An Action Description Language for Iterated Belief Change

Aaron Hunter and James P. Delgrande
Simon Fraser University
Burnaby, BC, Canada
{amhunter, jim}@cs.sfu.ca

Abstract

We are interested in the belief change that occurs due to a sequence of ontic actions and epistemic actions. In order to represent such problems, we extend an existing epistemic action language to allow erroneous initial beliefs. We define a non-Markovian semantics for our action language that explicitly respects the interaction between ontic actions and epistemic actions. Further, we illustrate how to solve epistemic projection problems in our new language by translating action descriptions into extended logic programs. We conclude with some remarks about a prototype implementation of our work.

1 Introduction

Reasoning about the effects of actions is an important problem in logical AI. Action formalisms are often defined for reasoning about so-called *ontic actions* that change the state of the world. However, *sensing actions* have also been incorporated in the epistemic extensions of several notable formalisms [Shapiro *et al.*, 2000; Lobo *et al.*, 2001; Son and Baral, 2001; Jin and Thielscher, 2004]. In such extensions, the effects of sensing actions are defined in terms of *belief revision*. However, simply supplementing an action formalism with a revision operator is not sufficient; the iterated belief change caused by a sequence of ontic actions and sensing actions can not be determined iteratively [Hunter and Delgrande, 2005]. Informally, the interpretation of a sensing result may be influenced by the preceding sequence of ontic actions. In this paper, we define an action formalism where the belief change caused by a sequence of actions respects the non-elementary interaction between ontic actions and sensing actions.

In order to ground the discussion, we frame our results in an epistemic extension of the action language $\mathcal{A}$ [Gelfond and Lifschitz, 1993]. We base the semantics of our action language on the belief evolution operators of [Hunter and Delgrande, 2005]; our results can be seen as an application of the belief evolution methodology in an action formalism. The two main contributions of this paper are as follows. First, we introduce an action formalism that is suitable for reasoning about iterated belief change where the interpretation of sensing results depends on the preceding actions. In the process, we generalize an existing epistemic extension of $\mathcal{A}$ by allowing erroneous beliefs and non-Markovian belief change. The second contribution of this paper is the introduction of a method for solving belief evolution problems through answer set planning. Using this method, we can implement a solver for epistemic projection problems in our action language.

2 Motivating Example

We present a commonsense example involving iterated belief change due to action, and we will return to this example periodically throughout the paper. The example that we present is framed in the context of a zoo, where a certain crocodile must be fed every morning. The crocodile is fed from a bag of food that either contains whole chickens or whole ducks; the contents of the food bag varies throughout the year. The crocodile is never sick after eating chicken; even if it is initially sick we suppose that eating chicken makes it feel better. However, the crocodile will become sick if it eats two ducks in row. The crocodile will always eat the food it is given. The first zoo keeper to arrive in the morning typically feeds the crocodile by giving it one unit of food.

Suppose that Bob the zoo keeper arrives in the morning for work. Bob believes that the crocodile is unfed when he arrives, and he believes that the food bag contains chickens. Suppose that Bob feeds the crocodile, then observes that it becomes sick. We suggest that Bob should conclude that his initial beliefs were incorrect; the sickness of the crocodile indicates that it has actually eaten two ducks. We are interested in using an action description language to formally model the belief change that occurs in problems of this form.

3 Preliminaries

3.1 Action Language $\mathcal{A}$

We briefly review the syntax and semantics of the action language $\mathcal{A}$, as introduced in [Gelfond and Lifschitz, 1993].

An *action signature* is a pair $\langle \mathbf{F}, \mathbf{A} \rangle$ where $\mathbf{F}$ denotes a fixed set of fluent symbols and $\mathbf{A}$ denotes a fixed set of action symbols. A *state* is a propositional interpretation over $\mathbf{F}$, S denotes the set of all states, and $|\phi|$ denotes the set of states satisfying the formula ϕ. A *literal* is either an element of $\mathbf{F}$ or an element of $\mathbf{F}$ prefixed with the negation symbol. Let $Lits$ denote the set of all literals. We use the upper case letter A

to range over actions, the lower case letters f, g to range over literals, and the lower case letter s to range over states.

Definition 1 *An effect proposition of the language $\mathcal{A}$ is an expression of the form*

$$A \textbf{ causes } f \textbf{ if } g_1 \wedge \cdots \wedge g_p$$

where $A \in \mathbf{A}$, $f \in Lits$ and each $g_i \in Lits$.

A set of effect propositions is called an *action description*. Every action description AD defines a transition relation on states as indicated in the following definition.

Definition 2 *Let AD be an action description, let s, s' be states and let A be an action symbol. Then $\Phi_{AD}(s, A, s')$ if*

$$E(A, s) \subseteq s' \subseteq E(A, s) \cup s$$

*where $E(A, s)$ is the set of literals such that $f \in E(A, s)$ if and only if $(A$ **causes** f **if** $g_1 \wedge \cdots \wedge g_p) \in AD$ and $s \models g_1 \wedge \cdots \wedge g_p$.*

Intuitively, the transition relation maps a pair (s, A) to a new interpretation s' that is exactly like s except for the values of the fluents affected by A.

3.2 Action Language $\mathcal{A}_K$

Our epistemic extension of $\mathcal{A}$ will be based on the language $\mathcal{A}_K$ of [Lobo *et al.*, 2001]. In this section, we briefly summarize the action description portion of $\mathcal{A}_K$. We remark that the complete specification of $\mathcal{A}_K$ also includes queries, plans, and non-deterministic action effects. We restrict attention to the portion of the language that is defined in this section.

A *belief state* is a set of states. The syntax of $\mathcal{A}_K$ is obtained by extending $\mathcal{A}$ with sensing actions, with effects given by propositions of the form

$$A \textbf{ causes to know } f \textbf{ if} g_1 \wedge \cdots \wedge g_p. \qquad (1)$$

The semantics of $\mathcal{A}_K$ associates an *epistemic transition relation* Φ^L_{AD} with every action description AD. An epistemic transition relation is a set of triples $\langle \kappa, A, \kappa^* \rangle$ where A is an action symbol and κ, κ^* are belief states. The relation Φ^L_{AD} is defined as follows. If A is a non-sensing action, then $(\kappa, A, \kappa^*) \in \Phi^L_{AD}$ if and only if κ^* is obtained by updating each world in κ in accordance with the semantics of $\mathcal{A}$. If A is a sensing action described by (1), then $(\kappa, A, \kappa^*) \in \Phi^L_{AD}$ just in case one of the following conditions holds.

1. κ^* is the subset of κ where $g_1 \wedge \cdots \wedge g_p$ and f hold
2. κ^* is the subset of κ where $g_1 \wedge \cdots \wedge g_p$ and $\neg f$ hold
3. κ^* is the subset of κ where $\neg(g_1 \wedge \cdots \wedge g_p)$ holds

Informally, $(\kappa, A, \kappa^*) \in \Phi^K_{AD}$ means that κ^* is a possible belief state after executing the action A with belief state κ.

3.3 Feeding the Crocodile in $\mathcal{A}_K$

We illustrate how to represent the crocodile problem. The crocodile problem can be described in terms of the actions $Feed$ and $LookAtCroc$, along with the fluents $Chicken, FullChicken, FullDuck$ and $Sick$. Informally, $Chicken$ is true if the food bag contains chickens, whereas $FullChicken$ and $FullDuck$ indicate what the crocodile has eaten. Action effects are described as follows.

$$Feed \textbf{ causes } FullChicken \textbf{ if } Chicken$$
$$Feed \textbf{ causes } \neg Sick \textbf{ if } Chicken$$
$$Feed \textbf{ causes } FullDuck \textbf{ if } \neg Chicken$$
$$Feed \textbf{ causes } Sick \textbf{ if } FullDuck \wedge \neg Chicken$$
$$LookAtCroc \textbf{ causes to know } Sick \textbf{ if} Sick.$$

Bob's initial belief state is $\kappa = |\neg FullChicken \wedge \neg FullDuck \wedge Chicken|$. We are interested in Bob's new beliefs after performing the actions $Feed$ and $LookAtCroc$.

After feeding the crocodile, Bob's new belief state κ' is a subset of $|\neg Sick|$. As a result, after looking at the crocodile, the semantics of $\mathcal{A}_K$ defines Bob's final belief state to be $\emptyset$. Hence, performing revision without considering the action history leads Bob to hold a vacuous set of beliefs, despite the fact that there are plausible world histories that support Bob's observation. This problem can be avoided by introducing an appropriate belief change operator.

3.4 Belief Evolution

One way to address erroneous beliefs in $\mathcal{A}_K$ would be to introduce an AGM revision operator $*$ [Alchourrón *et al.*, 1985], and then define Bob's new beliefs to be $\kappa' * Sick$. However, under this approach, it is possible that Bob's final belief state will contain states satisfying $Sick \wedge Chicken$. This is the case, for example, if $*$ is the Dalal operator [Dalal, 1988]. We suggest that such states should not be possible, because Bob is aware that sickness never follows eating chicken. Informally, Bob's observation after feeding should cause him to revise his initial belief state. *Belief evolution* operators have been proposed to model this kind of reasoning [Hunter and Delgrande, 2005]. We briefly present a simplified version of belief evolution.

Let Φ be a transition relation as in Definition 2. We associate a belief projection operator $\diamond$ with Φ as follows. For any belief state κ and action A,

$$\kappa \diamond A = \{s' \mid (s, A, s') \in \Phi \text{ for some } s \in \kappa\}.$$

Now suppose that $*$ is an AGM revision operator. We will actually let $*$ take a set of states as an argument rather than a formula, but it is clear that AGM revision can equivalently be formulated in this manner. We define the belief evolution operator $\circ$ associated with $\diamond$ and $*$ presently. Let $\bar{A}$ denote a finite sequence of non-sensing actions, and let ϕ denote a propositional formula. Define $\phi^{-1}(\bar{A})$ to be the set all states s such that $\bar{A}$ gives a path from s to a state where ϕ is true. Define $\circ$ as follows:

$$\kappa \circ \langle \bar{A}, \phi \rangle = \kappa * \phi^{-1}(\bar{A}) \diamond \bar{A}.$$

Hence, belief evolution operators essentially revise the initial belief state before applying the effects of non-sensing actions. In [Hunter and Delgrande, 2005], belief evolution operators are defined for arbitrary sequences of sensing and non-sensing actions. To simplify the discussion in the present paper, however, we restrict attention to sequences involving a single, terminal sensing action. It would be straightforward to extend our results to allow arbitrary action sequences by using the full definition of belief evolution.

4 An Action Language for Belief Change

4.1 Syntax

Our language is obtained by making a slight modification to $\mathcal{A}_K$. Let $\mathbf{A} = \mathbf{O} \cup \mathbf{N}$ where $\mathbf{O} \cap \mathbf{N} = \emptyset$. We refer to $\mathbf{O}$ as the set of *sensing actions* and we refer to $\mathbf{N}$ as the set of *non-sensing actions*. The symbol O ranges over sensing actions and the symbol A ranges over non-sensing actions.

Definition 3 *Propositions of $\mathcal{A}_B$ have the following forms:*

1. A **causes** f **if** $g_1 \wedge \cdots \wedge g_p$
2. O **causes to believe** ϕ **if** $g_1 \wedge \cdots \wedge g_p$

where $A \in \mathbf{N}$, $O \in \mathbf{O}$, each $g_i \in Lits$, and ϕ is a formula.

Note that the effect of a sensing action is now a formula rather than a fluent symbol.

4.2 Semantics

The semantics of $\mathcal{A}_B$ is defined with respect to *pointed belief states* and *epistemic action sequences*. A pointed belief state is a pair $\langle s, \kappa \rangle$ where $s \in S$ and $\emptyset \neq \kappa \subseteq S$. The state s represents the actual state of the world and κ represents the set of states believed to be possible. A *pointed knowledge state* is a pointed belief state $\langle s, \kappa \rangle$ where $s \in \kappa$. An epistemic action sequence is a sequence $A_1, \ldots, A_n, O$ where each $A_i \in \mathbf{N}$ and $O \in \mathbf{O}$. With each action description AD, we associate an epistemic transition relation Φ_{AD}. For easy of readability, we write Φ_{AD} as a function that takes a pointed belief state and an epistemic action sequence as arguments, and it returns a new pointed belief state.

Let $O \in \mathbf{O}$ and let $s \in S$. Define $EFF(O, s)$ to be the conjunction of every formula ϕ that occurs in a proposition of the form

$$O \textbf{ causes to believe } \phi \textbf{ if } g_1 \wedge \cdots \wedge g_p$$

where $s \models g_1 \wedge \cdots \wedge g_p$. We are now in a position to define the semantics of $\mathcal{A}_B$. Note that, for any action description AD, the non-sensing portion of AD describes a transition relation which in turn defines a projection operator $\diamond$. We refer to $\diamond$ as the projection operator defined by AD, and we restrict attention to deterministic actions. The following definition assumes a fixed underlying revision operator $*$.

Definition 4 *Let AD be an action description with corresponding projection operator $\diamond$. Let $\circ$ be the belief evolution operator obtained from $\diamond$ and $*$. For every pointed belief state $\langle s, \kappa \rangle$ and every epistemic action sequence $\bar{A} = A_1, \ldots, A_n, O$, define $\Phi_{AD}(\langle s, \kappa \rangle, \bar{A}) = \langle s', \kappa' \rangle$ where*

1. $s' = s \diamond A_1 \diamond \cdots \diamond A_n$
2. $\kappa' = \kappa \circ EFF(O, s')$.

Hence, the transition relation associated with AD returns a new pointed belief state. The new actual world is obtained by updating s by the non-sensing actions in $\bar{A}$. The new belief state is obtained by belief evolution.

The content of Definition 4 for action sequences of length 1 is as follows.

1. For non-sensing A: $\Phi_{AD}(\langle s, \kappa \rangle, A) = \langle s \diamond A, \kappa \diamond A \rangle$.
2. For sensing O: $\Phi_{AD}(\langle s, \kappa \rangle, O) = \langle s, \kappa * EFF(O, s) \rangle$.

For longer action sequences, we use belief evolution to revise the initial belief state before determining the effects of A.

Example (cont'd) The crocodile example can be represented in $\mathcal{A}_B$ by taking the representation from §3.3 and replacing the causes-to-know proposition with the corresponding causes-to-believe proposition. Note that $Sick^{-1}(Feed)$ is the set $|FullDuck \wedge \neg Chicken|$. Therefore, according to the semantics of $\mathcal{A}_K$, the final belief state should be

$$\kappa * |FullDuck \wedge \neg Chicken| \diamond Feed.$$

Regardless of the operator $*$, Bob's new belief state will be non-empty and it will only include states where the food bag contains duck.

4.3 Reliable Action Descriptions

Note that the crocodile example only involves propositions of the form: O **causes to believe** ϕ **if** ϕ. Observations of this form can be understood to represent reliable observations. In general, we say that an action description AD is *reliable* if $g_1 \wedge \cdots \wedge g_p \models \phi$ for every sensing effect proposition in AD with the form

$$O \textbf{ causes to believe } \phi \textbf{ if } g_1 \wedge \cdots \wedge g_p.$$

By contrast, the action description would not be reliable if it contained the proposition

$$LookAtCroc \textbf{ causes to believe } Sick.$$

In this case, looking at the crocodile causes the agent to believe it is sick, whether or not it is actually sick.

Reliable action descriptions describe infallible sensing actions. The following proposition formalizes the fact that, if an agent has correct knowledge of the world, then the conclusions drawn from reliable observations must also be correct.

Proposition 1 *Let AD be a reliable action description and let $\langle s, \kappa \rangle$ be a pointed knowledge state. For any epistemic action sequence $\bar{A}$, it follows that $\Phi_{AD}(\langle s, \kappa \rangle, \bar{A})$ is a pointed knowledge state.*

4.4 Representing $\mathcal{A}_K$

We can give a translation from $\mathcal{A}_K$ action descriptions to reliable $\mathcal{A}_B$ action descriptions.

Definition 5 *Let AD be an action description in $\mathcal{A}_K$. The $\mathcal{A}_B$ action description $\tau(AD)$ is obtained from AD by replacing every sensing proposition with sensing effect f and precondition ψ by the following propositions:*

$$O \textbf{ causes to believe } f \wedge \psi \textbf{ if } f \wedge \psi$$
$$O \textbf{ causes to believe } \neg f \wedge \psi \textbf{ if } \neg f \wedge \psi$$
$$O \textbf{ causes to believe } \neg \psi \textbf{ if } \neg \psi.$$

The following proposition illustrates the correspondence between the given epistemic action languages.

Proposition 2 *Let AD be an $\mathcal{A}_K$ action description, let O be a sensing action in AD and let κ be a belief state. Then $\Phi^K_{AD}(\kappa, O, \kappa')$ if and only if there is some $s \in \kappa$ such that $\Phi_{\tau(AD)}(\langle s, \kappa \rangle, O) = \langle s, \kappa' \rangle$.*

Proposition 2 illustrates that $\mathcal{A}_K$ action descriptions are interpreted disjunctively, by determining all possible outcomes κ' under the assumption that the actual world is in κ.

4.5 Comparison with Related Formalisms

Son and Baral define an alternative extension of $\mathcal{A}$, in which sensing effects are given by propositions of the form O **determines** f [Son and Baral, 2001]. We can define a translation σ from $\mathcal{A}_S$ to $\mathcal{A}_B$ by replacing each such proposition with two propositions: O **causes to believe** f **if** f and O **causes to believe** $\neg f$ **if** $\neg f$.

Proposition 3 *Let AD be a set of Son-Baral propositions. Restricted to pointed knowledge states, the transition relation $\Phi_{\sigma(AD)}$ is equivalent to the corresponding Son-Baral transition relation.*

Hence, $\mathcal{A}_B$ subsumes both of the epistemic extensions of $\mathcal{A}$. Moreover, $\mathcal{A}_B$ is the only one of the three extensions that allows erroneous beliefs and respects the interaction between sensing actions and ontic actions.

In the epistemic Situation Calculus, belief is defined through a ranking function on initial states that persists as ontic actions are executed [Shapiro *et al.*, 2000]. For any situation s, the belief set $Bel(s)$ is the set of minimally ranked situations consistent with all sensing actions executed. If we restrict attention to Situation Calculus theories where action effects are deterministic and the initial situations all correspond to distinct states, then we have the following result.

Proposition 4 *If A is an ontic action and O is a sensing action, then there is a belief evolution operator $\circ$ such that $Bel(do(O, do(A, S_0))) = Bel(S_0) \circ \langle A, O \rangle$.*

It follows that we can translate action theories in the epistemic Situation Calculus into equivalent action descriptions in $\mathcal{A}_B$. The same can not be said for the epistemic Fluent Calculus, where sensing results satisfy the AGM postulates [Jin and Thielscher, 2004]. Our work suggests that, in some action domains, simply revising the current belief state leads to unintuitive results. To represent such domains, we would need to define belief evolution operators directly in the Fluent Calculus. This would be straightforward to do.

5 Implementation Considerations

In this section, we illustrate how we can solve problems involving iterated belief change due to action through answer set planning. We proceed as follows. First, we define a revision operator based on path length. Next, we introduce an informal procedure that can be used to solve belief evolution problems with respect to this operator. We then present a translation from $\mathcal{A}_B$ to answer set programming to illustrate how the procedure can be automated.

5.1 Topological Revision Operators

A transition relation on states can be used to define a natural AGM revision operator. Let Φ be a transition relation, and let κ, α be sets of states. For technical reasons, we assume that every state in α is reachable from κ by a finite path in Φ. Define $\kappa * \alpha$ to be the subset of elements of α that can be reached by a minimum length Φ-path. Under the assumption that every state in α is reachable from κ, it follows that $*$ defines an AGM revision operator; we refer to this as the *topological revision operator* defined by T.

We are interested in topological revision operators for two reasons. First, topological revision operators do not require any external notion of similarity: they depend only on the underlying transition system. Second, topological revision operators are well-suited for the implementation that we propose in the next sections. We do not wish to imply, however, that topological revision is appropriate for all action domains; topological revision is only appropriate for domains where erroneous beliefs can be explained by action occurrences.

Example (cont'd) We extend the crocodile example by introducing a new action called $ExchangeFood$ which toggles the value of the fluent $Chicken$. This new action allows an agent to change the food available to feed the crocodile. Consider the crocodile example extended with this new action, and suppose that $*$ denotes the topological revision operator. We saw earlier that Bob needs to evaluate the expression

$$\kappa * |FullDuck \wedge \neg Chicken|.$$

By definition, we need to find the subset of $|FullDuck \wedge \neg Chicken|$ that can be reached by a minimal length path from κ. As such, the result of the revision is

$$\kappa \diamond ExchangeFood \diamond Feed.$$

Therefore, Bob's final belief state is

$$\kappa \diamond ExchangeFood \diamond Feed \diamond Feed.$$

Informally, Bob explains the fact that the crocodile is sick by postulating that the chicken was replaced with duck, and the crocodile was already fed once. This is a plausible conclusion in this example. Postulating actions to explain observations can also lead to non-trivial inferences in some action domains. For example, if we extend the example further to allow Bob to keep an inventory of the number of ducks remaining, then topological revision would suggest that he should reduce the total by two ducks after he observes the crocodile's sickness.

Topological revision is essentially a form of abduction, in which an agent looks for the shortest sequence of actions that can explain an observation. To be clear, we are not suggesting that this is suitable for all action domains. However, it is appropriate for extended crocodile-type domains where there are plausible exogenous actions explaining an observation.

5.2 Belief Evolution Under Topological Revision

In this section, we illustrate that belief evolution under topological revision can be reduced to finding shortest paths. We start by considering a single non-sensing action. Let κ denote a belief state, let A denote an action symbol, and let ϕ denote a formula. We are interested in determining

$$\kappa \circ \langle A, \phi \rangle = \kappa * \phi^{-1}(A) \diamond A.$$

Figure 1 illustrates how this is calculated with the topological revision function. The figure shows a large box representing $\phi^{-1}(A)$; these are the states that can reach $|\phi|$ by executing the action A. The circle inside $\phi^{-1}(A)$ represents

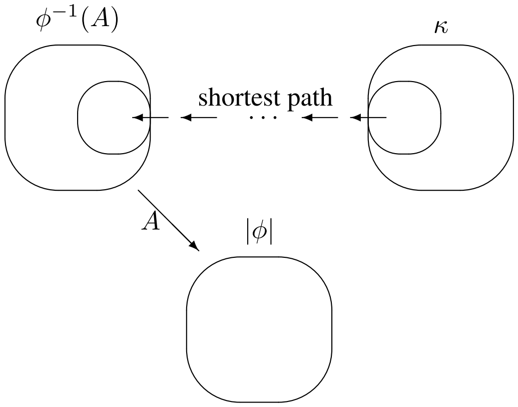

Figure 1: Visualizing Topological Evolution

the subset that is minimally distant from κ, which in this context means the elements that can be reached from κ by a minimal length path. In other words, the circle inside $\phi^{-1}(A)$ represents $\kappa * \phi^{-1}(A)$. This gives a simple procedure for computing $\kappa \circ \langle A, \phi \rangle$.

1. Determine $\phi^{-1}(A)$.
2. Let $PATH$ denote the set of shortest paths from κ to $\phi^{-1}(A)$.
3. Let κ_0 be the set of terminal nodes on paths in $PATH$.
4. Let $\kappa_1 = \kappa_0 \diamond A$.

Clearly $\kappa \circ \langle A, \phi \rangle = \kappa_1$. Hence, this procedure allows us to compute the outcome of belief evolution for trajectories of length 1. Note that steps 1,3 and 4 are straightforward; to implement a solver, we need some mechanism for determining the set of shortest paths from κ to $\phi^{-1}(A)$.

5.3 Translation to Answer Set Programming

Answer set planning refers to the approach to planning in which a problem is translated into an extended logic program where the answer sets correspond to plans [Lifschitz, 1999]. Many action languages have been translated into logic programming for answer set planning. We demonstrate how one existing translation can be modified for our purposes.

We need a translation from $\mathcal{A}$ into logic programming. Our translation is obtained by modifying a well known translation from $\mathcal{C}$ [Lifschitz and Turner, 1999]. Let AD be an action description in the action language $\mathcal{A}$. For any natural number n, we define an associated logic program $\tau_n(AD)$ with the property that answer sets for $\tau_n(AD)$ correspond to paths of length n in the transition relation described by AD. The language of $\tau_n(AD)$ consists of two disjoint classes of atoms, defined as follows. For each $i \leq n$ and each $f \in \mathbf{F}$, the language of $\tau_n(AD)$ contains an atom $f(i)$. For each $i < n$ and each $A \in \mathbf{A}$, the language of $\tau_n(AD)$ contains an atom $A(i)$. The logic program $\tau_n(AD)$ consists of the following rules:

1. for every proposition in AD of the form
$$A \textbf{ causes } (\neg)f \textbf{ if } g_1 \wedge \cdots \wedge g_p$$
for each $i < n$, $\tau_n(AD)$ contains the rules
$$(\neg)f(i+1) \leftarrow A(i), g_1(i), \ldots, g_p(i)$$
2. if B is either an action atom or B is $f(0)$ for some $f \in \mathbf{F}$, then $\tau_n(AD)$ contains the rules
$$\neg B \leftarrow \textit{not } B$$
$$B \leftarrow \textit{not } \neg B$$
3. for every $f \in \mathbf{F}$ and $i < n$, $\tau_n(AD)$ contains
$$f(i+1) \leftarrow \textit{not } \neg f(i+1), f(i)$$
$$\neg f(i+1) \leftarrow \textit{not } f(i+1), \neg f(i)$$
4. for every $i < n$, and every pair of distinct action symbols A_1, A_2, $\tau_n(AD)$ contains the rules
$$\neg A_1(i) \leftarrow A_2(i).$$

The first two sets of rules are taken directly from Lifschitz and Turner's translation of $\mathcal{C}$[Lifschitz and Turner, 1999]. Rule (3) states that all fluents are inertial, and (4) states that at most one action occurs at each point in time.

Proposition 5 *A complete set X is an answer set for $\tau_n(AD)$ if and only if it has the form*
$$\bigcup_{i=0}^{n}\{f(i) \mid f \in s_i\} \cup \bigcup_{i=0}^{n-1}\{A(i) \mid A = A_i\}$$
for some path $\langle s_0, A_0, s_1, \ldots, A_{n-1}, s_n \rangle$ in the transition relation described by AD.

Hence every answer set for $\tau_n(AD)$ corresponds to a path in the transition relation.

For the purpose of planning, it is useful to add a few rules to $\tau_n(AD)$ that restrict the admissible answer sets. Let K be the conjunction of literals $k_1 \wedge \cdots \wedge k_p$. Define $\tau_n(AD, K)$ to be the logic program obtained by adding the following rules to $\tau_n(AD)$: $k_1(0), \ldots, k_p(0)$. It is easy to see that the answer sets for $\tau_n(AD, K)$ correspond to all paths of length n which start in a state where K is true. We are interested in using answer sets to solve $|K| \circ \langle A, f \rangle$, where $\circ$ is given by the projection operator and the topological revision operator defined by AD.

To simplify the discussion, we assume that, for each A, the action description AD contains at most one proposition in AD of the form
$$A \textbf{ causes } f \textbf{ if } g_1 \wedge \cdots \wedge g_p.$$

If such a proposition exists for the action A, then define $PRE(A, f) = g_1 \wedge \cdots \wedge g_p$. Otherwise, define $PRE(A, f) = \bot$.

Proposition 6 *If $s \in S$, $f \in Lits$, and $A \in \mathbf{A}$, then*
$$s \diamond A \models f \iff s \models PRE(A, f) \vee (f \wedge PRE(A, \neg f)).$$

It follows that $f^{-1}(A) = |PRE(A, f) \vee (f \wedge PRE(A, \neg f))|$. We are now in a position to give a basic procedure for the implementation of a belief evolution solver. Given K, A, and f, define $\mathbf{evol}(K, A, f)$ to be the pair of belief states returned by the following procedure.

1. Set $n = 1$.
2. Determine all answer sets for $\tau_n(AD, K)$.
3. Let $PATH$ be the corresponding set of paths.
4. Remove from $PATH$ every path where the final state fails to satisfy $PRE(A, f) \vee (f \wedge PRE(A, \neg f)$.
    (a) If $PATH = \emptyset$, set $n = n + 1$ and goto 2.
    (b) If $PATH \neq \emptyset$, then continue.
5. Let κ_0 denote the set of final states in $PATH$.
6. Let $\kappa_1 = \{s \diamond A \mid s \in \kappa_0\}$.
7. Return $\langle \kappa_0, \kappa_1 \rangle$.

Proposition 7 *If K is a conjunction of literals, $A \in \mathbf{A}$ and $f \in Lits$, then* $\mathbf{evol}(K, A, f) = |K| \circ \langle A, f \rangle$.

Following the given approach, there are two computational problems to be solved. First, we need to find all answer sets for a given logic program at step 2; this can be accomplished by using an existing answer set solver such as smodels or DLV. The second computational task involves checking if each final state entails $f^{-1}(A)$.

Solving $|K| \circ \langle A, f \rangle$ allows us to solve projection problems for a restricted class of action descriptions in $\mathcal{A}_B$. In particular, let AD be an action description where the precondition of every sensing proposition is empty and the effect is a literal f. In this case,

$$\Phi_{AD}(\langle s, |K| \rangle, \langle A, O \rangle) = \langle s \diamond A, |K| \circ \langle A, f \rangle \rangle.$$

The actual state can be computed by the standard translation from $\mathcal{A}$ into logic programming, and the belief state can be computed as above.

We have a prototype implementation of the solver outlined in this section. The present version of the solver implements our algorithm in a straightforward manner, using smodels to determine the answer sets at step 2. In fact, the solver is slightly more powerful than the approach that we have outlined, because it allows sensing effects to be represented by an arbitrary formula rather than a single literal. We have only restricted attention to literal sensing effects in the present paper to simplify the presentation. We are currently working on extending the solver to deal with multiple observations, which introduces new complications if the observations are inconsistent. When completed, our solver will join the FLUX solver for the Fluent Calculus on a short list of implemented tools for solving problems involving iterated belief change caused by actions.

6 Discussion

Reasoning about iterated belief change caused by action requires more than an action formalism supplemented with a revision operator. The interpretation of a sensing action may depend on the preceding ontic actions, so action formalisms incorporating sensing actions need to define belief change in a manner that considers the entire action history.

We have considered an approach to the representation of iterated belief change in an action formalism by extending the action language $\mathcal{A}$ with sensing actions. Our extension differs from existing extensions in that we allow agents to have erroneous beliefs. Moreover, by defining the semantics in terms of belief evolution operators, we are able to respect the non-elementary interaction between ontic actions and sensing actions. To solve belief change problems in our action language, we presented a procedure for automating the solution of belief evolution problems through answer set planning.

In future work, we would like to extend the language to permit a wider range of sensing effects. In particular, we would like to be able to reason about the beliefs of multiple agents performing actions that affect each agent's beliefs in a different manner. Action domains of this form can be represented by allowing formulas of modal doxastic logic to be the effects of actions. The semantics of such formulas can be defined with respect to multi-agent belief structures [Herzig *et al.*, 2004]. We are currently working on a multi-agent extension based on modal logic.

References

[Alchourrón *et al.*, 1985] C.E. Alchourrón, P. G ardenfors, and D. Makinson. On the logic of theory change: Partial meet functions for contraction and revision. *Journal of Symbolic Logic*, 50(2):510–530, 1985.

[Dalal, 1988] M. Dalal. Investigations into a theory of knowledge base revision. In *Proceedings of AAAI88*, pages 475–479, 1988.

[Gelfond and Lifschitz, 1993] M. Gelfond and V. Lifschitz. Representing action and change by logic programs. *Journal of Logic Programming*, 17:301–321, 1993.

[Herzig *et al.*, 2004] A. Herzig, J. Lang, and P. Marquis. Revision and update in multi-agent belief structures. In *Proceedings of LOFT 6*, 2004.

[Hunter and Delgrande, 2005] A. Hunter and J.P. Delgrande. Iterated belief change: A transition system approach. In *Proceedings of IJCAI05*, pages 460–465, 2005.

[Jin and Thielscher, 2004] Y. Jin and M. Thielscher. Representing beliefs in the fluent calculus. In *Proceedings of ECAI04*, pages 823–827, 2004.

[Lifschitz and Turner, 1999] V. Lifschitz and H. Turner. Representing transition systems by logic programs. In *Proceedings of LPNMR99*, pages 92–106, 1999.

[Lifschitz, 1999] V. Lifschitz. Action languages, answer sets and planning. In K.R. Apt, V.W. Marek, M. Truszczynski, and D.S. Warren, editors, *The Logic Programming Paradigm: A 25-Year Perspective*, pages 357–373. Springer-Verlag, 1999.

[Lobo *et al.*, 2001] J Lobo, G. Mendez, and S.R. Taylor. Knowledge and the action description language $\mathcal{A}$. *Theory and Practice of Logic Programming*, 1(2):129–184, 2001.

[Shapiro *et al.*, 2000] S. Shapiro, M. Pagnucco, Y. Lesperance, and H.J. Levesque. Iterated belief change in the situation calculus. In *Proceedings of KR 2000*, pages 527–538. Morgan Kaufmann Publishers, 2000.

[Son and Baral, 2001] T. Son and C. Baral. Formalizing sensing actions: A transition function based approach. *Artificial Intelligence*, 125(1-2):19–91, 2001.