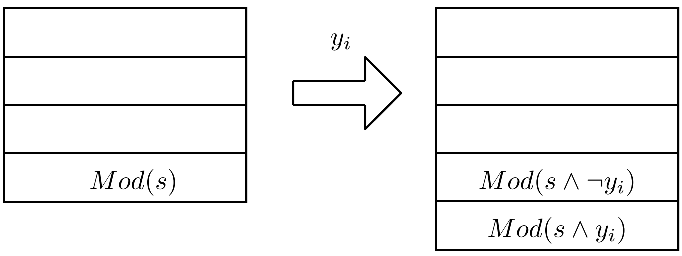
In order to prove it, consider that the maximal lexicographic model of t can be found with the following procedure:

```
s := t
for i := 1 to n do
  if s ∧ x_i is consistent
    then s := s ∧ x_i
    else s := s ∧ ¬x_i
return Mod(s)
```

We can prove that the revision of the sequence P above "simulates" this algorithm.

Essentially, what happens is that the class of the minimal models w.r.t. the relation R (of the current revision model $\langle \mathcal{W}, R, \phi \rangle$) is exactly the set of models of the formula s of the procedure. For the first formula of the sequence this is true since the minimal models of the revision model associated to $[t]$ are exactly the models of t.

The introduction of y_i in the sequence "splits" the current class of minimal models in two: the minimal models that contain y_i form the new class of minimal models, while the other models (those without y_i) form a new class just above.



The introduction of $x_i \equiv y_i$ modifies this relation. If s is consistent with x_i, then in the minimal class there are already models that satisfies $x_i \equiv y_i$. Thus those models become the new minimal models of the relation. On the other case (if $s \wedge x_i$ is inconsistent) the minimal models of $x_i \equiv y_i$ are in the class $Mod(s \wedge \neg y_i)$, thus the new minimal class is $Mod(s \wedge \neg y_i \wedge \neg x_i)$.

Now, since the only model of $*_{NR}P$ is the maximal lexicographical model of t, we have that MAXLEXMOD is equivalent to $*_{NR}P \models x_n$. □

Theorem 15. *Any sequence is equivalent (w.r.t. the natural revision) to a sequence of length 2^n, where n is the number of atoms in the considered alphabet. Moreover, there are sequences of length 2^n that are minimal.*

Proof (sketch). A given sequence P is equivalent to a sequence S if the revision models induced by them are identical. Now, consider the revision model $\langle \mathcal{W}, R, \phi \rangle$ associated with P, and define

$$\begin{aligned} \mathcal{W}_0 &= min(\mathcal{W}, R) \\ &\vdots \\ \mathcal{W}_i &= min(\mathcal{W}/\mathcal{W}_{i-1}, R) \end{aligned}$$

Since $\mathcal{W}$ contains 2^n interpretation (for an alphabet of size n), there are at most 2^n non-empty classes $\{\mathcal{W}_0, \ldots, \mathcal{W}_k\}$. Now, consider the sequence

$$S = [Form(\mathcal{W}_k), \ldots, Form(\mathcal{W}_0)]$$

One can prove that S and P induce the same revision model. □

5 Conclusions and Related Work

In this paper we have studied a fundamental issue of knowledge bases, that is, the complexity of a process of iterated revisions. Apart from the classical problems of belief revision (inference, model checking) we have introduced and studied new problems related to the iteration (commitment, equivalence, minimality).

Some questions are still open. For example, it is not clear if the problem of minimality for $*_{CT}$ and $*_{PR(FM)}$ is Π_2^p complete, PSPACE complete or lies in some point of the polynomial hierarchy. However, as we will show in the full version of this paper, this problem is related to the problem of counting the classes of equivalence induced by a certain relation, and thus it is probably better characterized with one of the "counting classes", such as #P.

Note also that in this paper we have considered only the problem of revision, and not the related problem of update [GO95,FH94].

Finally, we relate our work with [EG93]. In that paper, the authors analyze the complexity of inference in the (simple) iteration of the revision introduced by Fagin, Ullman and Vardi (also known as Ginsberg's revision). Other issues, related to the conditional logics are studied there. In our work, instead, we want to characterize the new semantics introduced for iterated revision. We also introduced the problems of equivalence as a measure of the possibility of committing the changes on a knowledge base.

Acknowledgments

We thank to Marco Cadoli and Marco Schaerf for their comments on earlier versions of this paper. We also thank to the anonymous referees for their valuable suggestions.

References

[AGM85] C. E. Alchourrón, P. Gärdenfors, and D. Makinson. On the logic of theory change: Partial meet contraction and revision functions. *Journal of Symbolic Logic*, 50:510–530, 1985.

[Bou93] C. Boutilier. Revision sequences and nested conditionals. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence (IJCAI-93)*, pages 519–525, 1993.

[CDLS95] M. Cadoli, F. M. Donini, P. Liberatore, and M. Schaerf. The size of a revised knowledge base. In *Proceedings of the Fourteenth ACM SIGACT SIGMOD SIGART Symposium on Principles of Database Systems (PODS-95)*, pages 151–162, 1995. Extended version available as Technical Report DIS 34-96, Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza", November 1996.

[Dal88] M. Dalal. Investigations into a theory of knowledge base revision: Preliminary report. In *Proceedings of the Seventh National Conference on Artificial Intelligence (AAAI-88)*, pages 475–479, 1988.

[EG92] T. Eiter and G. Gottlob. On the complexity of propositional knowledge base revision, updates and counterfactuals. *Artificial Intelligence*, 57:227–270, 1992.

[EG93] T. Eiter and G. Gottlob. The complexity of nested counterfactuals and iterated knowledge base revisions. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence (IJCAI-93)*, pages 526–531, 1993. To appear in *Journal of Computer and System Sciences*.

[FH94] N. Friedman and J. Y. Halpern. A knowledge-based framework for belief change: Part II: Revision and update. In *Proceedings of the Fourth International Conference on the Principles of Knowledge Representation and Reasoning (KR-94)*, pages 190–200, 1994.

[GO95] G. Grahne and Mendelzon A. O. Updates and subjunctive queries. *Information and Computation*, 2(116):241–252, 1995.

[Joh90] D. S. Johnson. A catalog of complexity classes. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume A, chapter 2. Elsevier Science Publishers (North-Holland), Amsterdam, 1990.

[KM91] H. Katsuno and A. O. Mendelzon. Propositional knowledge base revision and minimal change. *Artificial Intelligence*, 52:263–294, 1991.

[Kre88] M. V. Krentel. The complexity of optimization problems. *Journal of Computer and System Sciences*, 36:490–509, 1988.

[Leh95] D. Lehmann. Belief revision, revised. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95)*, pages 1534–1540, 1995.

[LS95] P. Liberatore and M. Schaerf. Relating belief revision and circumscription. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95)*, pages 1557–1563, 1995.

[LS96] P. Liberatore and M. Schaerf. The complexity of model checking for belief revision and update. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96)*, pages 556–561, 1996.

[Neb91] B. Nebel. Belief revision and default reasoning: Syntax-based approaches. In *Proceedings of the Second International Conference on the Principles of Knowledge Representation and Reasoning (KR-91)*, pages 417–428, 1991.

[Neb94] B. Nebel. Base revision operations and schemes: Semantics, representation and complexity. In *Proceedings of the Eleventh European Conference on Artificial Intelligence (ECAI-94)*, pages 341–345, 1994.

[Wil94] M. Williams. Transmutations of knowledge systems. In *Proceedings of the Fourth International Conference on the Principles of Knowledge Representation and Reasoning (KR-94)*, pages 619–629, 1994.