The Complexity of Belief Update

Paolo Liberatore
Dipartimento di Informatica e Sistemistica
Universita di Roma "La Sapienza"
Via Salaria 113, 00198 Rome, Italy
Email: liberato@dis.uniromal.it
WWW: http://www.dis.uniromal.it/~liberato/

Abstract

Belief revision and belief update are two different forms of belief change, and they serve different purposes. In this paper we focus on belief update, the formalization of change in beliefs due to changes in the world. The complexity of the basic update (introduced by Winslett [1990]) has been determined in [Eiter and Gottlob, 1992]. Since then, many other formalizations have been proposed to overcome the limitations and drawbacks of Winslett's update. In this paper we analyze the complexity of the proposals presented in the literature, and relate some of them to previous work on closed world reasoning.

1 Introduction

The study of belief change has received considerable attention from the AI, databases and philosophy communities. Belief change deals with the incorporation of new facts into an agent's beliefs. There are two basic forms of belief change: belief revision and belief update. The difference lies in what is the source of incorrectness (if any) in the previous agent's beliefs.

The old beliefs of the agent may be mistaken or incomplete: in this case the usual approach is that of belief revision, captured by the AGM postulates [Alchourr6n *et a*/., 1985]. On the other hand, an agent's beliefs, while correct at one time, may become obsolete due to changes in the world. The basic treatment of updates is given in [Winslett, 1990], while a general framework is proposed in [Katsuno and Mendelzon, 1991].

Many drawbacks and limitations of the initial proposal of Winslett have been discovered. One regards how it treats disjunctive information: as in many approaches of non-monotonic reasoning, the update treats incorrectly a new piece of information that is in disjunctive form (such as $x_1 \vee X_2$). The next example shows one of these situations.

Example 1 *The RCH company deals with computers. It stores computers not yet sold in a warehouse composed by three separated rooms. The first room contains PCs, the second Macintoshes and the third Suns, X-terminals etc. By accident, a fire is set up in the room that contains PCs, and all the items it contains are moved to the other two rooms. The old database is (for our purposes)* ¬icontains(PC, R2)^¬contains(PC', *R3), which means only the first room contains PCs. After the fire destroyed that room, all we know is that the PCs has been moved in the rooms R2 and R3. Some items may have been moved to a room and some to the other one. The new piece of information to be incorporated is u —* contains(PC, *R2)* V contains(PC, *R3), which denotes the fact that at least one room (possibly both) must have PCs in it. The intended result of the update should be u itself, but Winslett's gives instead* contains(PC,*R2)*≠ contains(PC, *R3). This is not the intuitive result, since the possibility that some items have gone to R2 and some to RS should not be ruled out.*

Another drawback of the original formulation of update is that it never allows changes to the previous states: sometimes updates lead agents to revise their knowledge about the previous state. In these cases, update is related with revision as intended in the AGM framework [Alchourr6n *et a*/., 1985; Dalai, 1988]. The next example, shamelessly stolen from [Boutilier, 1995], shows this fact.

Example 2 *We have a beaker with an unknown liquid in it. We want to determine if it is an acid or a base. Initially, all we know is* acid ≠ base: *the content of the beaker may be an acid or a base, but not both.*

In order to determine which kind of substance we are dealing with, we drop a litmus paper in the beaker. After that, we discover that the liquid is an acid. Updating the old knowledge base we obtain (correctly) acid A ¬base. *The problem is that this fact is true not only after the experiment, but was also true* before *it: the liquid was acid before the test, although we did not know this. As a result, the initial state should be modified accordingly. However, Winslett's approach modifies only the new state, and does not make any change on the belief about the previous state: using this update, we can say, about the liquid before the test, only* acid ≠ base.

In this paper we analyze the complexity of some of the proposals introduced so far. The complexity of the

PMA approach (Possible Models Approach, also known as Winsiett's update) has been proved by Eiter and Gottlob [1992] to be at the second level of the polynomial hierarchy, namely Π_2. The problem considered there is the basic entailment, that is

> given a knowledge base Φ, an update u and a propositions] formula ψ, decide whether ψ is implied by w, the updated knowledge base

The paper is organized as follows: in the next section we give the basic definitions of propositional calculus, belief update, and the various definitions of update proposed so fax. In section 3 we study the updates proposed for solving the problem of disjunctive updates: we start showing how some of the semantics of update are related to previous work on closed world reasoning, and use the results on complexity of closed world reasoning to prove some complexity results of update. In section 4 we show the complexity of the remaining proposals. In section 5 we draw some conclusions.

2 Definitions

Throughout this paper, we assume a propositional language C over an alphabet of atoms X. Any piece of information (such as previous agents' beliefs, updates, etc.) is represented by a propositional formula, where not otherwise specified. An interpretation is a truth assignment of the atoms, that is, a function from atoms into the set {true,false}. We extend this assignment to propositional formulas in the usual way. An interpretation I is a model of a formula Φ if and only if the formula is true in that interpretation. We denote interpretations and models by sets of atoms (those mapped into true). The set of all the possible interpretations over the given alphabet is denoted by M. A formula is said to be complete if it has exactly one model. Given a formula Φ, we denote by $Var\ \Phi$ the set of atoms it contains.

We denote by $Mod(\phi)$ the set of all the models of the formula ϕ. We use also *Form* to denote the inverse function of *Mod*, that is, $Form(A)$ is a propositional formula whose set of models is A. This function is not unique, since equivalent formulas have the same set of models; however, this is unimportant for our purposes.

Given two interpretations I and J, we denote by $Diff(I,J)$ their symmetric difference, that is, $Diff(I,J) = I\backslash J \cup J\backslash I$. Intuitively, $Diff(I,J)$ is the set of atoms to which the interpretations I and J assign a different truth value. Given a set A and an ordering $<$, we denote by $\min(A,<)$ the set of the minimal elements of A w.r.t. the ordering $<$.

We sometimes use substitutions. A substitution is a set of pairs *atom/formula*. Given a substitution $S = \{x_i/\psi_i\}$, and a propositional formula μ, we denote by $\mu[S]$ the formula obtained from μ by replacing any occurrence of x_i with the corresponding ψ_i.

2.1 Winslett's Update

Consider a propositional formula ϕ representing the state of the world. This information is assumed to be correct, but not (necessarily) complete. When a change in the world occurs, this description of the world must be modified. The assumption behind belief update is that what we know about the change is a propositional formula μ that is true in the new situation.

Winslett's update operates on a model by model base. Let I be an interpretation, and define $\leq_I$ to be the ordering on interpretations defined as

$$J \leq_I Z \text{ iff } Diff(I,J) \subseteq Diff(I,Z)$$

The update of the old state ϕ when a new formula μ becomes true after a change is

$$\phi *_W \mu = Form(\bigcup_{I \in Mod(\phi)} \min(Mod(\mu), \leq_I))$$

This update was initially proposed in the context of databases, in which one can safely assume that all the updates are atomic, that is, the formula μ is always a literal. In AI this is often considered to be too restrictive. When μ is allowed to be any formula, and mainly when it is a disjunction of literals, the result of the Winslett's update may be intuitively incorrect. The example 1 of the last section shows a situation of this kind.

2.2 Minimal Change with Exceptions

This approach, introduced in [Zhang and Foo, 1996], attempts to overcome the non-intuitive treatment of disjunctive information of the original formulation. The definition is similar to that of Winslett's update, except for a set of atoms EXC that are not considered in the computation of the symmetric difference between interpretations.

First, define $D(\mu)$ as the set of all the minimal clauses implied by μ, that is

$$D(\mu) = \left\{ d \;\middle|\; \begin{array}{l} \mu \models d, \text{ and there is no other clause } d' \\ \text{s.t. } Var(d') \subset Var(d) \text{ and } \mu \models d' \end{array} \right\}$$

Then define $EXC(I,\mu)$, where I is an interpretation and μ a formula, as

$$EXC(I,\mu) = \bigcup_{\substack{d \in D(\mu) \\ I \notin Mod(d)}} Var(d)$$

Then, define $\leq_I^R$, where R is a set of variables, as the ordering among interpretations such that

$$J \leq_I^R Z \text{ iff } Diff(I,J)\backslash R \subseteq Diff(I,Z)\backslash R$$

finally, the update is defined as

$$\phi *_{MCE} \mu = Form(\bigcup_{I \in Mod(\phi)} \min(Mod(\mu), \leq_I^{EXC(I,\mu)}))$$

2.3 Minimal Change with Maximal Disjunctive Inclusion

Given a model I and a set of models S, define [Zhang and Foo, 1996]

$$Dis(I,S) = \left\{ S' \;\middle|\; \begin{array}{l} S' \in Mod(\mu) \text{ and} \\ \text{i. } \forall S_i \in S,\ Diff(I,S_i) \subseteq Diff(I,S') \\ \text{ii. there are no other } S'' \text{ satisfying i.} \\ \quad \text{but } Diff(I,S'') \subset Diff(I,S') \end{array} \right\}$$

then define the result of the update as

$$\phi *_{MCD} \mu = \bigcup_{\substack{I \in mod(\phi) \\ S \subseteq \min(Mod(\mu), \leq_I)}} Dis(I,S)$$

This approach is different from the previous one. Let ϕ and μ be as follows.

$$\begin{array}{rcl} \phi & = & \neg x_1 \wedge \neg x_2 \wedge \neg x_3 \wedge \neg x_4 \\ \mu & = & ((x_1 \neq x_2) \wedge \neg x_3 \wedge \neg x_4) \vee (x_1 \wedge x_2 \wedge x_3) \end{array}$$

The initial knowledge base ϕ has only one model $I = \emptyset$ (the interpretation that assigns false to all the variables). The only element of $D(\mu)$ that has not I has a model is $x_1 \vee x_2$. As a result, $EXC(I,\mu) = \{x_1, x_2\}$ and thus $\phi *_{MCE} \mu = (x_1 \neq x_2) \wedge \neg x_3 \wedge \neg x_4$.

On the other hand, $\phi *_{MCD} \mu = \mu \wedge \neg x_4$. Indeed, $\min(Mod(\mu), \leq_I) = \{\{x_1\},\{x_2\}\}$, that has three subsets. The first is $\{x_1\}$: we have $Dis(I, \{\{x_1\}\}) = \{\{x_1\}\}$. The second is $\{x_2\}$, for which $Dis(I, \{\{x_2\}\}) = \{\{x_2\}\}$. The last subset is $\{\{x_1\},\{x_2\}\}$ itself. One can verify that $Dis(I, \{\{x_1\},\{x_2\}\}) = \{\{x_1, x_2, x_3\}\}$. Thus, the models of the updated knowledge base are $\{x_1\}$, $\{x_2\}$, and $\{x_1, x_2, x_3\}$. The formula $\mu \wedge \neg x_4$ represents this set of models.

2.4 Update with Dependence Function

Assume that there is a function DEP from atoms to set of atoms. We use $DEP(x_i)$ to denote the set of atoms whose value depends on that of x_i. This means that a change in the value of x_i may affect the value of a $x_j \in DEP(x_i)$ (see [Herzig, 1996], where the operator is introduced, for more details.) It is assumed that for any atom x_i, it holds $x_i \in DEP(x_i)$. Define $DEP(\mu)$ as the union of $DEP(x_i)$ for each $x_i \in Var(\mu)$.

Now, define

$$CTX(\mu) = \{l_1 \wedge \cdots \wedge l_k \mid l_i = x_i \text{ or } \neg x_i,\ x_i \in DEP(\mu)\}$$

and the update is defined as

$$\phi *_{He} \mu = \bigvee_{\gamma \in CTX(\mu)} (\phi *_W \gamma) \wedge \mu$$

2.5 Generalized Update

This operator was introduced in [Boutilier, 1995] to formalize scenarios where an update may lead to revising the knowledge about the initial state. Define a ranking κ as a partial function from interpretations to integers, such that there is at least one interpretation I such that $\kappa(I) = 0$. The meaning of this ranking is that if $\kappa(I) < \kappa(J)$ then the interpretation I is believed to be more plausible than J, the world described exactly by I is more likely than the world described by J, from the agent's perspective. The models of ϕ must be exactly those ranked 0. When $\kappa(I)$ is not defined, the model I is considered unplausible. In this case, we write $\kappa(I) = \infty$.

We formalize the possible changes in the world as a set of possible *events* that may occur. For each event, we must specify what is the plausibility that the possible world represented by a model I will be transformed into the world represented by another model J. This plausibility is a function $\bar{\varepsilon}$ from triples ⟨event,model,model⟩ to integers. It can be a partial function, and $\bar{\varepsilon}(e, I, J) = \infty$ means that in the event e, the transition from I to J is considered implausible.

A generalized update model is a 5-tuple $\langle \mathcal{M}, \kappa, \bar{\varepsilon}, E, m \rangle$, where $\mathcal{M}$ is as usual the set of all the interpretations, κ is a ranking, E is a set of events, $\bar{\varepsilon}$ is as said above a function from triples ⟨event,model,model⟩ to integers, and m is a function from pairs ⟨event,model⟩ to integers, representing the plausibility of the occurrence of an event in the world represented by a given model.

Define the new rank of the interpretations after the update as

$$\kappa^\circ(J) = \min_{I \in \mathcal{M}, e \in E} \{\bar{\varepsilon}(e, I, J) + m(e, I) + \kappa(I)\}$$

This function $\kappa^\circ(J)$ induces an ordering in a natural way: $I \leq^\circ J$ if and only if $\kappa^\circ(I) \leq \kappa^\circ(J)$. One way to define the update is

$$\phi *_{GU} \mu = Form(\min(Mod(\mu), \leq^\circ))$$

that is, the models of the result are the models of μ whose rank κ° is minimal.

2.6 Possible Causes Approach

This approach was proposed in [Li and Pereira, 1996] and is based upon the idea that changes in the world are caused by *actions*. Thus, update can be formalized with ad-hoc languages such as *A* (introduced in [Gelfond and Lifschitz, 1993]) with an appropriate semantics. The main difference between reasoning about actions and belief update formalisms is that in the formers, actions are often assumed to be known in advance, and the aim of the theory is to understand what is true after they are performed. In belief update, actions are always assumed unknown, and the only evidence they have happened is their consequences.

The language *A* is built over an alphabet composed of fluent names (that are the facts, or what in the previous formalizations are the atoms of the propositional language), and three special symbols after, causes and if. A fluent expression is a fluent name or a fluent name preceded b¬y . Propositions are of two types: value propositions *F* after $A_1,..;A_m$ where *F* is a fluent expression and *A*\;...; A_m are actions; and effect propositions *A* causes *F* if $P_1, \ldots, P_m$, where *A* is an action and $F, P_1 \ldots, P_m$ are fluent expressions. If $m = 0$ the after proposition F after $A_1, \ldots, A_m$ is written initially F, and is called initially proposition.

In order to formalize the belief update in this framework, we must introduce a finite set of (linearly ordered) temporal points T, and two other kinds of propositions, the *happens* propositions and the *holds* propositions. Given a time point t and an action A, a happens proposition is written A happens at t, and means that the action A happens in the instant t. Given a fluent expression F and a time point t, a holds proposition has the form F holds at t, and means that F is true in the time point t.

A domain description is a set of value, effect, and happens propositions. Given a domain description D, it may happen that new information, represented by a holds proposition H, has to be incorporated. Due to the lack of space, we cannot introduce formally the semantics of $\mathcal{A}$ neither that of the belief update based on it. We refer to [Li and Pereira, 1996], where the update is introduced, for a more detailed explanation. The new holds proposition H could be not implied by D. In this case, we must find an explanation, i.e., a possible cause of the change. Formally, an explanation is a set of happens and initially propositions P such that $D \cup P$ implies H. To decide whether a proposition H' is implied by the updated description, we check whether $D \cup P$ implies H' for each minimal explanation P.

2.7 Closed World Reasoning

In the sequel we use two formalizations of the closed world assumption, namely the generalized world closed assumption and the *CURB*.

Given a propositional formula μ, the set of its free for negation atoms, written $FFN(\mu)$, is the set of the atoms that are false in all the minimal (w.r.t. set inclusion) models of μ. The generalized closed world assumption of μ is the formula $GCWA(\mu) = \mu \wedge FFN(\mu)$.

The *CURB* is introduced in [Eiter *et al.*, 1993] as a variant of the circumscription. In this paper we use a simplified version of it, the $CURB^1$, also introduced in the paper above. Given a formula μ, the $CURB^1(\mu)$ is defined as the formula whose models are the models J of μ such that there exists a set of minimal models S of μ such that a) J contains all the models of S, and b) there is no other J^1 of μ, contained in J, with the same property.

2.8 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 $P \neq coNP$.

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]). The classes Σ_k^p, Π_k^p and Δ_k^p of the polynomial hierarchy are defined by $\Sigma_0^p = \Pi_0^p = \Delta_0^p = P$, and for $k \geq 0$,

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

3 Disjunctive Information

In this section we analyze the definitions proposed to manage the problem of disjunctive information in belief update. The reason of their introduction is that updating $\neg x \wedge \neg y$ with $x \vee y$ one does not want, in general, to obtain $x \neq y$, that is the result given by Winslett's update.

This problem is analogous of that of other forms of non monotonic reasoning, notably circumscription and closed-word reasoning.

We start with the two proposals by Zhang and Foo. The minimal change with exception update have some similarities with the generalized closed world assumption. Indeed, *GCWA* can be reduced to MCE, and vice versa.

Theorem 1 *For any propositional formulas ϕ and μ, it holds*

$$GCWA(\mu) = \left(\bigwedge_{x_i \in X} \neg x_i \right) *_{MCE} \mu$$

$$\phi *_{MCE} \mu = \bigvee_{I \in Mod(\phi)} GCWA(\mu[S])[S]$$

where X is the set of atoms of μ and S is the substitution $\{x_i / \neg x_i | x_i \in I\}$.

This proves that $*_{MCE}$ is based upon the concept of generalized closed world assumption. As a result of the first statement, the update $*_{MCE}$ is Π_2^p hard, since the generalized closed world assumption is so, [Eiter and Gottlob, 1993], and it can be reduced polynomially to $*_{MCE}$. About the membership in that class, we note that the second statement gives a non-polynomial reduction. The following theorem shows the complexity of $*_{MCE}$.

Theorem 2 *The update $*_{MCE}$ is Π_2^p complete.*

This can actually proved in a very simple manner using the result of Π_2^p membership of *GCWA* given by Nebel [1996].

While the minimal change with exception is based upon the generalized closed world assumption, the minimal change with maximal disjunctive inclusion update is similar to the variant of circumscription called $CURB^1$. About the relation between $*_{MCD}$ and $CURB^1$, we have the following theorem.

Theorem 3 *For any propositional formulas ϕ and μ, it holds*

$$CURB^1(\mu) = \left(\bigwedge_{x_i \in X} \neg x_i\right) *_{MCD} \mu$$

$$\phi *_{MCE} \mu = \bigvee_{I \in Mod(\phi)} CURB^1(\mu[S])[S]$$

where S is the substitution $\{x_i/\neg x_i | x_i \in I\}$.

This theorem shows also that the $*_{MCD}$ is Π_2^p hard, since this is the complexity of $CURB^1$, as proved in [Eiter *et al.*, 1993].

The second reduction is not polynomial. However, with a little effort it is possible to obtain the exact complexity of $*_{MCD}$.

Theorem 4 *The update $*_{MCD}$ is Π_2^p complete.*

The update with dependence function is based on a function DEP that represents the causal relations between literals of the given alphabet. The easiest case is having all literals independent from each other, that is $DEP(x_i) = \{x_i\}$. In this case, the following theorem shows that the Herzig's proposal coincides with the standard semantics update $*_{SSU}$ (for an explanation of the standard semantics update see [Winslett, 1990]).

Theorem 5 *When DEP is defined as $DEP(x_i) = \{x_i\}$ for each x_i, then for each pair of propositional formulas ϕ and μ it holds $\phi *_{He} \mu = \phi *_{SSU} \mu$.*

A more general statement can be given. Recall that a way to represent the logical consequences of $\phi *_{SSU} \mu$ is $\phi[\{x_i/z_i | x_i \in Var(\mu)\}] \wedge \mu$, where z_i are new variables appearing nowhere else. For $*_{He}$ we have the following lemma.

Lemma 1 *For each triple of propositional formulas ϕ, μ and ψ, it holds $\phi *_{He} \mu \models \psi$ if and only if $\phi[\{x_i/z_i | x_i \in DEP(\mu)\}] \wedge \mu \models \psi$, where z_i's are variables that do not appear in any of the three formulas.*

Assuming that verifying whether $x_i \in DEP(\mu)$ is a polynomial task, we have the following result.

Theorem 6 *The update with dependence function is* coNP *complete.*

4 Complexity of Generalized Updates

In this section, we show the complexity of the two other updates, the generalized update by Boutilier [1995] and the possible causes approach by Li and Pereira [1996]. The former turns out to be simpler than the basic (Winslett's) update, while the latter is one level higher in the polynomial hierarchy.

About Boutilier's generalized update, we have to make some computational assumptions on the functions involved. Namely, we still write $\phi *_{GU} \mu$, and we implicitly assume that E is part of the input, but not $\mathcal{M}$ (this is the set of all the interpretations over the given alphabet). Also, we assume that the functions κ, $\bar{\varepsilon}$ and m of the revision model can be calculated in polynomial time. This means that given an event e and two models I and J, it must be possible to determine the integer $\bar{\varepsilon}(e, I, J)$ in polynomial time. The same for κ and m.

Under these assumptions, the entailment problem for this update is a Δ_2^p problem.

Theorem 7 *The update $*_{GU}$ is Δ_2^p.*

We give a very short explanation of this result. As for Dalal's revision, the check $\phi *_{GU} \mu \models \psi$ can be done in two steps: first, determine the value of the minimum $\kappa(I) + \bar{\varepsilon}(e, I, J) + m(e, I)$. Using this value $k_{\phi,\mu}$, verifying whether $\phi *_{GU} \mu \models \psi$ is a coNP complete problem. Indeed, guess a model I of ϕ, a model J of μ and an event e such that $\kappa(I) + \bar{\varepsilon}(e, I, J) + m(e, I) = k_{\phi,\mu}$ and $J \notin Mod(\psi)$. If this is possible, then $\phi *_{GU} \mu \not\models \psi$.

The first step, the determination of $k_{\phi,\mu}$, can be done with a polynomial number of calls to an NP oracle. Start with a guess for $k_{\phi,\mu}$, and verify (with the oracle) whether there exist I, J and e such that $\kappa(I) + \bar{\varepsilon}(e, I, J) + m(e, I)$ is less than this number. If this is true, the value of this $k_{\phi,\mu}$ must be decreased (otherwise, it must be increased).

About the hardness of this update, we note that one of the revision introduced, namely Dalal's revision [Dalal, 1988], can be polynomially reduced to the generalized update.

Theorem 8 *Let $\bar{\varepsilon}$ be the function such that $\bar{\varepsilon}(e, I, J)$ is the number of elements in $Diff(I, J)$. Then, using the revision models $\langle \mathcal{M}, \kappa, \bar{\varepsilon}, \{e\}, m \rangle$ where $k(I) = 0$ if $I \in model(\phi)$ and $\kappa(I) = \infty$ otherwise, and $m(e, I) = 0$, then $\phi *_{GU} \mu = \phi *_D \mu$, where $*_D$ is the Dalal's revision.*

As a corollary, since the revision of Dalal is $\Delta_2^p[\log n]$ complete, the complexity of the generalized update is $\Delta_2^p[\log n]$ hard. This result can be strengthened.

Theorem 9 *The generalized update is Δ_2^p complete.*

The Generalized Update has a lower complexity than Winslett's update. Thus (unless the polynomial hierarchy collapses) it is impossible to reduce the latter to the former in polynomial time.

We now consider the Possible Causes Approach. The problem we analyze is still to decide whether a formula is entailed by the old knowledge (in this case, a domain description) updated with a holds proposition.

Theorem 10 *Entailment under the Possible Causes Approach is Π_3^p complete.*

This is the only update considered in this work that turns out to be more complex than Winslett's. We try to explain this result.

Consider the definition of the possible causes approach restricted with the following hypothesis.

a. **There are no holds proposition in the domain description (the only holds proposition is the update).**

b. **Given the set of fluents $\{F_1, \ldots, F_n\}$, there are only n actions $\{A_1, \ldots, A_n\}$, and the only effect propositions in the domain description are**

$$A_i \textbf{ causes } F_i \textbf{ if } \neg F_i$$
$$A_i \textbf{ causes } \neg F_i \textbf{ if } F_i$$

c. Explanations are composed only of after propositions (initially propositions are not allowed in explanations).

In the holds propositions *F* holds at *t* we allow *F* to be a boolean formula on the alphabet of the fluent names (instead of a fluent expression). Under these restrictions, the update is still Π_3 complete.

The reason of the increase of the complexity is that, in the Winslett's approach, each model of Φ is updated separately, and this makes easy to verify if the result of an action entails the new piece of information.

Given a model J € *Mod(Φ),* in order to decide if J € min(Mod(μ), </), we have to check whether *Diff(I, J)* is minimal. Each element of *Diff(I, J)* can be interpreted as an action of the kind of (b) above, that changes the value of a literal. As a result, an explanation of the change is a minimal set of actions that maps / into a model of μ. Given a model I and a set of those actions, deciding whether the resulting interpretation *J* is a model of *μ* is a polynomial problem.

In the PCA, the possible causes affect the initial state *D* altogether: a possible explanation is a set of propositions *P* such that *D U P* implies the new fact to be incorporated. This is more complex than verifying a possible explanation in Winslett's approach.

5 Conclusions

In this paper we have investigated some of the semantics that have been introduced to formalize changes in the world of interest. The complexity of most of them turns out to be at the second level of the polynomial hierarchy. This is quite a remarkable result: indeed, these frameworks have been proposed to overcome drawbacks and limitations of the original semantics of update introduced by Winslett, but this does not seem to introduce any increase in the computational complexity.

Two approaches turn out to have a different complexity than Winslett's. The first is Boutilier's Generalized Update, that is easier (A£ complete), and the second is the Possible Causes Approach by Li and Pereira, that is more difficult (II3 complete). The first one is a generalization of Dalal's revision, and has a slightly higher complexity. The reason of the higher complexity of the Possible Causes Approach seems to be the globality of this operator: a change is considered not to affect each single possible initial state, but all the possible initial states altogether. This makes NP complete a subproblem that is polynomial in Winslett's approach.

Acknowledgments

We would like to thank Marco Schaerf for revising a draft version of this paper. Most of the work reported here has been done while the author was visiting AT&T Bell Labs. We thank Henry Kautz for his support during this period.

References

[Alchourr6n *et al,* 1985] C. E. Alchourr6n, P. Gardenfors, and D. Makinson. On the logic of theory change: Partial meet contraction and revision functions. *J. of Symbolic Logic,* 50:510-530, 1985.

[Boutilier, 1995] C. Boutilier. Generalized update: belief change in dynamic settings. In *Proc. of IJCAI-95,* pages 1550-1556,1995.

[Dalai, 1988] M. Dalai. Investigations into a theory of knowledge base revision: Preliminary report. In *Proc. of AAAL-88,* pages 475-479, 1988.

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

[Eiter and Gottlob, 1993] T. Eiter and G. Gottlob. Propositional circumscription and extended closed world reasoning are Fig-complete. *Theor. Comp. Sci.,* 114:231-245, 1993.

[Eiter *et al.,* 1993] T. Eiter, G. Gottlob, and Y. Gurevich. CURB your theory! In *Proc. of IJCAI-93,* pages 634-639,1993.

[Gelfond and Lifschitz, 1993] M. Gelfond and V. Lifschitz. Representing action and change by logic programs. *J. of Logic Programming,* 17:301-322, 1993.

[Herzig, 1996] A. Herzig. The PMA revised. In *Proc. of KR-96,* pages 40-50, 1996.

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

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

[Li and Pereira, 1996] R. Li and L. Pereira. What is believed is what is explained. In *Proc. of AAAI-96,* pages 550-555,1996.

[Nebel, 1996] B. Nebel. How hard is it to revise a knowledge base? Technical Report 83, Albert-Ludwigs-Universitat Freiburg, Institut fur Informatik, 1996.

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

[Zhang and Foo, 1996] Y. Zhang and N. Foo. Updating knowledge bases with disjunctive information. In *Proc. of AAAI-96,* pages 562-568,1996.