The Complexity of Model Checking for Belief Revision and Update

Paolo Liberatore and **Marco Schaerf**
Dipartimento di Informatica e Sistemistica
Università di Roma "La Sapienza"
Via Salaria 113 - Roma
email: {liberato,schaerf}@dis.uniroma1.it

Abstract

One of the main challenges in the formal modeling of common-sense reasoning is the ability to cope with the dynamic nature of the world. Among the approaches put forward to address this problem are *belief revision* and *update*. Given a knowledge base T, representing our knowledge of the "state of affairs" of the world of interest, it is possible that we are lead to trust another piece of information P, possibly inconsistent with the old one T. The aim of revision and update operators is to characterize the revised knowledge base T' that incorporates the new formula P into the old one T while preserving consistency and, at the same time, avoiding the loss of too much information in this process. In this paper we study the computational complexity of one of the main computational problems of belief revision and update: deciding if an interpretation M is a model of the revised knowledge base.

Introduction

During the last years, many formalisms have been proposed in the AI literature to model common-sense reasoning. Particular emphasis has been put in the formal modeling of a distinct feature of common-sense reasoning, that is, its dynamic nature. The AI goal of providing a logic model of human agents' capability of reasoning in the presence of incomplete and changing information has proven to be a very hard one. Nevertheless, many important formalisms have been put forward in the literature.

Given a knowledge base T, representing our knowledge of the "state of affairs" of the world of interest, it is possible that we are lead to trust another piece of information P, possibly inconsistent with the old one T. The aim of revision operators is to incorporate the new formula P into the old one while preserving consistency and, at the same time, avoiding the loss of too much information in this process. This process has been called *belief revision* and the result of revising T with P is denoted as $T * P$.

This "minimal change assumption" was followed by the introduction of a large set of specific revision operators. Among the others, we mention Fagin, Ullman and Vardi (Fagin, Ullman, & Vardi 1983), Ginsberg (Ginsberg 1986) and Dalal (Dalal 1988). A general framework for belief revision has been proposed by Alchourron, Gärdenfors and Makinson (Alchourrón, Gärdenfors, & Makinson 1985; Gärdenfors 1988). A close variant of revision is *update*. The general framework for update has been studied by Katsuno and Mendelzon (Katsuno & Mendelzon 1989; 1991) and specific operators have been proposed, among the others, by Winslett (Winslett 1990) and Forbus (Forbus 1989).

While most of the early work aimed at defining the appropriate semantics for revision and update, more recently some researchers investigated the computational complexity of reasoning with the operators introduced in the literature. The most complete complexity analysis has been done by Eiter and Gottlob in (Eiter & Gottlob 1992). More precisely, in the paper the authors address the problem of characterizing the complexity, in a finite propositional language, of the following problem:

> Given a knowledge base T, a new formula P and a query Q, decide whether Q is a logical consequence of $T * P$, the revised knowledge base.

In this paper we consider a distinct computational problem of belief revision and update. Consider a knowledge base represented by a set of propositional formulae T. Any such knowledge base can be equivalently represented by the set of its models, denoted as $\mathcal{M}(T) = \{M_1, \ldots, M_n\}$. We say that a model M is *supported* by a knowledge base T if and only if $M \in \mathcal{M}(T)$, or equivalently $M \models T$.

The problem we address in this paper is to decide if a model is supported by a revised knowledge base:

> Given a knowledge base T, a new piece of information P and a model M, decide if $M \in \mathcal{M}(T * P)$.

This problem is better known as model checking. There are several reasons why model checking is of interest in AI. First of all, as convincingly advocated by Halpern and Vardi in (Halpern & Vardi 1991) model-based representations are considered a viable alternative to the standard approach of representing knowledge in terms of formulae. In model-based representations the basic computational task is model checking, not inference. In this setting it is also very important to study the computational complexity of model checking.

The computational complexity of model checking is also strictly related to another computational aspect of knowledge representation formalisms: their *representational succinctness.* Some recent papers (Cadoli, Donini, & Schaerf 1995; Gogic *et al.* 1995; Cadoli *et al.* 1995) have shown that the succinctness of a knowledge representation formalism is strictly related to the complexity of model checking.

While the computational complexity of inference and model checking are related, there is no way to automatically derive the results for model checking from those already known for inference. In fact, as our results show, there are operators that have the same complexity w.r.t. query inference but with different complexity w.r.t. model checking.

Preliminaries

In this section we (very briefly) present the background and terminology needed to understand the results presented later in the paper. For the sake of simplicity, throughout this paper we restrict our attention 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, $\alpha \rightarrow \beta$ denotes $\neg\alpha \vee \beta$, $\alpha = \beta$ is a shorthand for $(\alpha \wedge \beta) \vee (\neg\alpha \wedge \neg\beta)$ and $\alpha \neq \beta$ denotes $\neg(\alpha = \beta)$.

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 1). A theory T 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 T and a formula F we say that T entails F, written $T \models F$, if F is true in every model of T. Given a propositional formula or a theory T, we denote with $\mathcal{M}(T)$ the set of its models. We say that a knowledge base T *supports* a model M if $M \in \mathcal{M}(T)$, or equivalently $M \models T$. A knowledge base T is consistent, written $T \not\models \bot$, if $\mathcal{M}(T)$ is non-empty.

Let $\mathcal{F}$ be the inverse operator of $\mathcal{M}$, that is, given a set of models A, $\mathcal{F}(A)$ denotes one of the formulae that have exactly A as set of models.

Belief Revision and Update

Belief revision is concerned with the modeling of accommodating a new piece of information (the revising formula) into an existing body of knowledge (the knowledge base), where the two might contradict each other. A slightly different perspective is taken by knowledge update. An analysis of the relative merits of revision and update is out of the scope of this paper, for an interesting discussion on the differences between belief revision and update we refer the reader to the work of Katsuno and Mendelzon (Katsuno & Mendelzon 1991). We assume that both the revising formula and the knowledge base can be either a single formula or a theory.

We now recall the different approaches to revision and update, classifying them into formula-based and model-based ones. A more thorough exposition can be found in (Eiter & Gottlob 1992). We use the following conventions: the expression $card(S)$ denotes the cardinality of a set S, and symmetric difference between two sets S_1, S_2 is denoted by $S_1 \Delta S_2$. If S is a set of sets, $\cap S$ denotes the set formed intersecting all sets of S, and analogously $\cup S$ for union; $min_{\subseteq} S$ denotes the subset of S containing only the minimal (w.r.t. set inclusion) sets in S.

Formula-based approaches operate on the formulae syntactically appearing in the knowledge base T. Let $W(T, P)$ be the set of maximal subsets of T which are consistent with the revising formula P:

$$\begin{aligned} W(T,P) \quad = \quad & \{T' \subseteq T \mid T' \cup \{P\} \not\models \bot, \\ & \neg\exists U : T' \subset U \subseteq T, U \cup \{P\} \not\models \bot\} \end{aligned}$$

The set $W(T, P)$ contains all the plausible subsets of T that we may retain when inserting P.

Ginsberg. Fagin, Ullman and Vardi in (Fagin, Ullman, & Vardi 1983) and, independently, Ginsberg in (Ginsberg 1986) define the revised knowledge base as a set of theories: $T *_G P \doteq \{T' \cup \{P\} \mid T' \in W(T, P)\}$. That is, the result of revising T is the set of all maximal subsets of T consistent with P, plus P. Logical consequence in the revised knowledge base is defined as logical consequence in each of the theories, i.e., $T *_G P \models Q$ iff for all $T' \in W(T, P)$, $T' \cup \{P\} \models Q$. In other words, Ginsberg considers all sets in $W(T, P)$ equally plausible and inference is defined skeptically, i.e., Q must be a consequence of each set.

A more general framework has been defined by Nebel in (Nebel 1991). We do not analyze its definitions.

WIDTIO. Since there may be exponentially many new theories in $T *_G P$, a simpler (but somewhat drastical) approach is the so-called WIDTIO (When In Doubt Throw It Out), which is defined as $T *_{Wid} P \doteq (\cap W(T, P)) \cup \{P\}$: see (Winslett 1989).

Note that formula-based approaches are sensitive to the syntactic form of the theory. That is, the revision with the same formula P of two logically equivalent theories T_1 and T_2, may yield different results, depending on the syntactic form of T_1 and T_2. We illustrate this fact through an example.

Example. Consider $T_1 = \{a, b\}$, $T_2 = \{a, a \rightarrow b\}$ and $P = \neg b$. Clearly, T_1 is equivalent to T_2. The only maximal subset of T_1 consistent with P is $\{a\}$, while there are two maximal consistent subsets of T_2, that are $\{a\}$ and $\{a \rightarrow b\}$.

Thus, $T_1 *_G P = \{a, \neg b\}$ while $T_2 *_G P = \{a \vee (a \rightarrow b), \neg b\} = \{\neg b\}$. The WIDTIO revision gives the same results.

Model-based approaches instead operate by selecting the models of P on the basis of some notion of proximity to the models of T. Model-based approaches assume T to be a single formula, if T is a set of formulae it is implicitly interpreted as the conjunction of all the elements. Many notions of proximity have been defined in the literature. We distinguish them between pointwise proximity and global proximity.

We first recall approaches in which proximity between models of P and models of T is computed pointwise w.r.t. each model of T. That is, they select models of T one-by-one and for each one choose the closest model of P. These approaches are considered as more suitable for knowledge update (Katsuno & Mendelzon 1991). Let M be a model, we define $\mu(M,P)$ as the set containing the minimal differences (w.r.t. set inclusion) between each model of P and the given M; more formally, $\mu(M,P) \doteq min_{\subseteq}\{M\Delta N \mid N \in \mathcal{M}(P)\}$.

Winslett. In (Winslett 1990) Winslett defines the models of the updated knowledge base as $\mathcal{M}(T{*_W}P) \doteq \{N \in \mathcal{M}(P) \mid \exists M \in \mathcal{M}(T) : M\Delta N \in \mu(M,P)\}$. In other words, for each model of T it chooses the closest (w.r.t. set-containment) model of P.

Borgida. Borgida's operator $*_B$ (Borgida 1985) coincides with Winslett's one, except in the case when P is consistent with T, in which case Borgida's revised theory is simply $T \wedge P$.

Forbus. This approach (Forbus 1989) takes into account cardinality: Let $k_{M,P}$ be the minimum cardinality of sets in $\mu(M,P)$. The models of Forbus' updated theory are $\mathcal{M}(T{*_F}P) \doteq \{N \in \mathcal{M}(P) \mid \exists M \in \mathcal{M}(T) : card(M\Delta N) = k_{M,P}\}$. Note that by means of cardinality, Forbus can compare (and discard) models which are incomparable in Winslett's approach.

We now recall approaches where proximity between models of P and models of T is defined considering globally *all* models of T. In other words, these approaches consider at the same time all pairs of models $M \in \mathcal{M}(T)$ and $N \in \mathcal{M}(P)$ and find all the closest pairs. Let $\delta(T,P) \doteq min_{\subseteq} \bigcup_{M\in\mathcal{M}(T)} \mu(M,P)$.

Satoh. In (Satoh 1988), the models of the revised knowledge base are defined as $\mathcal{M}(T *_S P) \doteq \{N \in \mathcal{M}(P) \mid \exists M \in \mathcal{M}(T) : N\Delta M \in \delta(T,P)\}$. That is, Satoh selects all closest pairs (by set-containment of the difference set) and then projects on the models of P.

Dalal. This approach is similar to Forbus', but global. Let $k_{T,P}$ be the minimum *cardinality* of sets in $\delta(T,P)$; in (Dalal 1988), Dalal defines the models of a revised theory as $\mathcal{M}(T *_D P) \doteq \{N \in \mathcal{M}(P) \mid \exists M \in \mathcal{M}(K) : card(N\Delta M) = k_{T,P}\}$. That is, Dalal selects all closets pairs (by cardinality of the difference set) and then projects on the models of P.

Wrong Variables Revisions

These two revision operators are model based and are based upon the hypothesis that the interpretation of a subset of the variables, denoted with Ω, was wrong in the old knowledge base T. The difference between them is based on a different definition of Ω. In Hegner's revision, Ω is set to the variables of P. The underlying idea is that the original knowledge base T was completely inaccurate w.r.t. everything mentioned in P.

Hegner. Let Ω be the variables of P. The models of Hegner's revised theory are defined as $\mathcal{M}(T *_H P) \doteq \{N \in \mathcal{M}(P) \mid \exists M \in \mathcal{M}(T) : N\Delta M \subseteq \Omega\}$. For further details see (Winslett 1989).

Weber's revision is slightly less drastic. It assumes that the letters whose interpretation was wrong are a subset of the letters of P, i.e., only those occurring in a minimal difference between models of T and P.

Weber. Same definition with $\Omega \doteq \cup\delta(T,P)$.

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 (Johnson 1990). 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).

Clearly, P $\subseteq$ NP and P $\subseteq$ coNP. We assume, following the mainstream 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 superpolynomial 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 (Johnson 1990)). All the problems we analyze reside in the *polynomial hierarchy*, introduced by Stockmeyer in (Stockmeyer 1976), 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 $\Pi_1^p = \mathrm{coNP}$. Moreover, $\Sigma_2^p = \mathrm{NP}^{\mathrm{NP}}$, that is the class of problems solvable in nondeterministic polynomial time on a Turing machine that uses for free an oracle for NP. The class

$P^{NP[O(\log n)]}$, 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 prototypical Σ_2^p-complete problem is deciding the truth of the expression $\exists X \forall Y.F$, where F is a propositional formula using the letters of the two alphabets X and Y. This expression is true if and only if there exists a truth assignment X_1 to the letters of X such that for all truth assignments to the letters of Y the formula F is true.

The complexity of deciding $T * P \models Q$ (where $*$ is one of $\{*_G, *_W, *_B, *_F, *_S, *_D\}$, T, P and Q are the input) was studied by Eiter and Gottlob in (Eiter & Gottlob 1992). Very briefly, in Dalal's approach, the problem is $\Delta_2^p[\log n]$-complete, while for all other operators it is Π_2^p-hard (Π_2^p-complete for most of them).

A computational analisys has been done in (Grahne, Mendelzon, & Revesz 1992), for an extension of Winslett's pointwise approach, showing both tractable and intractable cases.

Overview and discussion of the results

The results are presented in Figure 1. The table contains five columns. The second and third show the complexity of model checking when T is a general propositional formula, while the fourth and fifth show the Horn case. In the Horn case we assume that P and all formulae in T are conjunctions of Horn clauses.

The first thing to notice is that the computational complexity of model checking for almost all operators is at the second level of the polynomial hierarchy. This means that model checking for belief revision is much harder than model checking for propositional logic (feasible in polynomial time).

We now give an intuitive idea why these problems are all in Σ_2^p. For simplicity we only consider the model-based approaches but this applies to the other systems as well. In model-based approaches we have that M is a model of $T * P$ if and only if:

1. $M \models P$ and
2. There exists a model N of T that is "close" to M.

The first step is obviously feasible in polynomial time, while the second one requires a nondeterministic choice of N and for each choice checking the "closeness" of M and N. This check can be performed with a new nondeterministic choice.

There are three exceptions to this rule, in fact Dalal's operator is complete for $P^{NP[O(\log n)]}$, while Hegner's approach is NP-complete and Ginsberg's coNP-complete. The most surprising result is the complexity of model checking for Ginsberg's operator. In fact, as shown in (Eiter & Gottlob 1992), inference for $*_G$ is as difficult as inference for $*_F, *_W, *_B$ and $*_S$, while model checking turns out to be significantly simpler.

Restricting the size of the revising formula P has a dramatic effect on the complexity of model checking for $*_F$ and $*_W$. In fact, the complexity decrease by two levels. This phenomenon does not arise for query inference.

While restricting to Horn form generally reduces the complexity by one level there are two exceptions $*_D$ and $*_F$. The intuitive explanation is that these two operators use a cardinality-based measure of minimality that cannot be expressed as an Horn formula. On the other side, set-containment based minimality can be expressed with a Horn formula.

General Case

As said above, in this section we study the complexity of deciding whether $M \models T * P$, given T, P and M as input. Due to the lack of space we cannot present complete proofs of all the results.

We will show that operators that have the same computational complexity for the query answering problem (i.e., deciding if $T * P \models Q$) may have different complexity for the model checking problem (deciding if $M \models T * P$).

For Ginsberg's revision, model checking is easier than query answering. Namely, it is one level down in the polynomial hierarchy. The significance of this result is that Ginsberg's operator is only a coNP problem, despite its query answering problem has the same complexity of the others (Π_2^p-complete).

Theorem 1 *Deciding whether $M \models T *_G P$ is a coNP-complete problem.*

Proof (sketch). Given a model M, we first decide if $M \models P$. This can be done in polynomial time, and if this is not the case, the model is not supported by $T *_G P$. Now, we have that $M \models T *_G P$ if and only if M satisfies at least one element of $W(K, P)$. Let T' be set of the formulae in T that are satisfied by M. This is a consistent set, since it has at least one model (M). To show that T' is in $W(K, P)$ we have to prove that given any other formula f of T, the set $T' \cup \{f\} \cup \{P\}$ is inconsistent. Thus, we have the following algorithm to decide whether $M \models T *_G P$.

1. Check if $M \models P$ (if not return false).
2. Calculate $T' = \{f \in T \mid M \models f\}$
3. Decide, for any $f \in T - T'$, if the set $T' \cup \{f\} \cup \{P\}$ is inconsistent.

The first two steps only require polynomial time, while the third one is a set of (at most n) unsatisfiability problems that can be resolved by a single call of a coNP algorithm. This proves that the problem is in coNP (proof of hardness in the full paper). □

Even though Ginsberg's and WIDTIO revision are very similar, the complexity result we obtain is different.

Theorem 2 *Deciding $M \models T *_{Wid} P$ is Σ_2^p-complete.*

	General case		Horn case	
	P generic	P bounded	P generic	P bounded
Ginsberg $*_G$	coNP complete Th. 1	coNP complete Th. 8	P Th. 11	P Th. 11
Forbus $*_F$	Σ_2^p complete Th. 3	P Th. 7	Σ_2^p complete Th. 10	P Th. 15
Winslett $*_W$	Σ_2^p complete Th. 3	P Th. 7	NP complete Th. 12	P Th. 15
Borgida $*_B$	Σ_2^p complete Th. 3	coNP complete Th. 8	NP complete Th. 12	P Th. 15
Satoh $*_S$	Σ_2^p complete Th. 3	coNP complete Th. 8	NP complete Th. 12	P Th. 15
Dalal $*_D$	$\mathrm{P}^{\mathrm{NP}[O(\log n)]}$ complete Th. 4	coNP complete Th. 8	$\mathrm{P}^{\mathrm{NP}[O(\log n)]}$ complete Th. 10	P Th. 15
Hegner $*_H$	NP complete Th. 5	P Th. 7	P Th. 11	P Th. 11
Weber $*_{Web}$	Σ_2^p complete Th. 6	coNP hard in $\mathrm{P}^{\mathrm{NP}[O(1)]}$ Th. 8	NP hard in $\mathrm{P}^{\mathrm{NP}[O(\log n)]}$ Th. 13	P Th. 15
WIDTIO $*_{Wid}$	Σ_2^p complete Th. 2	Σ_2^p complete Th. 9	NP Th. 14	NP Th. 14

Figure 1: The complexity of deciding whether $M \models T * P$

We now turn our attention to the model-based operators. All the model-based operators are based on the principle that a model $M \models P$ satisfies the result of a revision $M \models T * P$ if and only if there is a model $I \models T$ such that I and M are sufficiently close each other.

It is not surprising that these methods have almost the same complexity (exception made for Dalal's that is a bit easier). However, although for query answering the complexity could be proved with a single proof, for the model checking problem each operator requires its own proof.

Theorem 3 *Deciding $M \models T * P$ is Σ_2^p-complete, where $* \in \{*_F, *_W, *_S, *_B\}$.*

Theorem 4 *Deciding whether $M \models T *_D P$ is $\mathrm{P}^{\mathrm{NP}[O(\log n)]}$-complete.*

We now establish the complexity of the operator that use a set of variables Ω whose observation is considered wrong, that is, Weber's and Hegner's ones.

Theorem 5 *Deciding whether $M \models T *_H P$ is NP-complete.*

The more complex definition of Weber's revision shows up in an higher complexity of model checking for his operator.

Theorem 6 *Deciding whether $M \models T *_{Web} P$ is Σ_2^p-complete.*

Bounded Case

In the previous section we investigated the complexity of evaluating the revised knowledge bases. As it turned out, for most of the operators this complexity is at the second level of the polynomial hierarchy. From an analysis of the proofs, it turns out that this behavior depends on the new formula P being very complex.

However, in practical applications it is reasonable to assume that the size of the new formula is very small w.r.t. the size of the knowledge base. In this section we investigate the impact this assumption has on the existence of compact representations. In particular, throughout this section we assume that the size of the new formula P is bounded by a constant (k in the sequel).

Under this assumption, we have the following results.

Theorem 7 *If $|P| \leq k$, the complexity of deciding whether $M \models T * P$ is polynomial, where $* \in \{*_F, *_W, *_H\}$.*

Theorem 8 *If $|P| \leq k$, the complexity of $M \models T * P$ is coNP-complete, where $* \in \{*_G, *_B, *_S, *_D, *_{Web}\}$.*

On the other side, WIDTIO semantics is not affected by the bound imposed on the size of P.

Theorem 9 *Even if $|P| \leq k$, the complexity of deciding whether $M \models T *_{Wid} P$ is Σ_2^p-complete.*

Horn Case

So far, we have considered revision of arbitrary knowledge bases. However, it could be significant to consider the complexity for the case in which both the basis are in the Horn form (i.e., they are conjunctions of Horn clauses), since this is the more used limitation used to make the problems about propositional calculus tractable. Furthermore, it is also interesting to find the complexity in the case in which both the Horn limitation and $|P| \leq k$ hold. This leads to the tractable (polynomial) cases of belief revision.

First of all, the cardinality-based revisions have the same complexity of the general (non-Horn) case (this theorem uses an observation of (Eiter & Gottlob 1992)).

Theorem 10 *If T and P are Horn formulae, the model checking problem for $*_F$ is Σ_2^p-complete and for $*_D$ is $\mathrm{P}^{\mathrm{NP}[O(\log n)]}$-complete.*

For Ginsberg's and Hegner's revisions, the complexity decreases: they become tractable.

Theorem 11 *If T and P are Horn formulae, the model checking problem for $*_G$ and $*_H$ is polynomial.*

Finally, for the revision operators of Satoh, Winslett and Borgida, the complexity decreases of one level.

Theorem 12 *If T and P are Horn formulae, the model checking problem for $*_S, *_W$ and $*_B$ is* NP*-complete.*

Finally, for Weber's revision the complexity decreases "almost" one level.

Theorem 13 *If T and P are Horn formulae, the model checking problem for $*_{Web}$ is* NP*-hard and in* $\mathrm{P}^{\mathrm{NP}[O(1)]}$.

Theorem 14 *If T and P are Horn formulae, the model checking problem for $*_{Wid}$ is in* NP.

If we also assume that the size of P is bonded by a constant k we obtain that model checking becomes tractable for all operators (except $*_{Wid}$).

Theorem 15 *If T and P are Horn formulae and $|P| \leq k$, the model checking problem for $*_G, *_F, *_W, *_B, *_S, *_D, *_H$ and $*_{Web}$ is polynomial.*

Conclusions

In this paper we have investigated a key issue of belief revision systems: their computational feasibility. Namely, we have studied, in a propositional language, the problem of deciding whether a model is supported by a revised knowledge base.

As it turns out, model checking for belief revision and update is far more complex than model checking for classic propositional logic. Furthermore, the complexity of model checking is not always related to the complexity of inference.

References

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

Borgida, A. 1985. Language features for flexible handling of exceptions in information systems. *ACM Trans. on Database Systems* 10:563–603.

Cadoli, M.; Donini, F. M.; Liberatore, P.; and Schaerf, M. 1995. The size of a revised knowledge base. In *Proc. of PODS-95*, 151–162.

Cadoli, M.; Donini, F. M.; and Schaerf, M. 1995. On compact representations of propositional circumscription. In *Proc. of STACS-95*, 205–216. Extended version as RAP.14.95 DIS, Univ. of Roma "La Sapienza", July 1995.

Dalal, M. 1988. Investigations into a theory of knowledge base revision: Preliminary report. In *Proc. of AAAI-88*, 475–479.

Eiter, T., and Gottlob, G. 1992. On the complexity of propositional knowledge base revision, updates and conterfactuals. *AIJ* 57:227–270.

Fagin, R.; Ullman, J. D.; and Vardi, M. Y. 1983. On the semantics of updates in databases. In *Proc. of PODS-83*, 352–365.

Forbus, K. D. 1989. Introducing actions into qualitative simulation. In *Proc. of IJCAI-89*, 1273–1278.

Gärdenfors, P. 1988. *Knowledge in Flux: Modeling the Dynamics of Epistemic States.* Cambridge, MA: Bradford Books, MIT Press.

Ginsberg, M. L. 1986. Conterfactuals. *AIJ* 30:35–79.

Gogic, G.; Kautz, H.; Papadimitriou, C.; and Selman, B. 1995. The comparative linguistics of knowledge representation. In *Proc. of IJCAI-95*, 862–869.

Grahne, G.; Mendelzon, A. O.; and Revesz, P. 1992. Knowledge transformations. In *Proc. of PODS-92*, 246–260.

Halpern, J. Y., and Vardi, M. Y. 1991. Model checking vs. theorem proving: A manifesto. In *Proc. of KR-91.* Also in Lifshitz V. *Artificial Intelligence and Mathematical Theory of Computation. Papers in Honor of John McCarthy*, Academic Press, San Diego, 1991.

Johnson, D. S. 1990. A catalog of complexity classes. In van Leeuwen, J., ed., *Handbook of Theoretical Computer Science*, volume A. Elsevier. chapter 2.

Katsuno, H., and Mendelzon, A. O. 1989. A unified view of propositional knowledge base updates. In *Proc. of IJCAI-89*, 1413–1419.

Katsuno, H., and Mendelzon, A. O. 1991. On the difference between updating a knowledge base and revising it. In *Proc. of KR-91*, 387–394.

Nebel, B. 1991. Belief revision and default reasoning: Syntax-based approaches. In *Proc. of KR-91*, 417–428.

Satoh, K. 1988. Nonmonotonic reasoning by minimal belief revision. In *Proc. of FGCS-88*, 455–462.

Stockmeyer, L. J. 1976. The polynomial-time hierarchy. *Theor. Comp. Sci.* 3:1–22.

Winslett, M. 1989. Sometimes updates are circumscription. In *Proc. of IJCAI-89*, 859–863.

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