What Is Believed Is What Is Explained (Sometimes)

Renwei Li and **Luís Moniz Pereira**
Department of Computer Science
Universidade Nova de Lisboa
2825 Monte da Caparica, Portugal
{renwei | lmp}@di.fct.unl.pt

Abstract This paper presents a formal and computational methodology for incorporation of new knowledge into knowledge bases about actions and changes. We employ Gelfond and Lifschitz' action description language $\mathcal{A}$ to describe domains of actions. The knowledge bases on domains of actions are defined and obtained by a new translation from domain descriptions in $\mathcal{A}$ into abductive normal logic programs, where a time dimension is incorporated. The knowledge bases are shown to be both sound and complete with respect to their domain descriptions. In particular, we propose a possible causes approach (PCA) to belief update based on the slogan: What is believed is what is explained. A possible cause of new knowledge consists of abduced occurrences of actions and value propositions about the initial state of the domain of actions, that would allow to derive the new knowledge. We show how to compute possible causes with abductive logic programming, and present some techniques to improve search efficiency. We use examples to compare our possible causes approach with Ginsberg's possible worlds approach (PWA) and Winslett's possible models approach (PMA).

Introduction

Consider an agent having a knowledge base K about a domain of actions. Suppose that the agent somehow observes a new fact N about the domain. Then, how will the agent incorporate the new knowledge N into her knowledge base K? The task of incorporating N into K is generally called belief revision/update.

A representative of the work on belief revision is the so-called AGM theory, characterized by eight postulates. As is well-known, some belief revision operators satisfying the AGM postulates give intuitively incorrect results. In the literature, some other belief revision/update operators have been proposed to solve practical problems. These belief revision/update operators operate on a consistent finite set of formulas (called belief base, whose deductive closure is called belief set), and can be roughly classified into two types: the syntax-based approach [9, 13] and the semantics-based approach [4, 9]. The syntax-based approach to belief base revision selects set-inclusion maximal subsets of a belief base that do not allow for the derivation of the negation of the new sentence. The semantics-based approach to belief base revision operates on models of belief bases and selects those models satisfying the new sentence and differing minimally from the models of the original belief base.

Among all these belief revision operations there are two common criteria: Consistency should be preserved and change should be *minimal*. Different approaches vary with different definitions of the minimality in change. In this paper we advocate another criterion: A possible cause explaining the new fact and all the effects of the possible cause should be added and all cause-effect relations are maintained. Before a new fact N is added to the knowledge base K, the agent should look for all possible causes of N. If the agent admits some cause C of N, then the agent should also add all the effects of C besides N. Hence, our approach will be called *possible causes approach* (PCA), in contrast to Ginsberg's possible worlds approach (PWA) [9] and Winslett's possible models approach (PMA) [14]. Conceptually, the possible causes approach is different from all the above mentioned approaches. In the AGM theory, for example, if $K \cup N$ is consistent, then the new belief set will simply be $Cn(K \cup N)$. In our approach, however, if a chosen cause C of N has another effect, say X, then the agent should also add X to her knowledge base. The PCA approach is especially suitable to domains of actions, where there is a clear cause-effect relation. In domains of actions, new knowledge is about changes in truth values of fluents. The possible causes for changes in truth values of fluents are the occurrences of actions.

As an example, suppose that Tom has Book and Mary has Money on August 1, 1995. Assume that Mary is found to have Book on August 2, 1995. Then, there are several possible causes for the new knowledge: Mary buys it with her Money, Tom gives it to Mary, or even Mary steals it from Tom. For example, if the possible cause that Mary buys Book is admitted, then there should be another fact in the new knowledge base: Tom has Money.

The rest of this paper is organized as follows. First we review the action description language $\mathcal{A}$ of [8] for domain descriptions, then present a new transformation π from domain descriptions in $\mathcal{A}$ to abductive logic programs, which is different from other translations of, e.g.,[8, 6] in that a time dimension is incorporated so that one can represent and reason about narratives, next we use the translation π to represent temporal knowledge bases and give the soundness and completeness theorems for the representation, then we discuss how to incorporate new knowledge into a knowledge base by using the possible causes approach and compare PCA with PWA and PMA. All proofs are omitted but can be found in the long version of this paper.

Domain Description

Syntax

We begin with two disjoint non-empty sets of symbols, called *fluent names* and *action names*. An *action expression*, or simply *action*, is defined to be an action name. A *fluent expression*, or simply *fluent*, is defined to be a fluent name possibly preceded by $\neg$. A fluent expression is also called a *positive fluent* if it only consists of a fluent name; otherwise it is called a *negative fluent*.

In $\mathcal{A}$ there are two kinds of propositions: value propositions and effect propositions, simply called v-propositions, e-propositions. In this paper we only need a special class of v-propositions of the form

$$\textbf{initially } F \qquad (1)$$

where F is a fluent. An e-proposition is defined to be a statement of the form

$$A \textbf{ causes } F \textbf{ if } P_1, \ldots, P_n \qquad (2)$$

where A is an action expression, and each of F, P_1, ..., P_n $(n \geq 0)$ is a fluent expression. If $n = 0$, then we will write it as A **causes** F.

A *domain description* D is a set of v-propositions and e-propositions. For example, the domain description D_{scp} for the Stolen Car Problem [12] is as follows: $D_{scp} = \{$ **initially** $\neg Stolen$. $Steal$ causes $Stolen\}$.

Semantics

The semantics of $\mathcal{A}$ is defined by using states and transitions. A *state* is a set of fluent names. Given a fluent name F and a state σ, we say that F holds in σ if $F \in \sigma$; $\neg F$ holds in σ if $F \notin \sigma$. A *transition function* Φ is a mapping from the set of pairs (A, σ), where A is an action expression and σ is a state, to the set of states.

An *interpretation structure* is a pair (σ_0, Φ), where σ_0 is a state, called the *initial state* of (σ_0, Φ), and Φ is a transition function. For any interpretation structure $M = (\sigma_0, \Phi)$ and any sequence of action expressions $A_1; \ldots; A_m$ in M, by $\Phi(A_1; \ldots; A_m, \sigma_0)$ we denote the state $\Phi(A_m, \Phi(A_{m-1}, \ldots, \Phi(A_1, \sigma_0) \ldots))$.

A v-proposition of the form **initially** F is true in M iff F holds in the initial state σ_0. An interpretation structure (σ_0, Φ) is a *model* of a domain description D iff every v-proposition of D is true in (σ_0, Φ), and for every action expression A, every fluent name F and every state σ, the following conditions are satisfied: (i) If D includes an e-proposition A **causes** F **if** $P_1, \ldots, P_n$, whose preconditions P_1, ..., P_n hold in σ, then $F \in \Phi(A, \sigma)$; (ii) If D includes an e-proposition A **causes** $\neg F$ **if** P_1, ..., P_n, whose preconditions P_1, ..., P_n hold in σ, then $F \notin \Phi(A, \sigma)$; (iii) If D does not include any such e-propositions, then $F \in \Phi(A, \sigma)$ iff $F \in \sigma$. A domain description is *consistent* if it has a model. A domain description is *complete* if it has exactly one model.

Translation into logic programs

In the following we present a new translation π from domain descriptions in $\mathcal{A}$ to abductive logic programs, where a time dimension is incorporated so that one can represent and reason about narratives, actions and time. Let D be a domain description, the translation πD consists of the following logic programming rules:

1. Time dimension:

   In this paper we assume that time is structured on points, represented by a totally linearly-ordered set $(TP, \prec, succ, init)$, where TP is an infinite set of time points, $init \in TP$, and $succ(T) \in TP$ for any $T \in TP$. We will often use natural numbers which have a straightforward correspondence to terms for time points: $init$ corresponding to 0, $succ(init)$ to 1, etc.

2. Initialization: $holds(F, init) \leftarrow initially(F)$.

3. Law of Inertia:

   $$holds(F, succ(T)) \leftarrow holds(F, T), \; not\ noninertial(F, T).$$

   where *not* is the negation-as-failure operator. By the law of inertia, a fluent F is true at a time point T if it was true earlier and inertial then.

4. Each e-proposition a **causes** f **if** $p_1, \ldots, p_n$, with f being positive, is translated into

   $$holds(f, succ(T)) \leftarrow happens(a, T), \; holds(p_1, T), \ldots, holds(p_n, T).$$

5. Each e-proposition a **causes** $\neg f$ **if** $p_1, \ldots, p_n$, with f being positive, is translated into

   $$noninertial(f, T) \leftarrow happens(a, T), \; holds(p_1, T), \ldots, holds(p_n, T).$$

6. Each v-proposition **initially** f or **initially** $\neg g$, where f and g are positive, is respectively translated into

   $$\begin{array}{lcl} false & \leftarrow & not\ holds(f, init). \\ false & \leftarrow & holds(g, init). \end{array}$$

   The above two logic programming rules function as integrity constraints.

Representation of Knowledge Bases

In this section, we we will use the above translation to represent temporal knowledge bases about domains of actions.

Temporal knowledge bases

Given a domain description D, we want to determine whether a fluent holds at a particular time point after some actions have happened in the past. We start with the concept of histories, then employ abductive logic programming for the representation of temporal knowledge.

A domain history H is a finite set of pairs (A,T) for action A and time T. By $(A,T) \in H$ we mean that A happens at T. In this paper we do not consider concurrent actions. Thus we assume that for any $T \in TP$ there is at most one action A such that $(A,T) \in H$.

A domain evolution E is a pair (D,H), where D is a domain description and H a history. Given a domain evolution (D,H), we can have an abductive logic program $KB(D,H)$ defined as follows:

$$KB(D,H) = \pi D \cup \{false \leftarrow happens(a,t) : (a,t) \in H\}$$

The logic program $KB(D,H)$ will be simply called *the knowledge base generated from D and H.* The literal $false$ is always interpreted as logical falsity, and all rules with $false$ as heads function as integrity constraints. The predicates $initially(F)$ is taken as an abducible predicate used to capture the incomplete knowledge about the initial situation/time. The semantics of $KB(D,H)$ is defined to be the union of the integrity constraints, the Clark Equality Theory, and the first-order theory obtained by completing all the non-abducible predicates (all predicates except $initially(F)$). We will simply write $COMP(KB(D,H))$ to denote the semantics of of $KB(D,H)$, and often write $KB(D,H) \models Q$ to stand for $COMP(KB(D,H)) \models Q$. The following two results justify the above semantics definition.

Proposition 1 *$KB(D,H)$ is an acyclic logic program.*

Corollary 2 *The semantics $COMP(KB(D,H))$ of the abductive logic program $KB(D,H)$ coincides with its generalized stable model semantics and generalized well-founded model semantics.*

Soundness and completeness

Given a history H, we can have a unique sequence of actions $A_1; A_2; \ldots; A_m$ and a unique sequence of time points $T_1 \prec T_2 \prec \ldots \prec T_m$ such that $H = \{(A_i,T_i) : 1 \leq i \leq m\}$. The sequence of actions $A_1; A_2; \ldots; A_m$ will be called *the trajectory generated from H, denoted by $Traj(H)$.* A pre-history of H is defined to be a subset H' of H such that $Traj(H')$ is a prefix of $Traj(H)$.

Given an interpretation structure $M = (\sigma_0, \Phi)$, a history H and a time point T, we can derive a state S as follows: if $H' = \{(A_1,T_1), \ldots, (A_k,T_k)\}$ is a pre-history of H such that $T_k \prec T$ and there is no time point Tx and action Ax with $T_k \prec Tx \prec T$ and $(Ax,Tx) \in H$, then $S = \Phi(A_1; A_2; \ldots; A_k, \sigma_0)$, called *the state generated from H and T in M*, denoted by $GS_M(H,T)$. If T is a time point such that $(A,T) \notin H$ for any action A, and $(A,Tx) \notin H$ for any action A and any time point Tx with $T \prec Tx$, then the state $GS_M(H,T)$ is called *the state generated from H in M*, denoted by $GS_M(H)$. For example, let $M = (\sigma_0, \Phi)$ be an interpretation structure, and $H = \{(a,1),(a,6),(b,4),(c,10)\}$ a history. Then, $GS_M(H,5) = \Phi(a;b,\sigma_0)$ and $GS_M(H) = \Phi(a;b;a;c,\sigma_0)$.

Proposition 3 *Given an interpretation structure M, a history H, and a time point T, the state $GS_M(H,T)$ generated from H and T in M is unique.*

Note that $\mathcal{A}$ has an infinite tree-like time structure. Given a history H, the translation π only takes into account a special path in the tree corresponding to H. In order for π to be correct, we require that π be correct with respect to any history.

Definition 4 *Let D be a domain description, and H a history. The logic program $KB(D,H)$ is said to be sound with respect to H iff, for every model M of D, fluent F and time T, if $KB(D,H) \models holds(F,T)$, then $F \in GS_M(H,T)$. The translation π is said to be sound iff, for every history H, $KB(D,H)$ is sound with respect to H.*

Definition 5 *Let D be a domain description, and H a history. The logic program $KB(D,H)$ is said to be complete with respect to H iff, for every fluent F and time T, if $F \in GS_M(H,T)$ for every model M of D, then $KB(D,H) \models holds(F,T)$. The translation π is said to be complete iff, for every history H, $KB(D,H)$ is complete with respect to H.*

Theorem 6 (Soundness) *Assume that D is a domain description. For any history H, model I of D, fluent F, and time T, if $KB(D,H) \models holds(F,T)$, then $F \in GS_I(H,T)$.*

A domain description D is *e-consistent* [6] iff for each pair of e-propositions a **causes** f **if** $p_1, \ldots, p_n$ and a **causes** $\neg f$ **if** $p_{n+1}, \ldots, p_m$ in D, there exist i and j such that p_i is the complement of p_j.

Theorem 7 (Completeness) *Assume that D is an e-consistent domain description. For any history H, fluent F, and time T, if $F \in GS_I(H,T)$ for any model I of D, then $KB(D,H) \models holds(F,T)$.*

Incorporation of New Knowledge

For easy reference, we will simply write $KB(D,H) \oplus N$ to informally denote all the possible new knowledge bases obtained by incorporating N into $KB(D,H)$ before we formally define it later. For convenience, we often write $N = L$ if $N = \{L\}$.

Possible causes approach(PCA)

Suppose that $N = holds(F,T)$ is a newly observed fact. In the process of incorporating N into $KB(D,H)$, first the agent should check whether N is already stored in her knowledge base $KB(D,H)$, i.e., whether $KB(D,H) \models N$. If it is so, the agent will do nothing. In this case, $KB(D,H) \oplus N \equiv KB(D,H)$. Suppose that N is not stored in the knowledge base, that is, $KB(D,H) \not\models N$. In this case, either the fluent F was true and inertial at the last time point or an action has happened at the last time point and a initiates F. After a careful analysis, it can be seen that the v-propositions **initially** F and occurrences of actions which initiate N or recursively initiate preconditions of actions which initiate N are possible causes for N in $KB(D,H)$.

Let (D,H) be a domain evolution, and Δ the set of instances of $initially(F)$ and $happens(A,T)$. For any finite subset δ of Δ we will write $KB(D,H) + \delta$ to stand for $KB(D',H')$, where $D' = D \cup \{$ **initially** F : $initially(F) \in \delta\}$ and $H' = H \cup \{(A,T) : happens(A,T) \in \delta\}$. We assume that no concurrent actions appear in H'. We will often omit $initially(F)$ from δ when both **initially** $F \in D$ and $initially(F) \in \delta$.

Definition 8 *Let D be a domain description, H a history, and N a set of literals. A possible cause for N in $KB(D,H)$ is a subset δ of Δ such that (i) $KB(D,H) \dot{+} \delta \models N$; (ii) $KB(D,H) \dot{+} \delta$ is consistent; (iii) For any time T, there are no different actions A and B such that $(A,T),(B,T) \in H \cup \{(C,T) : happens(C,T) \in \delta\}$; (iv) δ is minimal in the sense that no subset δ' of δ exists such that δ' satisfies the first three conditions. We will write $KB(D,H) \uparrow N$ to stand for all the possible causes for N in $KB(D,H)$.*

As an example, consider the Stolen Car Problem: $D_{scp} = \{$ **initially** $\neg Stolen$. *Steal* **causes** *Stolen*.$\}$ and $H_{scp} = \emptyset$. Suppose that the car is missing from the parking lot at time 3. Then there are three possible causes for $holds(stolen,3)$ in $KB(D_{scp},H_{scp})$:

$$\begin{aligned} \delta_1 &= \{happens(steal,0)\}, \\ \delta_2 &= \{happens(steal,1)\}, \\ \delta_3 &= \{happens(steal,2)\} \end{aligned}$$

In order to incorporate N into $KB(D,H)$, if there is a possible cause δ for N in $KB(D,H)$, then we can simply add δ into the knowledge base to have a new knowledge base. Now we have to question about whether there is always a possible cause δ for N in $KB(D,H)$. The answer turns out to be negative.

Proposition 9 *Let D be a domain description, and H a history. Let N be a set of literals. It does not hold that there is always a possible cause for N in $KB(D,H)$.*

For incorporation of N into $KB(D,H)$, when there is no possible cause for N in $KB(D,H)$, we will simply discard N because N cannot be explained.

Definition 10 *Given a knowledge base $KB(D,H)$, and a set of new facts N, all the possible new knowledge bases $KB(D,H) \oplus N$ are defined to be $\{KB(D,H) \dot{+} \delta : \delta \in KB(D,H) \uparrow N\}$. The operator $\oplus$ is called the PCA knowledge incorporation operator.*

For example, $KB(D_{scp},H_{scp}) \oplus holds(stolen,3)$ includes $KB(D_{scp},\{(steal,0)\})$, $KB(D_{scp},\{(steal,1)\})$, and $KB(D_{scp},\{(steal,2)\})$ in the Stolen Car Problem.

Note that the new knowledge N is not directly added to the knowledge base. This can be problematic in general. As discussed in its long version, there is a simple way to modify the above definition so that both new knowledge and its possible causes are incorporated, described as follows. For each fact $holds(F,T)$ and $\neg holds(G,T)$ in N, we add the following two constraints, respectively:

$$\begin{aligned} false &\leftarrow not\ holds(F,T) \\ false &\leftarrow holds(G,T) \end{aligned}$$

Then, N is kept permanently in the knowledge base.

The PCA knowledge incorporation operator can also be used to evaluate counterfactual queries: if F_1 were true at time T_1, would F_2 be true at time T_2? This amounts to evaluate the query "$? - holds(F_2,T_2)$" in $KB(D,H) \oplus holds(F_1,T_1)$. The details are omitted.

Computational considerations

The critical step in the PCA approach is the computation of the possible causes. In this section we show how to compute them with abductive logic programming and how to improve computational efficiency.

First we introduce a new predicate $occurs(A,T)$, for action A and time T, and add a new programming rule to $KB(D,H)$:

$$happens(A,T) \leftarrow occurs(A,T) \tag{3}$$

Now we define the abducibles to be $initially(F)$ and $occurs(A,T)$. The above rule means that if action A is abduced to occur at time T, it happens at T. As said before, this paper does not consider concurrent actions. Thus we add the following rule:

$$false \leftarrow A_1 \neq A_2, happens(A_1,T), happens(A_2,T) \tag{4}$$

The definitions for $A1 \neq A2$ should also be added such that every two syntactically different actions are not equal. However, later we will find that the above rule is redundant and can be removed for efficiency improvement. Let $KB_{occurs}(D,H) = KB(D,H) \cup \{(3),(4)\}$. It is easy to see that KB_{occurs} is still acyclic. It can also be shown that possible causes for N become abductive answers to N in $KB_{occurs}(D,H)$.

Proposition 11 *$\delta \in KB(D,H) \uparrow N$ iff $\{initially(F) : initially(F) \in \delta\} \cup \{occurs(A,T) : happens(A,T) \in \delta\}$ is an abductive answer to N in the abductive logic program $KB_{occurs}(D,H)$.*

In what follows we develop some techniques to improve the search efficiency.

Proposition 12 *Let δ be any abductive answer to $holds(F_n, T_n)$ in the program $KB_{occurs}(D, H)$. Then, for any $occurs(A, T) \in \delta$ we have $T \prec T_n$.*

Thus it is not necessary to consider predicate instances $occurs(A, T)$ for $T \not\prec T_n$. The second observation is that we have assumed that no concurrent actions have happened in histories. Thus, if $(A, T) \in H$, we need not consider $occurs(B, T)$ for any other action B.

Proposition 13 *Let δ be any abductive answer to N in $KB_{occurs}(D, H)$. If $occurs(a, t) \in \delta$, then there is no other action b such that $(b, t) \in H$.*

The third observation is about the consistency check. By Definition 8, $KB(D, H) \dotplus \delta$ needs to be consistent for δ to be a possible cause. To check the consistency of a knowledge base is very expensive. In the following we give a sufficient and cheap technique to check the consistency.

Proposition 14 *Let D be an e-consistent domain description. If D is not consistent, then there are two value propositions* **initially** *F and* **initially** *G such that $F = \neg G$ or $G = \neg F$.*

By the above proposition, the consistency check can be done in the following way: For each fluent F, we add a new programming rule to $KB(D, H)$:

$$false \leftarrow holds(F, init), not\ holds(F, init)$$

All the examples of this paper and a few more benchmark examples have been experimented with the latest version of the REVISE system [5].

Related Work

There have been many proposals for belief revision and update. Katsuno and Mendelzon [11] made a distinction between the belief revision and belief update. In the sense of [11], our PCA operator seems to be better called a belief update operator. However, when the complete history is unknown but new facts are observed out of the incompleteness of the history, the PCA operator seems to be called a belief revision operator as the world does not change. In what follows we will use an example to compare our possible causes approach PCA with the possible worlds approach PWA [9]) and the possible models approach PMA [14], since both PWA and PMA have also been used in domains of actions[1].

Now consider a domain where there are two people: *Tom* and *Mary* , and two objects: *Money* and *Book*. In the beginning, say November 1, 1995, *Tom* has *Book* while *Mary* has *Money*. Assume that either of the two people can buy *Book* from the other if he/she has *Money* and the other has *Book*; either can give *Book* or *Money* to the other; either can steal an object from the other. Suppose that *Mary* is found to have *Book* later, say November 5, 1995. We will use this example to compare our PCA with PWA and PMA.

The PWA approach was first proposed by Ginsberg to evaluate counterfactuals [9] and later used for, e.g. reasoning about actions by Ginsberg and Smith. Let $*_G$ denote the PWA operator defined as follows:

$$S *_G p = \{T \cup \{p\} : T \in W_G(p, S)\}$$

where $W_G(p, S) = \{T \subseteq S : T \not\models \neg p$ and $T \subset U \subseteq S \Rightarrow U \models \neg p\}$.

In the spirit of [9], the time parameter is not directly considered and temporal knowledge is represented by a set of fluent names which are known to be true or false. For the *Tom–Mary* example, the temporal knowledge base is defined to be $KB_1 = \{has(Tom, Book), \neg has(Tom, Money), \neg has(Mary, Book), has(Mary, Money)\}$. And the new knowledge is defined to be $N_1 = has(Mary, Book)$. Then we can prove that $KB_1 *_G N_1 = \{\{has(Tom, Book), \neg has(Tom, Money), has(Mary, Book), has(Mary, Money)\}\}$. Since it cannot be the case for both *Tom* and *Mary* to have the same *Book*, one may want to add the following integrity constraints:

$$\begin{aligned} has(X, Book) \wedge has(Y, Book) &\rightarrow X = Y \ (5) \\ has(X, Money) \wedge has(Y, Money) &\rightarrow X = Y \ (6) \end{aligned}$$

Even if these integrity constraints are protected, and the unique names axioms are adopted, the temporal knowledge base KB_3 defined below is derivable from the PWA approach, that is the best the PWA approach can derive. $KB_3 = \{\neg has(Tom, Book), \neg has(Tom, Money), has(Mary, Book), has(Mary, Money)\}$. However, by the PCA approach, there is also another possible new knowledge base KB_4, which cannot be derived by PWA: $KB_4 = \{\neg has(Tom, Book), has(Tom, Money), has(Mary, Book), \neg has(Mary, Money)\}$. Intuitively, KB_3 corresponds to the possible cause: *Tom* gives *Book* to *Mary* or *Mary* steals *Book* from *Tom*, while KB_4 corresponds to the possible cause: *Mary* buys *Book* from *Tom* with Mary's *Money*. In our PCA approach, the counterparts in the real time line of both KB_3 and KB_4 can be derived.

Now let's see Winslett's possible models approach PMA [14]. We denote Winslett's belief update operator with $*_W$. Let K be a knowledge base and N a new sentence. For each model I of K, from the models of N Winslett selects those closest to I by set inclusion. Formally, Winslett says that an interpretation J_1 is closer to I than J_2, denoted by $J_1 \preceq_I J_2$, iff $diff(J_1, I) \subset diff(J_2, I)$, where $diff(X, Y)$ is defined as follows: $diff(X, Y) = (X \setminus Y) \cup (Y \setminus X)$. Intuitively, $diff(X, Y)$ includes the atoms which have different truth values in X and Y, respectively. Let $Incorporate(N, I)$ be the set of all the minimal ele-

[1] An anonymous referee pointed out that Boutilier [3] and Friedman and Halpern [7] are very relevant to our work.

ments of $mod(N \wedge IC)$ with respect to $\preceq_I$, that is,

$$Incorporate(N,I) = \{J \in mod(N \wedge IC) : \text{There is no } J' \in mod(N \wedge IC) \text{ such that} J' \preceq_I J\}$$

where IC is the set of protected formulas as integrity constraints, which are supposed to be satisfied in every possible state of beliefs. Then, the models $mod(K *_W N)$ of the new belief base are defined as follows:

$$mod(K *_W N) = \bigcup_{I \in mod(K \wedge IC)} Incorporate(N,I)$$

In the style of [14], the knowledge bases are represented by fluent names which are known to be true or false. Now let's consider the *Tom–Mary* example by computing $KB_1 *_W N_1$. We require the formulas (5) and (6) be protected. Then, we can prove: $KB_1 *_W N_1 = \{\neg has(Tom, Book), \neg has(Tom, Money), has(Mary, Book), has(Mary, Money)\}$. As discussed before, although KB_3 is an acceptable new possible knowledge base which corresponds to the possible cause: *Tom* gives *Book* to *Mary* or *Mary* steals *Book* from *Tom*, we cannot obtain another equally acceptable new possible knowledge base KB_4 which corresponds to another possible cause: *Mary* buys *Book* from *Tom*.

In summary, by the PCA approach we can find all possible desirable new temproal knowledge bases whereas by the PWA and PMA we can only find some of possible desirable new temproal knowledge bases.

Conclusion

We have developed a formal, provably correct and yet computational methodology for incorporation of new knowledge into knowledge bases about actions and changes. We started with a simple action description language $\mathcal{A}$, used it to describe domains of actions, presented a new translation from domain descriptions in $\mathcal{A}$ to abductive normal logic programs where a time dimension is incorporated, defined knowledge bases about domains of actions as abductive logic programs, showed the soundness and completeness of the knowledge bases with respect to their domain descriptions, and in particular proposed a possible causes approach PCA to belief update. A possible cause of new knowledge consists of some abduced occurrences of actions and value propositions about the initial state of the domain of actions. To incorporate new knowledge into a knowledge base, all the effects of the actions which possibly initiate the new knowledge, recursively together with all the effects of actions which possibly initiate the preconditions of the actions which possibly initiate the new knowledge, are also incorporated. In our PCA approach, the possible causes play a central role. We have shown how to compute possible causes with abductive logic programming, and presented some techniques to improve the search efficiency. We have employed an example to show the advantages of the PCA approach over the PWA and PMA approaches. All the examples of this paper and a few more benchmark examples have been experimented with the latest version of the REVISE system [5].

Acknowledgements

This work was supported by a post-doctoral fellowship from JNICT under PRAXIS XXI/BPD/4165/94 to the first author and a JNICT PROLOPPE project under PRAXIS/3/31/TIT/24/94.

References

[1] Alchorrón, C., Gärdenfors, P. and Makinson, D., On the logic of theory change: Partial meet contraction and revision functions, *J. of Symbolic Logic*, 50:2, 1985, 510 - 530

[2] Apt, K.R. and Bezem, M., Acyclic programs, *Proc. of ICLP 90*, MIT Press, 579–597

[3] Boutilier, C., Generalized update: belief change in dynamic settings, Proc. of IJCAI'95, Montreal, 1995, 1550–1556

[4] Dalal, M., Investigations into a theory of knowledge base revision: Preliminary report, *Proc. of AAAI 88*, 475 - 479

[5] Damásio, C., Pereira, L. M., and Nejdl, W., REVISE: An Extended Logic Programming System for Revising Knowledge Bases, *Proc. of KR'94*, 1994

[6] Denecker, M. and de Schreye, D., Representing incomplete knowledge in abductive logic programming, *Logic Programming: Proc. of the 1993 Int'l Symposium*, 1993, 147–163

[7] Friedman, N. and Halpern, J., A knowledge-based framework for belief change, II: Revision and update, Proc. of KR'94, Bonn, 1994, 190–201

[8] Gelfond, M. and Lifschitz, V., Representing action and change by logic programs, *Journal of Logic Programming*, Vol.17, 1993, 301–322

[9] Ginsberg, M., Counterfactuals, *Artificial Intelligence*, 30:1, 1986, 35 - 79

[10] Kakas, A.C., Kowalski, R.A., Toni, F., Abductive logic programming, *J. of Logic and Computation*, 2;6, 1993, 719-770

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

[12] Kautz, H.A., The logic of persistence, *Proc. of the AAAI86*, 1986, 401-405

[13] Nebel, B., Belief revision and default reasoning: Syntax-based approaches, *Proc. of KR'91*, 1991

[14] Winslett, M., Reasoning about action using a possible models approach, *Proc. of AAAI'88*, 89 - 93